

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Хміль Олег Олегович

Прізвище, ім'я, по батькові студента

на здобуття ступеня вищої освіти Бакалавра

Вебсайт біржі фрілансу

Назва теми

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

Шифр КвРПЗ. 200175.01.22.ПЗ

Виконав студент IV курсу, група ПЗ-20-1

Підпис

Олег ХМІЛЬ

Ім'я, ПРІЗВИЩЕ

Керівник канд. пед. наук, доцент

Науковий ступінь, вчене звання

Підпис

Наталія ПРАВОРСЬКА

Ім'я, ПРІЗВИЩЕ

Нормоконтролер к.т.н., доцент

Посада

Підпис

Оксана ЯШИНА

Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення

Підпис


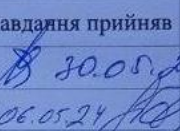
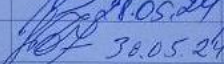
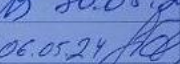
Леонід БЕДРАТЮК

Ім'я, ПРІЗВИЩЕ

6 червня 2024 р.

Хмельницький 2024

6. Консультанти розділів дипломного проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Яшина О.М., доцент	 28.05.24	 30.05.24
Антиплагиат	Форкун Ю. В., доцент	 30.05.24	 06.05.24

7. Дата видачі завдання «08» 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Збір матеріалу за темою кваліфікаційної роботи (КвР); дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	01.01 – 20.02.2024	
3 Проектування програмного забезпечення	21.02 – 20.03.2024	
4 Програмна реалізація з використанням відповідних засобів розробки. Тестування ПЗ	21.03 – 30.04.2024	
5 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог стандартів	01.05 – 25.05.2024	
6 Попередній захист КвР	Травень 2024	Згідно графіка
7 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2024	
8 Здача КвР на кафедрі; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР	з 01.06.2024	

Студент


Підпис

Олег ХМІЛЬ

Ім'я, ПРІЗВИЩЕ

Керівник роботи


Підпис

Наталія ПРАВОРСЬКА

Ім'я, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Вебсайт біржі фрілансу».

Автор проєкту: Хміль Олег Олегович.

Керівник проєкту: Праворська Наталія Іванівна.

Пояснювальна записка: 109 с., 20 рис., 4 табл., 3 дод., 41 джерел.

Графічна частина: 15 слайдів.

БІРЖА ФРІЛАНСУ, ВЕБСАЙТ, PHP

Метою проєкту є розробка вебсайту для біржі фрілансу, що дасть можливість виставляти проєкти або відгукуватись на проєкти і забезпечити зручний та зрозумілий інтерфейс для користувачів.

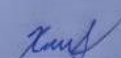
У кваліфікаційній роботі було проведено дослідження предметної області та постановку задач, спроектовано програмне забезпечення, розроблено і протестовано застосунок та підведено підсумки.

Для розробки була використана мова програмування PHP, JS і також мова стилю CSS і мова розмітки HTML та база даних MySQL.

В результаті було розроблено вебсайт для біржі фрілансу, який має дружній інтерфейс, підтримує різноманітні функціональні можливості, такі як: створення акаунтів користувача, створення проєктів, відстеження прогресу.

06.06.2024

Дата



Підпис

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ. 200175.01.22.ПЗ	Пояснювальна записка	109		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4		Презентаційні матеріали	15		
5	A3		Креслення	3		

КвРІПЗ. 200175.01.22.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав		Хміль О.О.	<i>[Підпис]</i>	21.05
Керівник		Праворська Н.І.	<i>[Підпис]</i>	21.05
Н. Контр.		Яшина О.М.	<i>[Підпис]</i>	21.05
Зав. Каф.		Бедратюк Л.П.	<i>[Підпис]</i>	21.05
			Вебсайт біржі фрілансу	Літ.
			Відомість документів	Арк.
				Аркушів
				5
				109
ХНУ, ІПЗ-20-1				

ЗМІСТ

ВСТУП.....	7
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей.....	9
1.2 Аналіз наявного програмно-технічного забезпечення предметної області.....	12
1.3 Визначення вимог до програмного забезпечення та технічне завдання.....	22
1.4 Висновки дослідження предметної області та постановки задачі.....	26
2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	28
2.1 Проєктування архітектури та структури системи.....	28
2.2 Проєктування логічної моделі бази даних.....	34
2.3 Проєктування інтерфейсу користувача.....	39
2.4 Аналіз та вибір технологій і методів реалізації.....	44
2.5 Висновки проєктування програмного забезпечення.....	49
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ.....	50
3.1 Програмна реалізація модулів.....	50
3.2 Розробка бази даних.....	55
3.3 Керівництво користувача.....	59
3.4 Вимоги до технічних та програмних засобів.....	60
3.5 Вибір та обґрунтування методів тестування програмного забезпечення.....	61
3.6 Тестування застосунку.....	63
3.7 Аналіз результатів тестування.....	69
3.8 Висновки програмної реалізації та тестування.....	71
ВИСНОВКИ.....	73
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	75
ДОДАТОК А.....	79
ДОДАТОК Б.....	86
ДОДАТОК В.....	101

КвРІПЗ. 200175.01.22.00				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав.		Хмізь О.О.	<i>[Підпис]</i>	05.05
Керівник.		Праворська Н.І.	<i>[Підпис]</i>	05.05
Рецензент				
Н. Контр.		Яшица О.М.	<i>[Підпис]</i>	30.05
Затверд.		Бедратюк Л.П.	<i>[Підпис]</i>	05.05
Вебсайт біржі фрілансу			Літ.	Арк.
Пояснювальна записка				6
			ХНУ, ІПЗ-20-1	
			Аркуші	109

ВСТУП

Вебсайти бірж фрілансу є складними системами, які надають платформу для спілкування та виконання проєктів замовниками та фрілансерами. Ці платформи включають різноманітні технічні компоненти, спрямовані на забезпечення неперервної та ефективної роботи.

Вебсайти бірж фрілансу відрізняються складною архітектурою та багатофункціональністю. Вони служать для створення, редагування та видалення профілів користувачів, публікації та пошуку проєктів, взаємодії між замовниками та виконавцями, а також для фінансових транзакцій.

Технічні аспекти роботи бірж фрілансу охоплюють високу масштабованість, захист даних, оптимізацію завантаження серверів, ефективне управління базами даних та реалізацію алгоритмів пошуку та рекомендацій. Для забезпечення надійності та стабільності роботи системи необхідне регулярне тестування, моніторинг та реагування на потенційні проблеми.

Дана кваліфікаційна робота має на меті докладне вивчення технічних аспектів функціонування вебсайтів бірж фрілансу, їх архітектури, методів розробки та управління, а також виявлення можливих шляхів покращення та оптимізації їх роботи для користувачів.

Детальне розуміння технічних особливостей вебсайтів бірж фрілансу дозволить розширити їхню структуру та функціональність, а також знайти шляхи для вдосконалення роботи. Особлива увага буде звертатися на технології, що використовуються у цих вебсайтах, такі як: серверні та клієнтські фреймворки, мови програмування, бази даних, системи керування контентом та забезпечення безпеки. Аналіз цих компонентів допоможе зрозуміти, як вебсайти бірж фрілансу забезпечують стабільну та продуктивну роботу, а також виявити можливості для оптимізації процесів.

Також важливим аспектом є розгляд існуючих проблем розробки з якими стикаються вебсайти бірж фрілансу, таких як:

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

- проблеми з безпекою;
- нестабільна робота платформи при великому обсязі трафіку;
- складнощі з реалізацією нового функціоналу та забезпеченням задоволення потреб користувачів.

Основна мета даного дослідження полягає у виявленні та аналізі технічних аспектів функціонування вебсайтів бірж фрілансу, а також у визначенні можливостей для поліпшення їх ефективності. Це дослідження спрямоване на розкриття різноманітних аспектів технічної інфраструктури вебсайтів, щоб допомогти розробникам, адміністраторам та користувачам краще розуміти, як можна оптимізувати та поліпшити їх функціонування.

Завдання, які потрібно виконати:

- дослідити предметну область, з урахування особливостей її реалізації і функціоналу;
- виконати аналіз існуючих рішень на ринку, визначити їх переваги та недоліки і мінімальний функціонал;
- розробити систему, яка буде задовольняти вимоги користувачів;

Результатом кваліфікаційної роботи являється вебсайт біржі фрілансу, який буде надавати послуги для потреби користувачів у цій галузі.

					КВРІПЗ. 200175.01.22.00	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Вебсайти біржі фрілансу на сьогоднішній день стали не лише важливим елементом економіки, але й відображенням розвитку сучасного ринку праці. Онлайн-платформи відкривають віртуальний простір, де фрілансери та клієнти з різних країн можуть легко знаходити одне одного для виконання різноманітних проєктів та послуг. За допомогою біржам фрілансу, талановиті фахівці отримують можливість знайти роботу у будь-якій галузі для виконання найрізноманітніших завдань, тоді як компанії та підприємці можуть знаходити професіоналів для реалізації своїх проєктів. Цей аналіз є ключовим етапом для розуміння та вивчення основних аспектів цієї динамічної галузі, оскільки дана галузь відкриває можливість дослідити характеристики функціонування таких платформ, їх технологічні можливості та можливості подальшого розвитку.

Біржа фрілансу є вебплатформою, спрямованою на об'єднання фрілансерів та клієнтів для виконання різноманітних проєктів та послуг. Основною метою таких бірж є створення ефективного середовища для співпраці між самостійними спеціалістами та особами, які потребують їхніх послуг або експертизи [1].

Основні учасники на біржі фрілансу – це самі фрілансери та клієнти. Фрілансери – це незалежні спеціалісти різних галузей, які пропонують свої послуги за замовленням. Клієнти – це компанії, підприємства або приватні особи, які шукають певні послуги та готові оплатити їх виконання [2].

Область вебсайтів бірж фрілансу продовжує розвиватися, адаптуючись до змін у вимогах користувачів та технологічних інновацій. На сьогоднішній день, біржі фрілансу пропонують різноманітні послуги та можливості як для фрілансерів, так і для клієнтів. Вони створюють платформу для знаходження та виконання проєктів у різних галузях, таких як: ІТ, маркетинг, дизайн, письменництво, переклад та інші.

					КВРІПЗ. 200175.01.22.00	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Одним з основних напрямків розвитку вебсайтів бірж фрілансу є постійне удосконалення користувацького досвіду. Це охоплює вдосконалення інтерфейсу, розширення функціональності, оптимізацію швидкості та реакції вебсайту, а також підвищення рівня безпеки.

Технологічні інновації відіграють важливу роль у розвитку бірж фрілансу. Розробники систематично впроваджують нові рішення та інструменти для покращення процесу знаходження та виконання проєктів, роблячи його більш ефективним та зручним для всіх учасників платформи.

Біржі фрілансу також активно працюють над розвитком системи контролю якості та ставлять акцент на підвищенні рівня довіри між фрілансерами та клієнтами. Це охоплює впровадження систем оцінювання, збору відгуків та рекомендацій, спрямованих на забезпечення якості та надійності послуг. Огляд предметної області та опис систем управління проєктами надають базове розуміння її особливостей та вимог, що є важливими для подальшого розвитку програмного забезпечення для управління проєктами.

Аналіз потреб користувачів вебсайту біржі фрілансу становить ключовий етап у вдосконаленні цих платформ. Важливо розглядати потреби як фрілансерів, так і клієнтів, щоб забезпечити їм максимальний комфорт та зручність у використанні. Ось деякі аспекти, які важливо враховувати під час аналізу потреб користувачів:

- характеристики користувачів: вивчення особливостей та потреб різних категорій користувачів (фрілансерів, клієнтів) є важливим для створення індивідуалізованих сервісів та інтерфейсів;
- потреби фрілансерів: фахівці можуть шукати інформацію про доступні проєкти, отримувати поради щодо покращення своєї роботи;
- потреби клієнтів: клієнти можуть цікавитися процесом найму фрілансерів, оглядом їхніх робіт;
- інструменти пошуку: ефективні пошукові інструменти грають важливу роль у задоволенні інформаційних потреб користувачів;

									Арк.
									10
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

– контент: надання актуальної та цікавої інформації, такої як: статті, поради, курси, може допомогти користувачам розвиватися у своїй сфері діяльності;

– взаємодія: можливість взаємодії між користувачами, така як: обмін коментарями, відгуками, чатами, сприяє покращенню комунікації та спільної роботі на платформі.

Розробка також була висвітлена в тезах Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023» – Хміль О.О, Праворська Н.І. Вебсайт біржі фрілансу [3].

Аналіз даних інформаційних потреб дає можливість глибше зрозуміти, як саме використовувати підхід до створення програмного продукту, який відповідатиме конкретним потребам користувачів.

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

У процесі дослідження вебсайту біржі фрілансу важливо детально проаналізувати наявне програмне забезпечення, яке використовується в цій галузі. Це включає огляд та оцінку існуючих програмних рішень і технічних інструментів, що сприяють роботі з платформою фрілансу.

Аналіз наявного програмного забезпечення допомагає зрозуміти, які інструменти та методики вже застосовуються у даній галузі, їх переваги та недоліки, а також виявити можливості для подальшого вдосконалення та інновацій даної предметної області.

У цьому аналізі вивчається різноманітність програмних засобів та технічних рішень, які використовуються у керуванні біржею фрілансу. Оцінюється їхні функціональність, ефективність, рівень безпеки та зручність використання з метою визначення оптимального набору інструментів для підтримки роботи біржі та задоволення потреб її користувачів.

Для оцінки програмно-технічного забезпечення у даній галузі було вибрано три відомі платформи для управління проектами: Freelancer, Fiverr та Upwork.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Freelancer – це платформа для фрілансу, яка займає високі позиції у світі. На цьому сайті зібрана велика кількість фахівців із різних галузей, які готові виконувати завдання та проекти для клієнтів з усього світу [4].

Ось деякі переваги та недоліки цієї платформи:

Переваги:

– масштабність та надійність серверів: Freelancer.com має потужну інфраструктуру серверів, що забезпечує надійну та швидку роботу навіть при великому навантаженні;

– захист від кіберзлому та шахрайства: вебсайт використовує різні технології для захисту особистих даних користувачів;

– ефективне кешування та оптимізація швидкодії: Freelancer.com використовує методи кешування та оптимізації ресурсів для прискорення завантаження сторінок;

– розширені можливості розробки API: платформа надає розробникам широкі можливості для інтеграції зовнішніх сервісів.

Недоліки:

– недостатня стійкість до DDoS-атак: платформа іноді стикається з DDoS-атаками, що може призвести до перебоїв у роботі;

– потенційні проблеми з безпекою даних: інформація користувачів може бути під загрозою через вразливості в програмному забезпеченні;

– обмежені можливості налаштування: користувачі можуть зіткнутися з обмеженнями у налаштуванні деяких параметрів своїх профілів.

З технічної точки зору, Freelancer має солідну технічну базу, проте деякі аспекти можуть бути вдосконалені для забезпечення ще більшої кращої ефективності роботи вебсайту.

Користувачі спочатку потрапляють на головну сторінку вебсайту (рисунок 1.1), де вони бачать верхнє меню навігації, що допомагає їм легко переходити до різних розділів з визначеним функціоналом. Кожен розділ містить популярні підрозділи, які полегшують навігацію користувача, дозволяючи йому швидко

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

знайти потрібну інформацію або виконати певні дії без зайвих переходів між сторінками. Такий підхід значно економить час користувачів.

Дизайн вебсайту відзначається своєю простотою і зрозумілістю, що робить його дуже зручним для користувачів будь-якого рівня досвіду. Його мінімалістичний підхід сприяє відсутності зайвих елементів, які могли б відволікти користувача, і дозволяє легко ознайомитися з функціоналом сайту. Крім того, легко запам'ятовуваний дизайн залишає приємне враження про сервіс, що важливо для користувачів, які планують використовувати платформу на тривалий час. Це впливає на їхні враження та вибір платформи для роботи, ставлячи його у перевагу серед конкурентів.

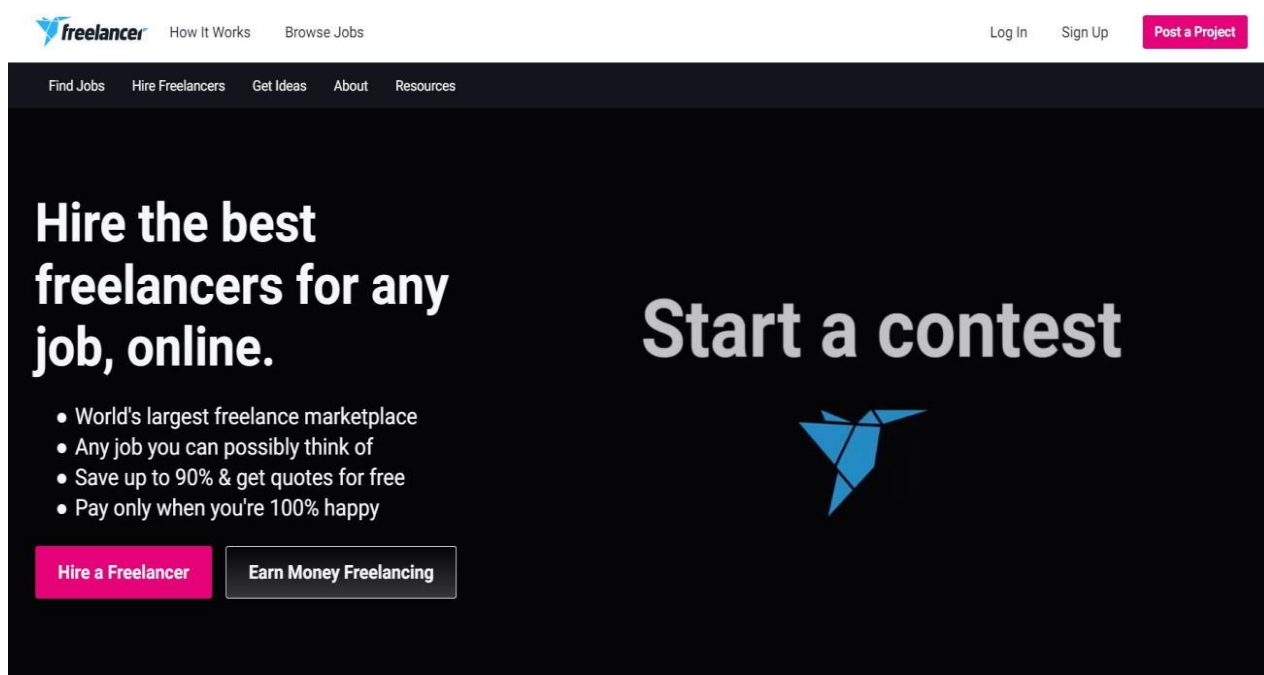


Рисунок 1.1 – Головна сторінка Freelancer

Основна функціональність сайту полягає в можливості публікації проєктів (рисунок 1.2), що дозволяє замовникам визначити завдання, які потрібно виконати, та приймати пропозиції від фрілансерів. Після цього замовник може оглянути пропозиції, відгуки та портфоліо фахівців і обрати найбільш підходящого для свого проєкту.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Важливою особливістю є можливість обговорювати винагороду за проєкт між сторонами. Це може бути здійснено через платформу, якщо вона діє як посередник, або за допомогою окремої угоди між замовником і фрілансером. Такий підхід забезпечує прозорість та визначеність умов для обох сторін. У випадку неуспішного завершення проєкту, коли робота не відповідає очікуванням або виконана неналежним чином, кошти повертаються замовнику. Але якщо проєкт завершено успішно та відповідає усім умовам, фрілансер отримує винагороду за виконану роботу [5].

Також дана модель бізнесу дозволяє платформам заробляти відсоток від кожної успішно завершеної угоди. Такий підхід є вигідним, оскільки стимулює якість та ефективність роботи фрілансерів і забезпечує надійність платформи для замовників. Дана функціональність є ключовою для вебсайтів бірж фрілансу, оскільки вона дозволяє забезпечити прозору та безпечну співпрацю між сторонами, збільшуючи довіру користувачів до платформи [6].

freelancer

Tell us what you need **done.**

Project name

Web Application Performance Testing

Project description Clear

As an individual seeking to perfect my web application, I'm in need of expert assistance to perform rigorous functionality, performance and user experience testing. Some core areas to focus on include:

- The login and registration process: Ensuring a smooth user onboarding experience by checking form validations, error handling, and successful user creation.
- Search and Filter system: Evaluate the efficiency and precision of the search operations, focusing on the responsiveness and accuracy of the filtering system.
- Transaction Processing: Validate the reliability, speed, and security of all transaction processes.

Ideal candidates for this project should have a proven track record in software testing, with an emphasis on web application analysis. Experience using testing tools and methodologies is highly desirable. Your input will play a key role in refining and evolving my application.

Drag & drop any images or documents that might be helpful in explaining your brief here. (Max 25 MB).

Next Press ENTER

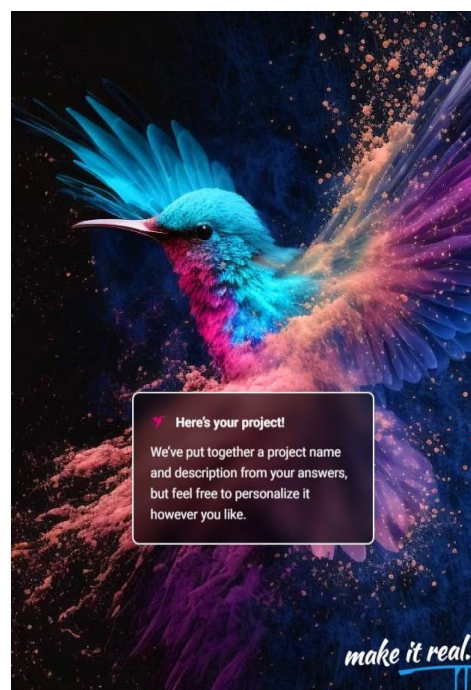


Рисунок 1.2 – Створення проєкту

									КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						14

В цілому, Freelancer представляє собою значну платформу для знаходження та замовлення послуг, проте перед використанням цієї системи користувачам слід урахувати як її переваги, так і недоліки.

Наступним вебсайтом для розгляду являється Fiverr.

Fiverr – це вебплатформа для фрілансу, що отримала значну популярність у останні роки. На Fiverr можна знайти фахівців різних напрямків, які готові виконати різноманітні завдання за різними цінами, починаючи від п'яти доларів. Основні категорії послуг включають графічний дизайн, веброзробку, копірайтинг, переклад, анімацію, маркетинг і багато інших [7].

Огляд переваг та недоліків сервісу:

Переваги:

– різноманітність послуг: Fiverr пропонує широкий спектр послуг у багатьох галузях, що дозволяє користувачам замовляти різноманітні проекти, які знаходяться на одній платформі;

– доступні ціни: багато фрілансерів на Fiverr пропонують свої послуги за доступні ціни, що дозволяє клієнтам отримувати якісні роботи без значних витрат;

– швидкість виконання проєктів: багато фрілансерів на Fiverr готові виконувати проєкти в найкоротші терміни, що робить цю платформу привабливою для тих, хто потребує швидких результатів;

– система рейтингів та відгуків: Fiverr має систему оцінок та відгуків, яка допомагає користувачам вибирати найкращих фахівців та перевіряти їх репутацію перед замовленням послуг;

– захист платежів: платформа забезпечує безпечність транзакцій через систему Escrow, що дозволяє замовникам захистити свої кошти до завершення виконання проєкту.

Недоліки:

– обмеженість варіантів комунікації: Fiverr обмежує прямий контакт між замовниками та фрілансерами, що може ускладнити обговорення деталей проєкту та узгодження умов співпраці;

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

– потенційні ризики невдачі вибору фахівця: незважаючи на систему рейтингів та відгуків, існує ризик, що фрілансер може не відповідати очікуванням замовника або надавати незадовільні послуги;

– додаткові витрати: деякі функції та опції на Fiverr можуть бути доступні лише за додаткову плату, що може збільшувати загальний бюджет проекту;

– неможливість безпосередньо контролювати роботу: замовникам іноді може бути складно контролювати процес виконання проекту, оскільки вони не мають можливості безпосередньо впливати на роботу фрілансера;

– конфіденційність даних: іноді може виникати питання щодо конфіденційності даних, оскільки замовники та фрілансери мають обмінюватися конфіденційною інформацією для виконання проектів через платформу.

При переході на вебсайт Fiverr, першою сторінкою, яку зустрине користувач, є головна сторінка (рисунок 1.3). Тут звертає на себе увагу поле пошуку, яке дозволяє швидко знаходити необхідні послуги або фахівців, а також пункти навігації та можливість реєстрації для нових користувачів.

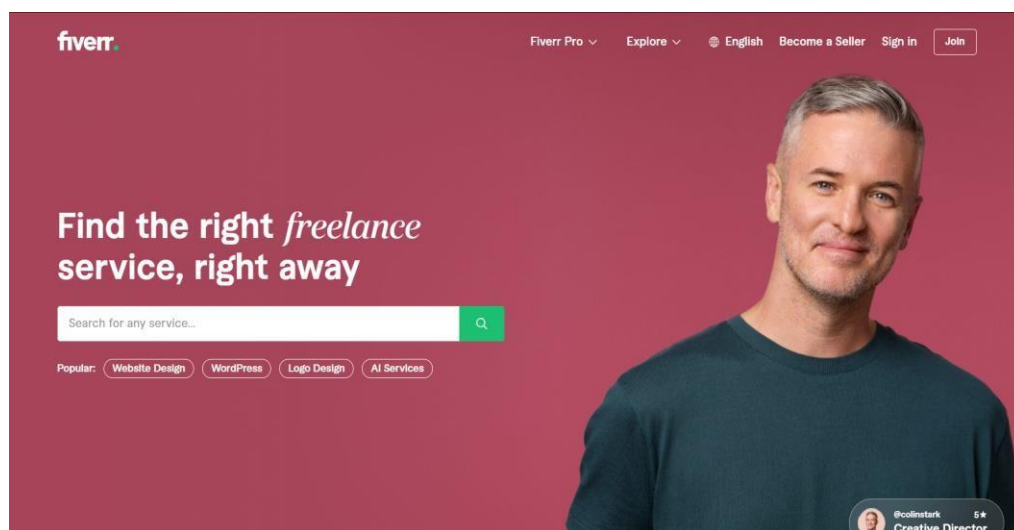


Рисунок 1.3 – Головна сторінка Fiverr

Основна особливість цього сервісу полягає в тому, що він пропонує широкий спектр послуг за певну вартість. Наприклад, для того щоб залишити відгук про виконання завдання, користувачеві доведеться оплатити відповідну суму. Проте

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

дана особливість привертає увагу тих, хто цінує якість та відповідальність. Замовники можуть бути впевнені, що на цій платформі знайдуть досвідчених фрілансерів, оскільки оплата за відправлення запиту на виконання вказує на те, що виконавець серйозно ставиться до проєкту і готовий до співпраці.

Сторінка пошуку на Fiverr (рисунок 1.4) відкриває перед користувачем можливість самостійно знайти відповідного виконавця та запропонувати проєкт для виконання. Це дає можливість користувачам активно взаємодіяти з платформою та знаходити ідеальний варіант для виконання своїх проєктів чи завдань.

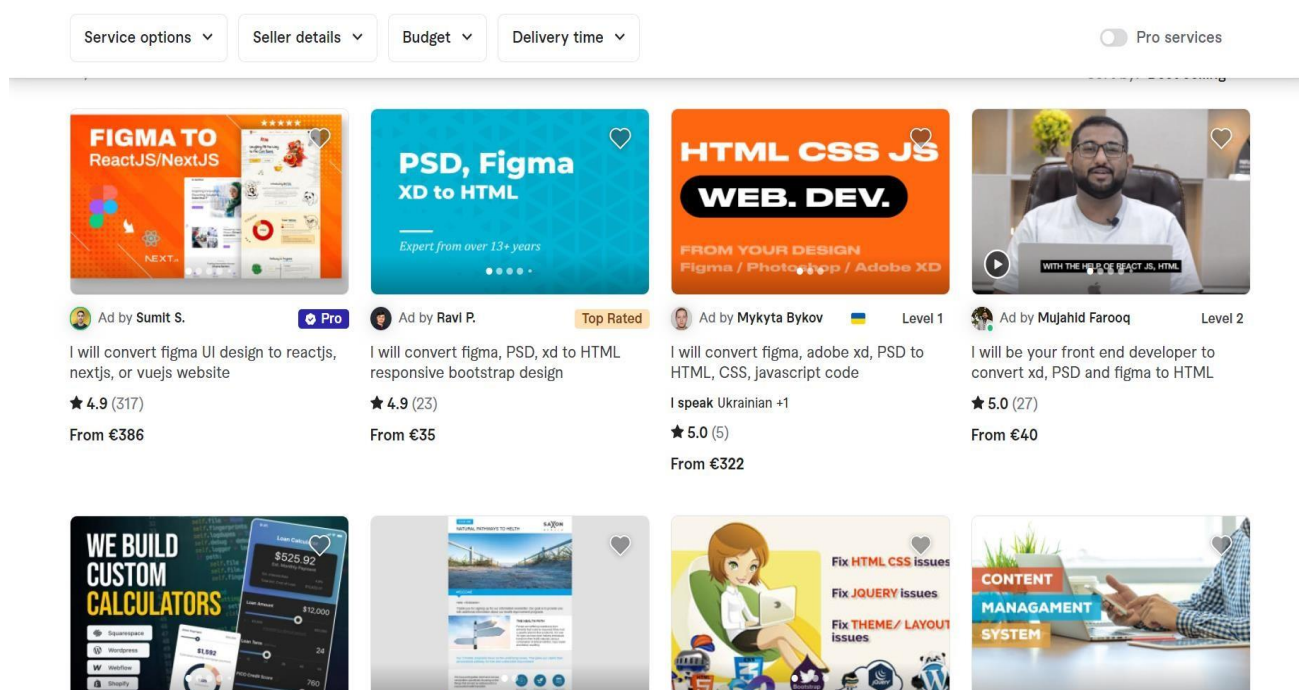


Рисунок 1.4 – Пошук фрілансерів на Fiverr

Якщо виконавець погоджується на запропоновану суму за успішне виконання завдання, це відкриває шлях до співпраці між обома сторонами. Такий підхід сприяє ефективній комунікації та взаєморозумінню між замовником і виконавцем, що допомагає досягти успішного результату в роботі над проєктом.

Хоча дизайн вебсайту Fiverr достатньо зручний, він все ж має свої мінуси, які можуть вплинути на сприйняття інформації. Наприклад, обмеженість функціоналу,

									Арк.
									17
Змн.	Арк.	№ докум.	Підпис	Дата					

який доступний лише за додаткову плату, може стати перешкодою для користувачів, які не бажають або не можуть витратити додаткові кошти на розширення можливостей платформи.

Сервіс Fiverr може бути досить зручним і корисним для багатьох користувачів, проте варто мати на увазі, що значна частина його функціоналу доступна лише за додаткову плату. Це може стати перешкодою для тих, хто шукає безкоштовні або більш доступні рішення.

Останнім в огляді існуючих рішень виступає Upwork.

Upwork є популярною онлайн-платформою, яка забезпечує зв'язок між фрілансерами та клієнтами з усього світу. Завдяки широкому спектру послуг, включаючи: веброзробку, дизайн, маркетинг, переклад, написання текстів та адміністрування, також дана платформа виконує потреби різноманітних бізнесів та індивідуальних користувачів [8].

Переваги Upwork:

- широкий вибір фрілансерів: платформа має велику кількість зареєстрованих фахівців із різних куточків світу;
- різноманіття послуг: тут можна знайти експертів практично у всіх галузях, які існують на сервісі;
- зручний пошук: потужний інструмент пошуку та фільтрації дозволяє швидко знаходити потрібних фрілансерів;
- деталізовані профілі: інформативні профілі фахівців містять усю необхідну інформацію для прийняття правильного рішення;
- система рейтингу та відгуків: оцінка та відгуки користувачів допомагають знайти надійних спеціалістів;
- безпечна платформа: використання HTTPS-протоколу забезпечує захист персональних даних користувачів.

Недоліки Upwork:

- висока конкуренція: через велику кількість фрілансерів може бути складно здобути нових клієнтів;

					КВРІПЗ. 200175.01.22.00	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

- низькі ціни: багато фахівців, які пропонують послуги за дуже низькими тарифами за виконану роботу;
- комісійні: платформа забирає частку від кожної транзакції;
- обов'язкова реєстрація: для зв'язку з фрілансерами необхідно мати акаунт на сайті.

На головній сторінці вебсайту зустрічає досить простий та зрозумілий інтерфейс (рисунок 1.5), що робить його приємним для сприйняття. Весь необхідний функціонал відображений у досить простому вигляді, що створює позитивне враження від користування сервісом.

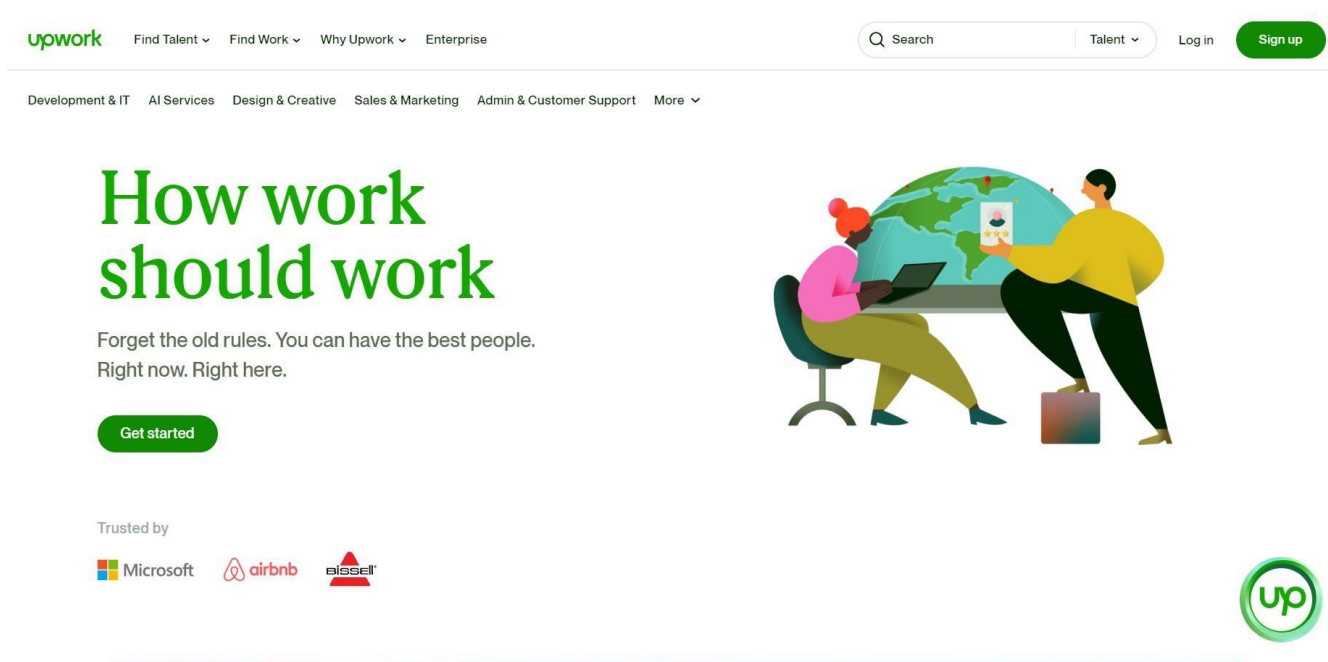


Рисунок 1.5 – Головна сторінка Upwork

На головній сторінці присутня навігаційна панель, яка полегшує пошук потрібного функціоналу, а також рядок пошуку. Сама головна сторінка надає можливість переглянути основні аспекти роботи вебсайту та перейти до дій, проте для початку роботи рекомендується зареєструватись, щоб отримати повний функціонал. Головною перевагою вебсайту є його простота та широкий спектр функцій, що робить його дуже зручним для будь-якого фрілансера або замовника, оскільки сервіс вже відомий у просторах Інтернету.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

При створенні акаунту на вебсайті, користувачеві одразу надається можливість опублікувати або прийняти замовлення (рисунок 1.6), проте це варіюється залежно від обраної ним ролі.

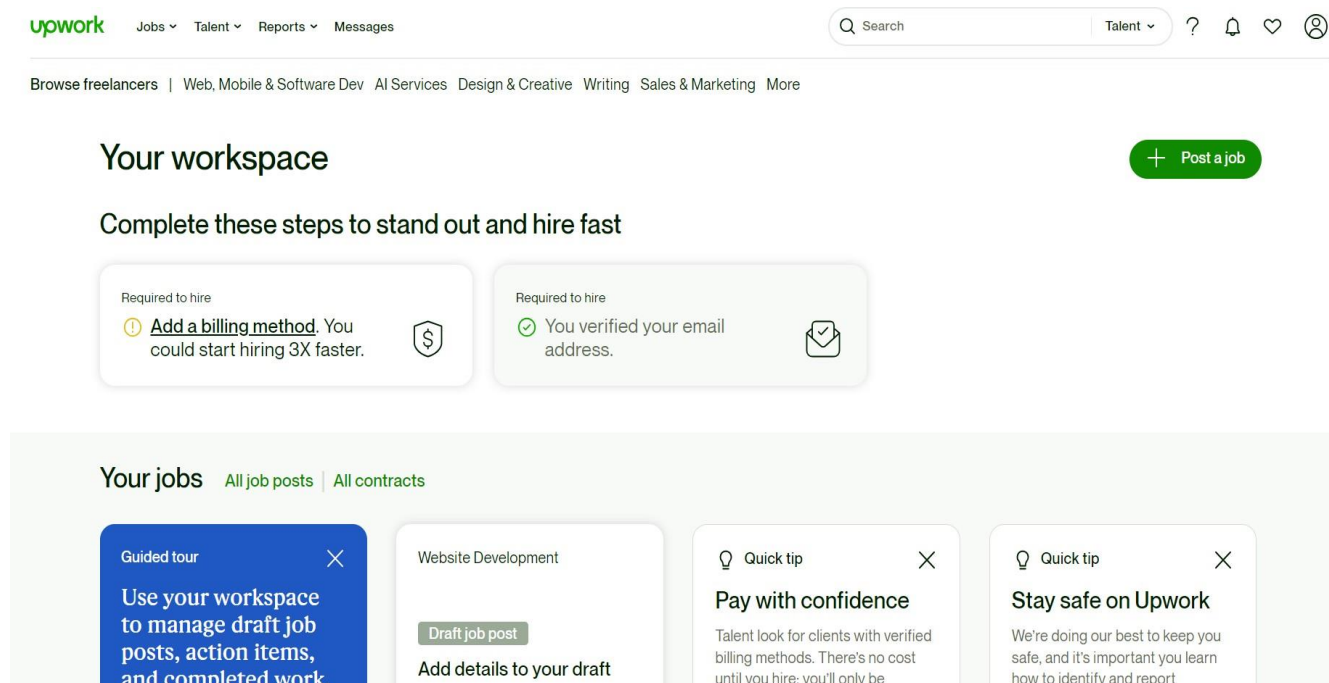


Рисунок 1.6 – Пропозиції Upwork

Розглядаючи вебсайт сервісу Upwork, можна відзначити, що сервіс виявляє схожість з попередніми платформами. Проте, завдяки широкій популярності, яку платформа має, можна впевнено стверджувати, що над платформою виконано значну кількість роботи. Даний вебсайт успішно впроваджується у даній сфері, що свідчить про його ефективність та значимість.

На основі короткого розбору цих трьох сервісів, можливо, визначити їх порівняльні характеристики в вигляді таблиці (таблиця 1.1).

Таблиця 1.1 – Порівняльні характеристики фріланс бірж

Критерій	Freelancer	Upwork	Fiverr
Кількість користувачів	50+ мільйонів	12+ мільйонів	3+ мільйони
Різноманіття проєктів	Широкий спектр	Широкий спектр	Більше орієнтовано на невеликі проєкти

Ціни	Різноманітні	Різноманітні, але загалом вищі	Фіксовані ціни за «гіги»
Комісія	3-10% (з фрілансера)	10-20% (з фрілансера)	20% (з фрілансера)
Рівень конкуренції	Високий	Високий	Дуже високий
Складність отримання проєкту	Складний	Складний	Дуже складний
Способи оплати	Різноманітні	Різноманітні	PayPal, кредитні картки
Підтримка	Базова	Розширена	Базова
Інтерфейс	Сучасний, складний для навігації	Сучасний, зручний	Сучасний, інтуїтивно зрозумілий
Переваги	Велика спільнота, багато проєктів	Багато проєктів, високі ціни	Прості у використанні, фіксовані ціни
Недоліки	Висока конкуренція, низькі ціни	Висока конкуренція, високі комісії	Дуже висока конкуренція, низькі ціни

Тому, після огляду та аналізу зібраних даних про різноманітні сервіси фрілансу, можливо зробити кілька важливих висновків. В першу чергу, існуючі платформи, такі як Fiverr, Freelancer та Upwork, мають схожі характеристики та функціонал, але кожна має свої унікальні особливості та переваги.

Однією з основних переваг цих платформ є великий вибір фахівців та різноманітність послуг, які можна замовити. Додатково, наявність системи рейтингів та відгуків дозволяє користувачам обирати надійних фрілансерів та забезпечує додатковий рівень довіри. Проте, існують і певні недоліки, такі як висока конкуренція на деяких платформах, низькі ціни за послуги, а також необхідність сплати комісій чи плати за додатковий функціонал.

Отже, враховуючи отримані дані, можливо використати їх для подальшого проєктування програмного застосунку, спрямованого на поліпшення ефективності та зручності для користувачів. Проєктування такого застосунку має на меті вирішити існуючі проблеми та задовольнити потреби користувачів у сфері біржі фрілансу.

									Арк.
									21
Змн.	Арк.	№ докум.	Підпис	Дата	КВРІПЗ. 200175.01.22.00				

1.3 Аналіз вимог до програмного забезпечення та розробка технічного завдання

Перед початком розробки програмного забезпечення, зокрема вебсайту, важливо провести детальний аналіз та уточнення всіх необхідних функцій, які повинен виконувати продукт. Цей етап визначення функціональності відіграє критичну роль у процесі розробки, оскільки від нього залежить подальший успіх проєкту у майбутньому.

Даний процес також допомагає визначити пріоритети функціональності та встановити межі проєкту, що є важливим для керування проєктом та визначення обсягу роботи. Крім того, чітко визначені функції дозволить краще розуміти завдання та це призведе до більш точних оцінок часу і ресурсів, необхідних для реалізації проєкту.

Такий підхід також сприяє підвищенню якості програмного продукту, оскільки він дозволяє виявити та вирішити потенційні проблеми та недоліки ще на ранніх етапах розробки. Крім того, такий підхід сприяє забезпеченню високої задоволеності замовника та кінцевих користувачів, оскільки враховує їх потреби та очікування від програмного забезпечення.

На діаграмі потоків даних (рисунок 1.7) відображено основний функціонал, який буде виконувати вебсайт біржі фрілансу та зв'язок між ними. Ця діаграма відображає ключові характеристики вебсайту, які задають функціонал програмного забезпечення та їх взаємодію між собою, формуючи загальну структуру вебсайту. Усі діаграми проєкту виконанні у візуальному редакторі [9].

Даний підхід дозволяє краще усвідомити, як користувачі будуть взаємодіяти з вебсайтом та які функції потрібно реалізувати для забезпечення потреб користувачів. Це важливий етап перед розробкою програмного забезпечення, оскільки користувач визначає кінцевий продукт та його можливості.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

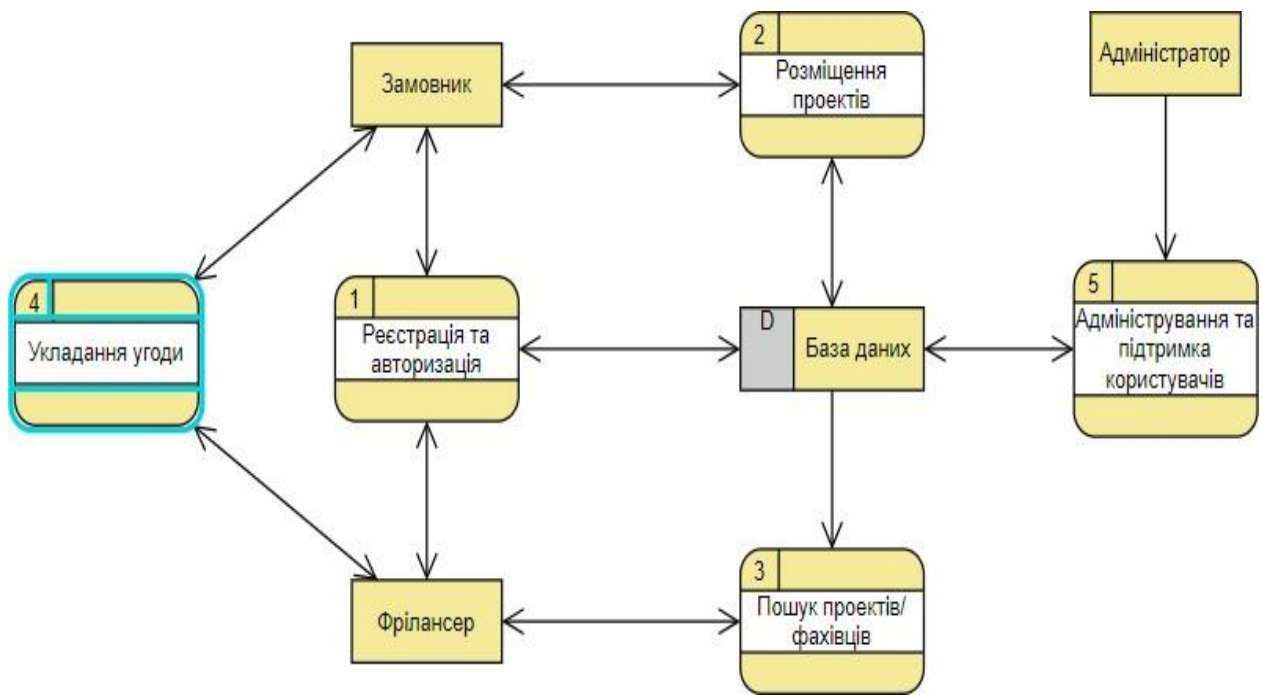


Рисунок 1.7 – DFD Діаграма

Для кращого усвідомлення ролей та процесів, які відбуваються в системі, корисно використовувати діаграми, що дозволяють візуально подати ці процеси. Одним із ефективних інструментів демонстрації для цього являється діаграма варіантів використання [10].

Для побудови діаграми використання була створена таблиця, в якій описані всі актори, що беруть участь у системі (таблиця 1.2). Це дозволить краще зрозуміти, які ролі існують та що вони можуть виконувати на вебсайті та виконувати деякі дії, які дозволені із-за отриманої ролі.

Не зареєстровані користувачі мають обмежений функціонал на вебсайті, зазвичай обмежуючись правом на перегляд контенту та можливістю реєстрації. Ці користувачі можуть читати контент, переглядати зображення або відео, а також ознайомлюватися з іншою публічно доступною інформацією. Однак, вони часто не можуть залишати коментарі, брати участь у форумах, або користуватися іншими інтерактивними можливостями сайту.

Таблиця 1.2 – Опис акторів розроблюваного ПЗ

Тип користувачів	Короткий опис
Не зареєстрований користувач	Немає доступу до функціоналу, але має можливість переглядати контент та реєструватись, щоб отримати в подальшому усі функції вебсайту відповідно від вибраної ролі.
Замовник	Має доступу до таких функцій, як створення замовлення, тобто проекту, перегляд фрілансерів та пошук інформації на веб сайті
Фрілансер	Може створювати та працювати вже з існуючими проектами, задачами, додавати коментарі, редагувати інформацію про себе
Адміністратор	Має усі права доступу, а також можливість змінювати контент вебсайту, додавати новий функціонал та перегляд усіх даних

Адміністратор (рисунок 1.8), натомість, має повний доступ до усіх функцій на вебсайті. Він може виконувати будь-які маніпуляції, які дозволені у доступі, включаючи управління контентом, користувачами, налаштуваннями безпеки та інших параметрів.

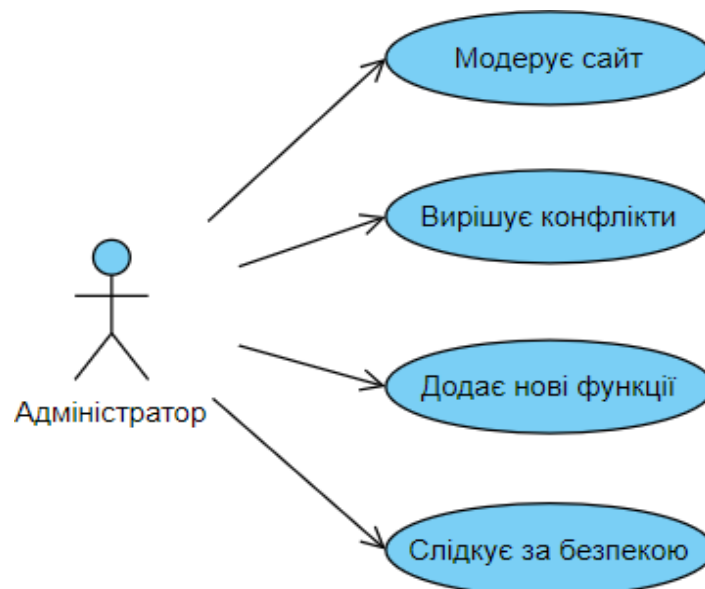


Рисунок 1.8 – Діаграма варіантів використання для адміністратора

Зареєстровані користувачі, як замовники, так і виконавці (рисунок 1.9), матимуть доступ до всього функціоналу вебсайту, в залежності від обраної ролі під час реєстрації.

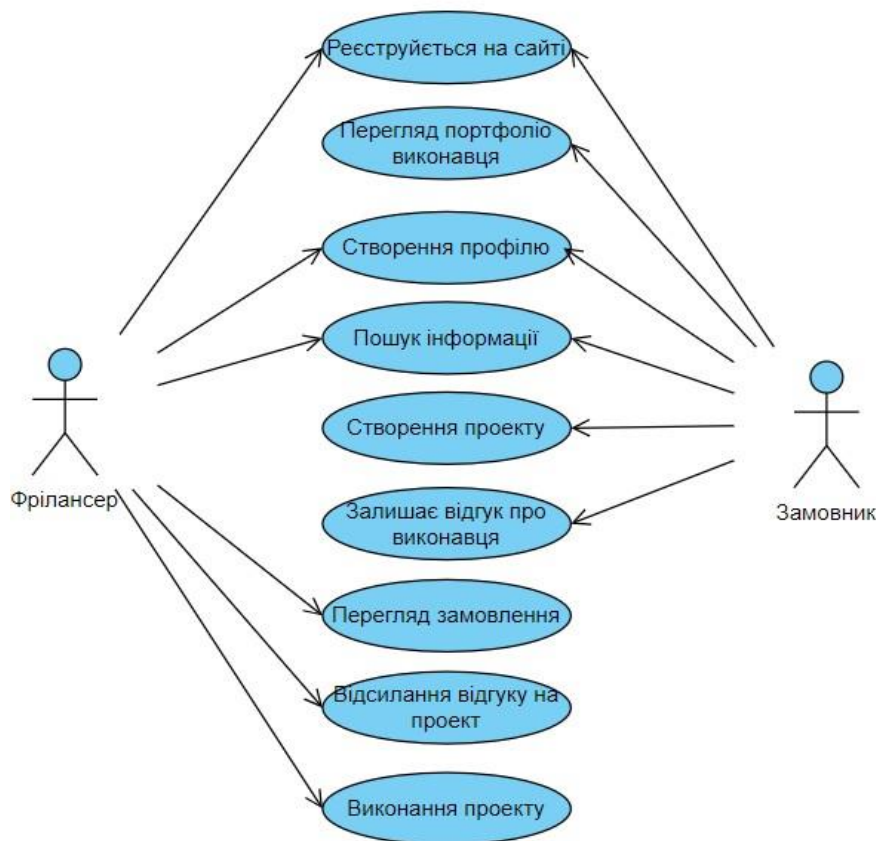


Рисунок 1.9 – Діаграма варіантів використання користувачів

Це означає, що вони зможуть виконувати всі необхідні маніпуляції, пов'язані з їхніми потребами та цілями на платформі, а саме зареєстровані користувачі можуть:

- створювати профілі: користувачі можуть створювати власні профілі, де вони представляють інформацію про себе, свої навички, досвід та робочі здібності;
- переглядати та редагувати профілі: користувачі можуть переглядати та редагувати свої особисті дані, включаючи фотографії, описи та інші деталі;
- пошук та перегляд проектів: користувачі можуть шукати проекти, які відповідають їхнім навичкам та інтересам, і переглядати деталі кожного проекту;

- подача пропозицій: користувачі можуть подавати пропозиції на проекти, які їх зацікавили та вказувати свої умови та ціни;
- управління проектами: користувачі можуть керувати всіма аспектами своїх проектів, включаючи комунікацію з клієнтами або виконавцями, встановлення термінів та обговорення деталей;
- надання та отримання відгуків: користувачі можуть залишати відгуки та оцінки для інших користувачів після завершення проекту, що допомагає будувати репутацію та довіру;
- комунікація: користувачі можуть спілкуватися один з одним через вбудовані інструменти чату або електронної пошти, обговорюючи деталі проектів та узгоджуючи умови співпраці.

Також технічне завдання було розроблено та наведено в «Додатку А».

Отже, у всіх попередніх схемах були відзначені функціональність та ролі, що будуть присутні в майбутньому вебсайту біржі фрілансу. Головними складними компонентами розробки розглядаються розподіл ролей та рівні доступу, які будуть надаватися відповідно до певного шаблону, який слід буде ретельно підготувати, а також реалізація взаємодії між замовником та виконавцем.

1.4 Висновки дослідження предметної області та постановки задачі

Проведений аналіз предметної області дозволив зрозуміти основні аспекти та особливості, що впливають на розробку програмного продукту. Це дало можливість коректно визначити потреби користувачів та вимоги до функціональності системи.

Проведений аналіз наявного програмно-технічного забезпечення дозволив виявити існуючі рішення та їхні переваги та недоліки. Це допомогло уникнути повторного винайдення велосипеда та вибрати оптимальні технології та методи реалізації.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Чітко сформульовані вимоги та технічне завдання визначили основні цілі та обсяг робіт. Це надало зрозуміле керівництво для подальшої розробки програмного продукту та допомогло уникнути непорозумінь між розробниками та замовником.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проєктування архітектури та структури системи

В реалізації проєкту надзвичайно важливим є вибір архітектурного підходу, що найбільш ефективно відповідає вимогам та цілям проєкту. Для цього досліджується широкий спектр архітектур, призначених для розробки вебсайтів. При цьому критичним є ретельний аналіз кожної архітектури з погляду її здатності забезпечити високий рівень продуктивності, масштабованості та зручного користувацького досвіду. Вибір належної архітектури ґрунтується на об'єктивних оцінках та ретельному вивченні їхніх переваг і недоліків. Такий підхід гарантує оптимальне використання ресурсів та забезпечує відповідність проєкту сучасним стандартам розробки вебсайту [11].

Тому для реалізації проєкту ретельно розглядаються різноманітні архітектурні концепції з метою вибору оптимального рішення, що відповідає вимогам та специфіці проєкту. Серед найбільш розповсюджених архітектур на сьогоднішній день варто відзначити класичну клієнт-серверну архітектуру, яка забезпечує чітке розділення обов'язків між клієнтом та сервером, а також однострінкові програми (SPA), що надають можливість покращення швидкості завантаження сторінок та вдосконалення користувацького досвіду. Зазначені архітектурні підходи відомі своєю ефективністю та широким застосуванням у сфері веброзробки. Однак на відміну від них, існують і більш спеціалізовані концепції, такі як мікросервісна архітектура, прогресивні застосунки (PWA), а також peer-to-peer (P2P) мережі, кожна з яких має свої особливості та переваги. Вибір оптимального архітектурного підходу базується на комплексному аналізі вимог проєкту, ресурсів та завдань, що потрібно врахувати при розробці [12].

Різні архітектурні концепції для вебсайтів пропонують різні підходи до організації та розробки програмного забезпечення. Класична клієнт-серверна архітектура, що є базовою для більшості сайтів, розподіляє систему на дві основні складові: клієнтську частину, яка відповідає за відображення інтерфейсу

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

користувача, і серверну частину, яка обробляє запити від клієнта та надає йому необхідні дані. Цей підхід дозволяє ефективно відділяти функціонал клієнта від серверної логіки та спрощує розробку та підтримку системи.

Однострінкові програми (SPA) відрізняються тим, що всі необхідні ресурси (HTML, CSS, JavaScript) завантажуються під час першого запуску, а подальша навігація відбувається без перезавантаження сторінки. Цей підхід забезпечує швидку відповідь на дії користувача та знижене навантаження на сервер, але може призвести до збільшення розміру сторінки та потребує складного управління станом додатку.

Мікросервісна архітектура відрізняється розбиттям великої системи на невеликі незалежні мікросервіси, кожен з яких відповідає за виконання конкретного функціоналу. Цей підхід дозволяє спростити розробку, розгортання та підтримку системи, але може призвести до збільшення складності управління та моніторингу, а також до проблем з координацією між сервісами.

Прогресивні вебдодатки (PWA) використовують сучасні технології для створення відчуття роботи нативних застосунків. Ці додатки мають можливість працювати офлайн, швидко реагувати на дії користувача та підтримують ряд інших продвинутих функцій, що розширюють їхню функціональність, але можуть вимагати складнішої реалізації та обмежені можливості на деяких пристроях та браузерах у використанні.

Серед усіх розглянутих архітектур обраною для вебсайту фрілансу є клієнт-серверна. Цей вибір здійснено з ряду причин, серед яких переважають два ключових фактори.

Одна з причин те, що клієнт-серверна архітектура вже давно є стандартом у веброботці та є добре вивченою та зрозумілою. Стабільність даної архітектури та ефективність підтверджена та перевірена часом, що робить її кращим варіантом для реалізації проєктів будь-якої складності [13].

А друга з причин, клієнт-серверна архітектура відзначається простотою та зручністю у розробці та підтримці. Розділення функціональності на клієнтську та

									Арк.
									29
Змн.	Арк.	№ докум.	Підпис	Дата					

серверну частину спрощує розробку та підтримку коду, дозволяючи розробникам ефективно працювати над окремими аспектами системи.

У порівнянні з альтернативними архітектурними концепціями, які включають однострінкові програми, мікросервісну архітектуру та прогресивні вебсайти, клієнт-серверна архітектура може виявитися переважним вибором у контексті даного проєкту. Тому потрібно виконати аналіз переваг та недоліків кожної з цих архітектур з метою обґрунтування даного вибору архітектури.

Однострінкові сайти відзначаються прискореним відгуком на дії користувача та зменшенням навантаження на сервер, оскільки всі необхідні ресурси завантажуються один раз під час першого запуску. Однак дана архітектура може страждати від проблем, таких як збільшення розміру сторінки та складне управління станом додатку. Ці недоліки можуть стати неприйнятними для даного проєкту, де важливими є простота та ефективність у розробці та підтримці вебсайту у майбутньому для майбутніх користувачів.

Мікросервісна архітектура, хоча і дозволяє спростити розробку, розгортання та підтримку системи через розбиття її на невеликі незалежні мікросервіси, може страждати від проблем, пов'язаних зі збільшенням складності управління та моніторингу, а також з координацією між сервісами. Для даного проєкту, який може не потребувати настільки високого рівня гнучкості, ці проблеми можуть виявитися надмірними.

Насправді прогресивні вебдодатки, хоча і надають можливість працювати офлайн та швидко реагувати на дії користувача, можуть вимагати складнішої реалізації та обмежувати функціональність на деяких пристроях та браузерах. Для проєкту, де головними вимогами є простота та стабільність, це може бути не оптимальним вибором.

Отже, у зазначеному контексті, клієнт-серверна архітектура, завдяки своїй простоті, стабільності та зручності у розробці та експлуатації, може виявитися кращим вибором для досягнення поставлених цілей проєкту.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Клієнт-серверна архітектура є однією з провідних моделей у розробці даного програмного забезпечення, що базується на принципі розділення функцій між двома основними складовими: клієнтською та серверною частинами. У цій архітектурі клієнтська частина відповідає за інтерфейс та взаємодію з користувачем, в той час як серверна частина забезпечує обробку запитів та управління даними.

Клієнтська частина, як правило, представляється інтерфейсом користувача, що може бути браузером, мобільним додатком або іншим програмним засобом для взаємодії з користувачем. Вона містить логіку клієнта, яка відповідає за обробку подій користувача та відправку запитів до сервера.

Серверна частина включає в себе серверний застосунок та базу даних. Серверний застосунок виконується на сервері та відповідає за обробку запитів від клієнтів, включаючи обробку логіки додатку та доступ до бази даних. База даних, у свою чергу, забезпечує збереження та організацію даних, які використовуються організованою системою.

Взаємодія між клієнтом та сервером здійснюється за допомогою протоколів та API (Application Programming Interface). Найпоширенішим протоколом для взаємодії з вебзастосунками є HTTP, а для передачі даних використовуються REST API. Тому API серверної частини визначає набір правил та конвенцій для взаємодії з клієнтами та доступу до функцій, та даних сервера через спеціальні запити на сервер [14].

Даний вебсайт буде працювати наступним чином (рисунок 2.1), користувач буде відвідувати вебсторінку через браузер та за допомогою браузера користувач буде відсилати запити, тому за це буде відповідати front-end частина. Після надіслання запиту з сторони користувача надійде запит на дані, які має відправити сервер, якщо дані існують для даного запиту. Для роботи сервера відповідає back-end частина, яка буде обробляти отримані запити від користувача. В свою чергу сервер проводить перевірку запиту за допомогою розробленої бізнес-логіки.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Сервер оброблятиме запит в залежності від типу запиту та відправить відповідні дані з сервера, використовуючи дані з бази даних, якщо це потрібно.

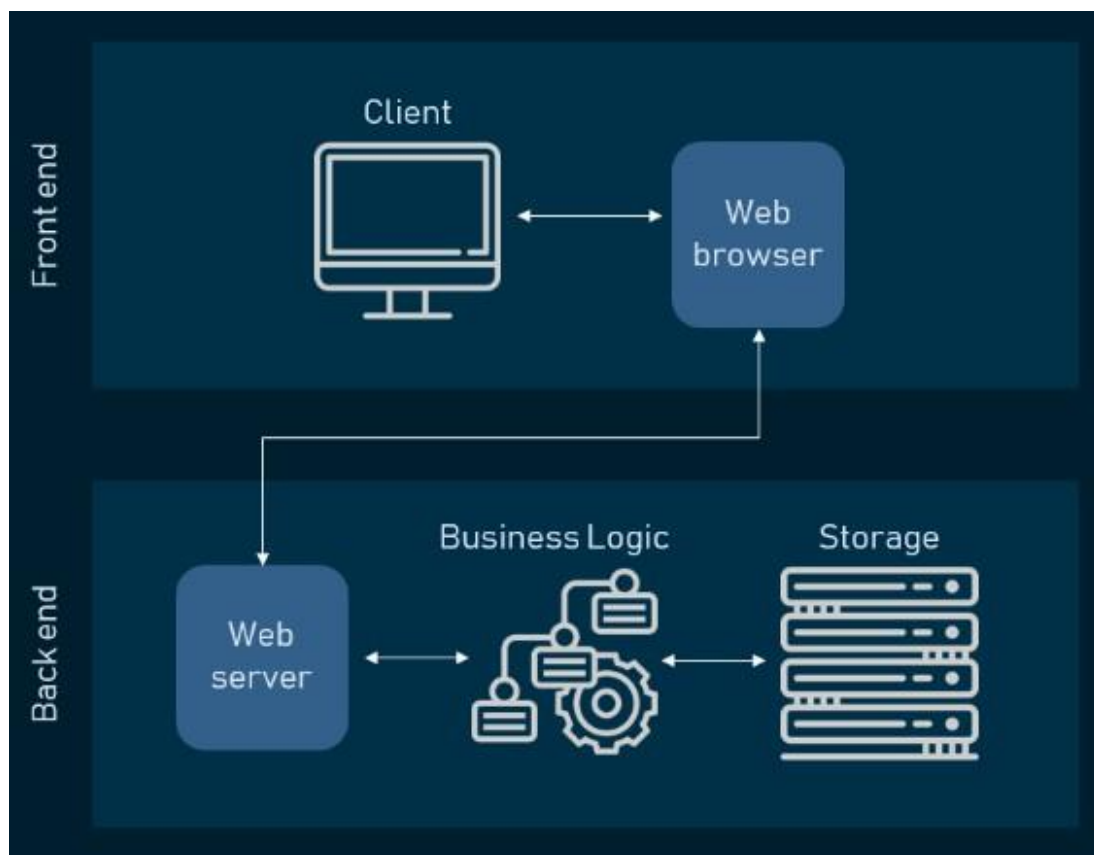


Рисунок 2.1 – Процес роботи вебсайту

Система вебсайту буде використовувати архітектурний шаблон MVC (Model–View–Controller), тобто типову архітектурну модель, що базується на розділенні функціоналу на компоненти та модулі, які взаємодіють між собою у відповідності з визначеною логікою та структурою. Основною метою цієї структури є забезпечення гнучкості, розширюваності та ефективності розробки та підтримки вебзастосунків [15].

Контролер основна складова, що розміщує ядро системи, яке визначає базові функції та принципи роботи всієї системи. Контролер відокремлений від бізнес-логіки та відображення даних, і він не має прямого доступу до бази даних. Замість цього він координує виклики до моделі для отримання необхідних даних та до виду для відображення цих даних користувачеві. Однією з ключових властивостей

						КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			32

контролера є розділення обробки запитів від їх виконання. Він відповідає за визначення потрібної логіки для обробки запитів та викликів інших компонентів системи для виконання необхідних дій. Контролер також забезпечує механізм маршрутизації запитів, тобто він визначає, які дії повинні бути виконані для кожного конкретного запиту користувача. Це дозволяє ефективно керувати різноманітними діями, які можуть виконуватися на вебсайті та забезпечує легкість розширення та підтримки системи.

Модель реалізує внутрішнє представлення даних та відповідає за їхнє зберігання та управління. Однією з ключових функцій моделі є забезпечення доступу до даних та їх обробка. Це включає в себе: зчитування, запис, оновлення та видалення даних з використанням відповідних операцій. Крім того, модель може містити логіку, пов'язану з бізнес-процесами, яка реалізується у вигляді правил обробки даних та виконання певних операцій. Основна мета моделі полягає у забезпеченні цілісності та безпеки даних, а також у виконанні важливих операцій з ними відповідно до вимог функціональності системи. Таким чином, модель є важливим елементом архітектури програмного забезпечення, що сприяє його ефективності та стабільності роботи.

Представлення відокремлений від бізнес-логіки та моделі даних і відповідає за відображення даних у вигляді, зрозумілому для користувача, інтерфейсу. Він може містити різноманітні елементи інтерфейсу, такі як: тексти, зображення, форми введення даних, кнопки та інші компоненти. Основним завданням виду є створення графічного інтерфейсу, який дозволяє користувачеві взаємодіяти з програмним забезпеченням. Тому вид відповідає за відображення даних, які надає модель, та взаємодію з користувачем через обробник подій, що забезпечує взаємодію з контролером. Таким чином, вид виступає важливим елементом архітектурного шаблону MVC, що дозволяє забезпечити високий рівень зручності та ефективності користувачам програмного забезпечення.

Но у системі використовується архітектурний шаблон, але дивлячись на певне концепції Model-View-Controller (MVC), архітектурний шаблон виявлятиме

									Арк.
									33
Змн.	Арк.	№ докум.	Підпис	Дата					

відмінності та матиме свої внутрішні особливості. Даний шаблон є ключовим елементом структури проєктів, розроблених з використанням даної системи, та відповідає за організацію взаємодії між компонентами, де контролер, модель та представлення можуть бути об'єднані у межах одного компонента чи модуля.

У контексті даної системи, контролер відповідає за обробку запитів та керування потоком вебзастосунка, модель відповідає за роботу з даними та бізнес-логікою, а представлення відповідає за відображення інформації на вебсторінках. Проте, відмінність полягає у тому, що ці компоненти часто не розділені на окремі сутності, а знаходяться у єдиному контексті, що спрощує взаємодію між ними та полегшує процес розробки.

Крім того, важливими аспектами є використання специфічних концепцій, таких як шаблони сайтів та компоненти, які забезпечують побудову вебсторінок та функціональності відповідно до потреб конкретного проєкту. Ці конструкції, спільно з основним архітектурним шаблоном, формують основу для розробки вебпроєктів та забезпечують ефективну та масштабовану реалізацію функціональності та інтерфейсу користувача.

Отже, архітектурний шаблон не лише відтворює загальні принципи MVC, але також впроваджує свої унікальні підходи та концепції, що сприяють ефективності та гнучкості процесу розробки проєктів.

Тому дані компоненти тісно взаємодіють між собою, створюючи відповідну структуру вебсайту та забезпечуючи необхідний функціонал. Такий підхід дозволяє забезпечити ефективну та стабільну роботу системи у відповідності з вимогами та потребами користувачів.

2.2 Проєктування логічної моделі бази даних

Проєктування логічної моделі бази даних є ключовим етапом у розробці будь-якого вебсайту. Дане проєктування є важливою складовою процесу створення програмного забезпечення, оскільки структура бази даних визначає, як дані будуть зберігатися та використовуватися в системі [16]. Тому для вебсайту була

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

спроектована логічна модель бази даних (рисунок 2.2), що дозволить у майбутньому зберігати усю необхідну інформацію для основного функціоналу з яким будуть взаємодіяти користувачі постійно.

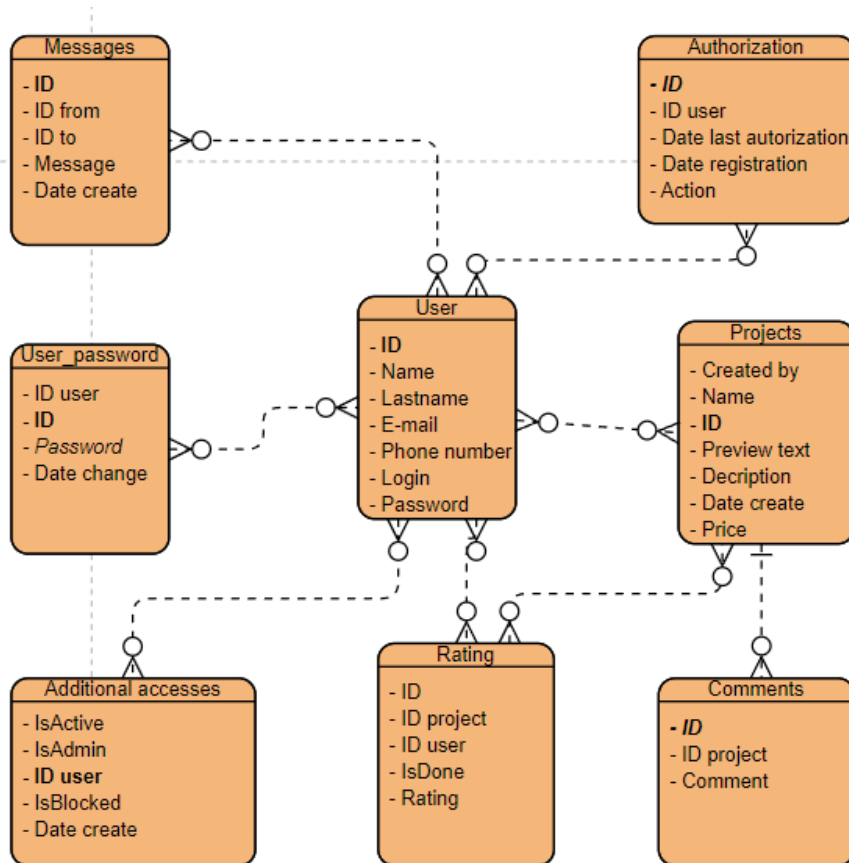


Рисунок 2.2 – Логічна модель бази даних

Сутність «User» включає в себе інформацію про зареєстрованих користувачів, таку як: їхні імена, паролі, адреси електронної пошти, номери телефонів, дати реєстрації, статус активності та додаткові доступи. «Messages» зберігає дані про повідомлення, включаючи ідентифікатори, відправників, одержувачів, текст та дату створення. «Projects» містить інформацію про проекти в системі, включаючи: їхні ідентифікатори, назви, описи, ціни, рейтинги та дати створення. «Additional accesses» зберігає дані про додаткові доступи до контенту. «Rating» зберігає дані надані користувачам для проектів, включаючи: ідентифікатор, ідентифікатори користувачів, ідентифікатори проектів, прапори

Змн.	Арк.	№ докум.	Підпис	Дата

виконання завдань, рейтинги роботи користувачі. «User password» зберігає в собі: ідентифікатор користувача, дату зміни пароля та пароль, який знаходиться окремо від основної таблиці для мір безпеки акаунту. Таблиця «Authorization» містить в собі інформацію про ідентифікатор користувачів та дату останнього відвідування, а також дату реєстрації та чи поточний користувач в даний момент знаходиться онлайн. Таблиця «Comments» містить в собі ідентифікатор користувача та контент, який користувач оставив на сторінці.

Детально потрібно виділити типів полів і їх розміри, та які саме дані зберігатимуть відповідатимуть до поля в базі даних.

Структура таблиці «User»:

- ID: INT - унікальний ідентифікатор користувача;
- Name: VARCHAR (255) - ім'я користувача;
- Lastname: VARCHAR (255) - прізвище користувача;
- Email: VARCHAR (255) - електронна пошта користувача;
- PhoneNumber: VARCHAR (20) - номер телефону;
- Login: VARCHAR (255) - логін користувача;
- Password: VARCHAR (255) - пароль користувача.

Структура таблиці «User_password»:

- ID: INT - унікальний ідентифікатор запису;
- ID_user: INT - ідентифікатор користувача;
- Password: VARCHAR (255) - зашифрований пароль користувача;
- Date_change: DATE - дата зміни паролю.

Структура таблиці «Projects»:

- ID: INT - унікальний ідентифікатор проєкту;
- Name: VARCHAR (255) - назва проєкту;
- Description: TEXT - опис проєкту;
- Preview_text: VARCHAR (255) - текст попереднього перегляду проєкту;
- Date_create: DATE - дата створення проєкту;
- Price: DECIMAL (10,2) - вартість проєкту;

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

– Created_by: INT - ідентифікатор користувача, який створив проєкт.

Структура таблиці «Additional_accesses»:

– IsActive: BOOLEAN - прапорець активності додаткового доступу;

– IsAdmin: BOOLEAN - прапорець адміністраторських прав;

– ID_user: BOOLEAN - ідентифікатор користувача;

– IsBlocked: BOOLEAN - прапорець блокування доступу;

– Date_create: DATE - дата створення додаткового доступу.

Структура таблиці «Authorization»:

– ID: INT - унікальний ідентифікатор запису;

– ID_user: INT - ідентифікатор користувача;

– Date_last_authorization: DATE - дата останньої авторизації користувача;

– Date_registration: DATE - дата реєстрації користувача;

– Action: BOOLEAN - прапорець активності користувача;

Структура таблиці «Comments»:

– ID: INT - унікальний ідентифікатор коментаря;

– ID_project: INT - ідентифікатор проєкту, до якого відноситься коментар;

– Comment: TEXT - текст коментаря.

Структура таблиці «Messages»:

– ID: INT - унікальний ідентифікатор повідомлення;

– ID_from: INT - ідентифікатор користувача, який відправляє повідомлення;

– ID_to: INT - ідентифікатор користувача, якому призначене повідомлення;

– Message: TEXT - текст повідомлення;

– Date_created: DATETIME - дата та час створення повідомлення.

Структура таблиці «Rating»:

– ID: INT - унікальний ідентифікатор оцінки;

– ID_project: INT - ідентифікатор проєкту, до якого відноситься оцінка;

– ID_user: INT - ідентифікатор користувача, який залишив оцінку;

– IsDone: BOOLEAN - прапорець, який позначає, чи завершено оцінювання;

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

– Rating: INT - оцінка проєкту.

Основна мета спроектованої бази даних полягає в збереженні даних, необхідних для відслідковування діяльності користувачів та можливості проведення майбутньої аналітики. Однак для успішного аналізу на основі накопичених даних важливо забезпечити ефективне управління великим обсягом інформації. Чим більше дані зберігаються, тим складніше їх керувати, і час виконання запитів, може значно збільшитись [17].

У зв'язку з цим в майбутньому може виникнути необхідність у розширенні бази даних або розгляді нових стратегій управління даними, щоб запобігти перевантаженню та підвищенню продуктивності системи. Важливо, щоб будь-які зміни та розширення не негативно впливали на роботу вже існуючих таблиць та функціоналу системи. Тому враховуючи потенційні труднощі, при проєктуванні бази даних була урахована можливість майбутнього масштабування та оптимізації для забезпечення ефективної роботи системи в будь-який момент [18].

Дану базу даних буде не важко реалізувати для проєктів різної складності, оскільки дана модель є універсальною і підходить під більшість ситуацій, що дозволяє зменшити потрібність розширювати базу даних у майбутньому. Якщо все ж буде розширення бази даних у майбутньому, то розширити базу даних буде нескладно через те, що дані при створенні нових таблиць не будуть заважати один одному при використанні полів зі старих таблиць, пов'язуючи їх з новими полями або таблицями, які нестимуть мету розширити базу даних.

Основні переваги цієї логічної моделі даних полягає у її простоті, чіткій структурі, масштабованості та гнучкості у майбутньому розширенні проєкту. База даних забезпечує ефективне і оптимізоване зберігання та обробку даних, що є ключовим чинником для продуктивності вебсайту.

У підсумку, дана логічна модель даних є ефективною основою для системи керування користувачами, повідомленнями, проєктами та додатковими доступами для окремих ролей.

					КВРІПЗ. 200175.01.22.00	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3 Проєктування інтерфейсу користувача

Під час проєктування інтерфейсу користувача для вебсайту біржі фрілансу враховуються унікальні потреби та очікування користувачів у цій специфічній сфері. Основна мета полягає в створенні зручного та ефективного інтерфейсу, що сприяє продуктивній взаємодії між замовниками та виконавцями фріланс-проєктів на біржі.

Аналіз потреб користувачів полягає у визначенні основних завдань, які вони будуть виконувати на вебсайті. До цього входить створення та пошук проєктів, взаємодія з іншими користувачами, управління проєктами та спілкування.

Зручність використання інтерфейсу є критичними аспектами для біржі фрілансу. Важливо забезпечити легкий доступ до всіх функцій, а також інтуїтивно зрозумілу навігацію та простоту використання.

Створення прототипів дозволяє визначити оптимальну структуру і розташування елементів інтерфейсу для забезпечення зручності навігації та доступу до функцій [19]. Це включає в себе розміщення кнопок для створення нових проєктів, списків для перегляду доступних завдань, а також можливості спілкування та обміну інформацією між користувачами.

Тому для вебсайту було розроблений концепт інтерфейсу за допомогою візуального редактора, який доволі сучасний та простий у використанні [20].

Наприклад, форма реєстрації (рисунок 2.3) спеціально розроблена з урахуванням простоти сприйняття користувачем, що вперше відвідує вебсайт та намагається зареєструватись. Форма містить мінімум обов'язкових полів і простий та зрозумілий інтерфейс, що дозволяє користувачам легко введення необхідної інформації. Такий підхід сприяє зменшенню бар'єрів для реєстрації та забезпечує позитивне перше враження під час взаємодії з вебсайтом.

У результаті, розроблена форма реєстрації представляє собою важливий елемент позитивного користувацького досвіду на вебсайті. Її спрощений та

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

доступний інтерфейс сприяє зручності та швидкості процесу реєстрації для нових користувачів даного сайту.

Рисунок 2.3 – Форма реєстрації

Використання форми листування (рисунок 2.4) у контексті вебсайту біржі фрілансу виявляється засобом забезпечення зручного та ефективного спілкування між користувачами. Цей інструмент безпосередньо на платформі, створює оптимальне середовище для обміну повідомленнями, дозволяючи здійснювати взаємодію через текстові комунікації. Форма листування має важливе значення для забезпечення комфортного обміну інформацією та сприяє створенню ефективної спільної роботи між учасниками платформи. Підтримка вкладень в повідомленнях розширює можливості обміну даними, надаючи користувачам зручний механізм для передачі файлів, документів або мультимедійних матеріалів. Це сприяє розвитку комунікаційного процесу, забезпечуючи швидку та ефективну передачу ресурсів між учасниками. Додатково, функціонал перегляду історії повідомлень дозволить зберігати важливі діалоги та легко отримувати доступ до попередніх

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

обмінів для подальшого аналізу. Це створює можливість для збереження важливої інформації та забезпечує зручний механізм для відстеження ходу обговорень.

Тому, вигляд структури відкритого листа від користувача, який надсилав повідомлення, допомагає зрозуміти особливості внутрішньої організації системи листування. Такий підхід підтримує зручну та зрозумілу взаємодію між учасниками платформи, що є ключовим аспектом забезпечення ефективного функціонування вебсайту біржі фрілансу з точки зору розробки інтерфейсу.

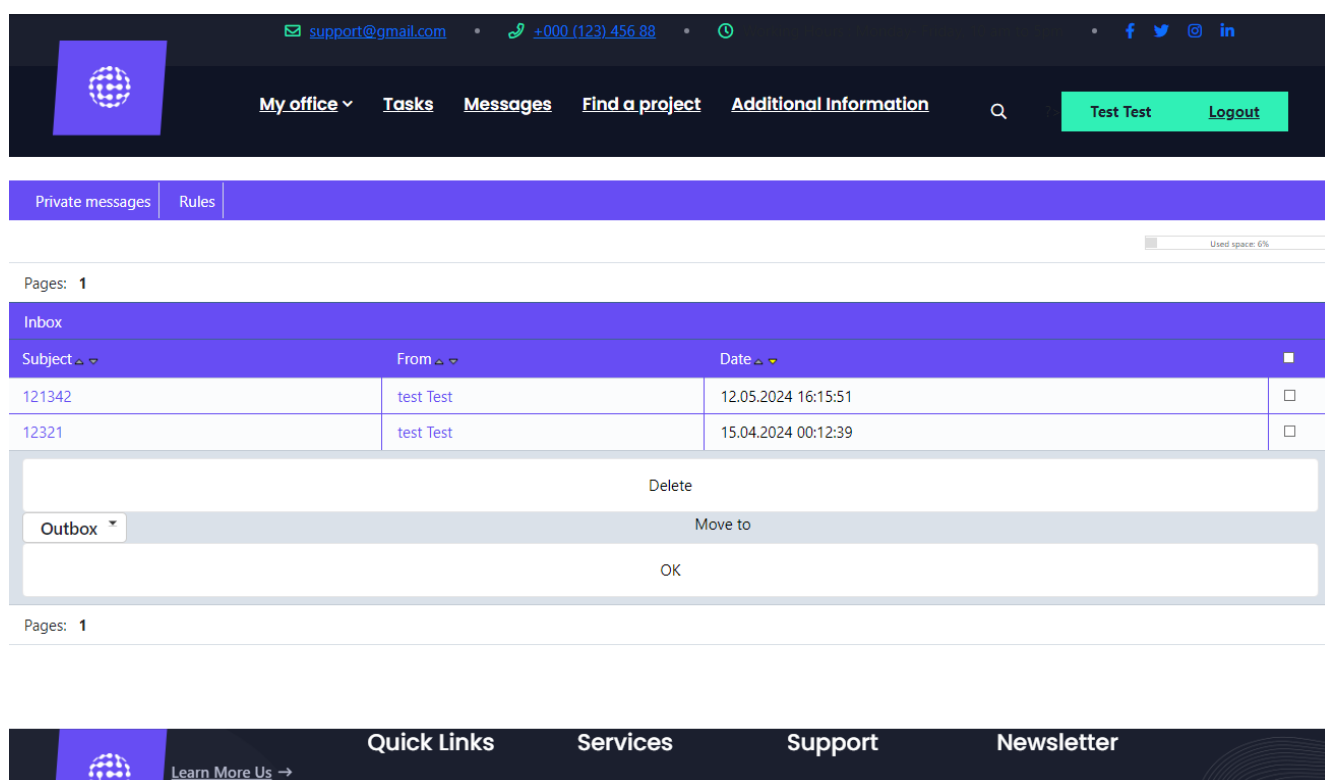


Рисунок 2.4 – Форма листування

До важливих складових ефективної комунікаційної платформи належить також форма контактних даних (рисунок 2.5), що створює можливість прямого зв'язку між користувачем та власником вебресурсу. Ця форма надає можливість користувачам надіслати повідомлення адміністрації вебсайту з будь-яких питань, пов'язаних з його функціонуванням, включаючи технічні проблеми, запити на підтримку або ж пропозиції щодо вдосконалення сервісу. Завдяки формі контактних даних користувачі можуть висловлювати свої побажання, виносити

пропозиції та спілкуватися з адміністрацією вебсайту безпосередньо, що створює відчуття взаємного співробітництва та допомагає зберігати позитивне співвідношення між всіма учасниками платформи. Такий механізм зв'язку є важливим елементом створення дружнього та доступного середовища для користувачів, де кожен може вільно висловлювати свої думки та потреби.

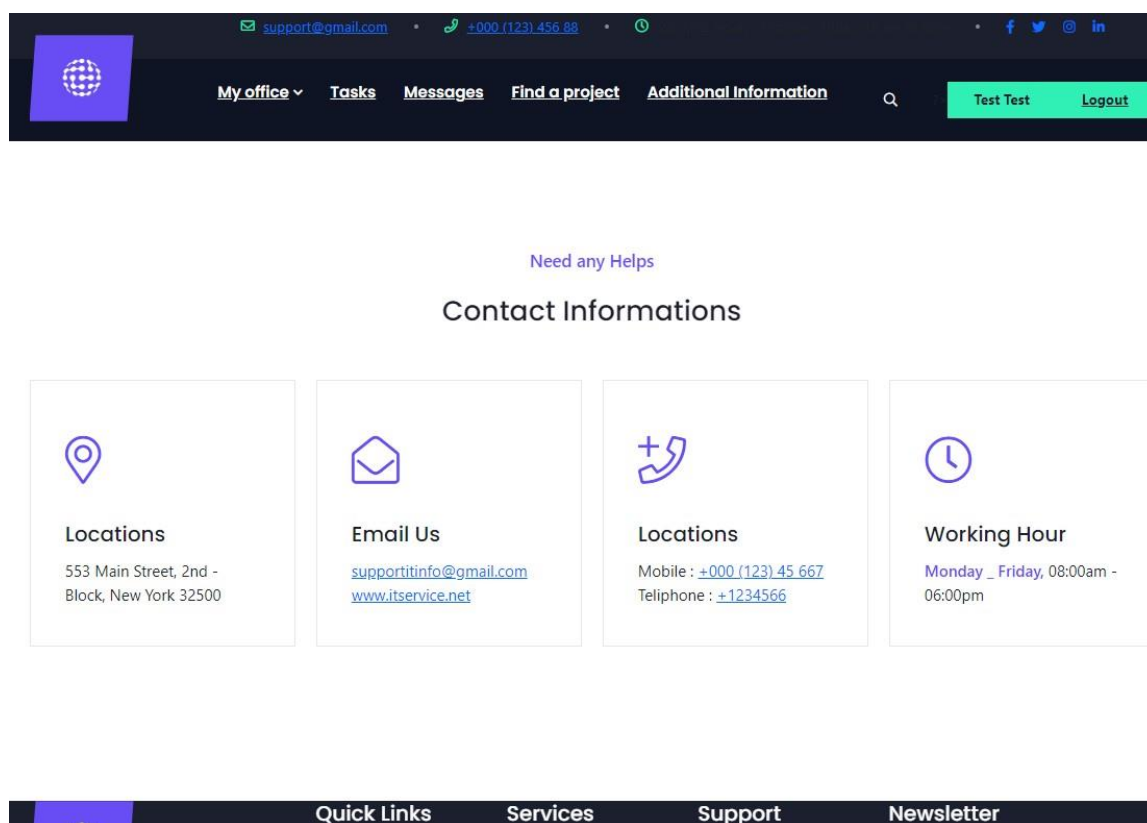


Рисунок 2.5 – Форма контактних даних

Одна з важливих складових є вигляд вебсайту на екрані телефону, тому що більшість користувачів навідують вебресурси за допомогою мобільного пристрою, тому що так зручніше і швидко в будь-якому місці де є вільний доступ до мережі інтернету [21].

Тому адаптивність сайту до екранів мобільних пристроїв (рисунок 2.6) є важливою складовою сучасних проєктів, враховуючи зростаючу популярність використання мобільного пристрою серед користувачів. Враховуючи обмежену площу екрану та особливості сенсорного інтерфейсу мобільних пристроїв,

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

адаптивний дизайн забезпечує оптимальне відображення контенту та його легке взаємодію на різних типах пристроїв.

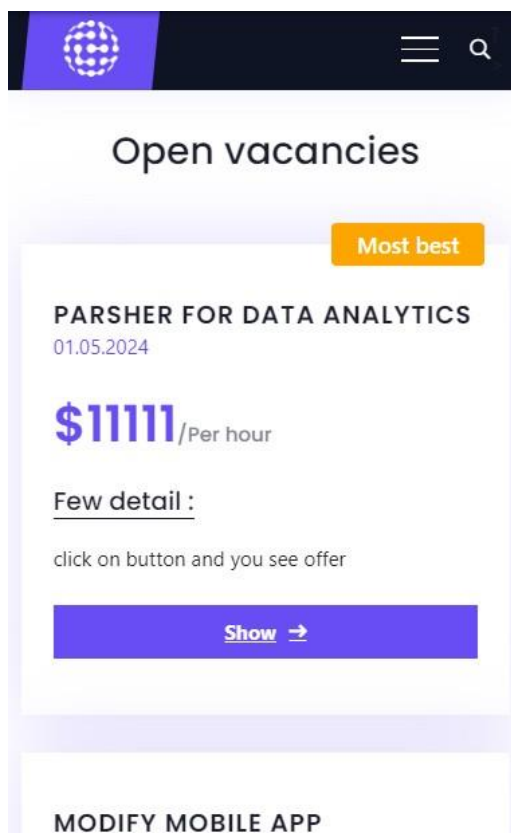


Рисунок 2.6 – Форма мобільного варіанту вебсайту

Метою адаптивного дизайну є створення вебсайту, який максимально комфортно відображається на будь-якому пристрої, незалежно від розміру та технічних характеристик екрану. Це досягається шляхом використання гнучких макетів та інших технік, що дозволяють адаптувати розміщення та вигляд елементів сторінки залежно від характеристик пристрою. Адаптивність вебсайту до мобільних пристроїв має ряд переваг, зокрема поліпшення користувацького досвіду та збільшення зручності взаємодії з контентом [22].

Отже, адаптивний дизайн вебсайту на екрані мобільного пристрою вважається ключовим аспектом розвитку інтерфейсу. Реалізація подібного інтерфейсу сприяє поліпшенню доступності та користувацького досвіду, забезпечуючи оптимальне відображення контенту на різних пристроях та екранах.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Адаптивний дизайн дозволяє автоматично пристосовувати вебсайт до розміру екрану, що відображається, що сприяє зручності взаємодії з сайтом навіть на пристроях з невеликими екранами.

Даний метод розробки інтерфейсу підвищує не лише комфорт користувачів у використанні сайту, але й сприяє підвищенню ефективності та конкурентоспроможності проєкту. Користувачі, які отримують приємний досвід користування сайтом на мобільних пристроях, користувачі схильні більше залишатися на сайті та повертатися до нього у майбутньому. Таким чином, адаптивний дизайн відіграє важливу роль у залученні та утриманні аудиторії, а також у підвищенні популярності сайту.

2.4 Аналіз та вибір технологій і методів реалізації

При розробці вебсайту біржі фрілансу вибір мов програмування та технологій для реалізації проєкту має велике значення для досягнення поставлених цілей щодо функціональності, продуктивності та зовнішнього вигляду. Однією з ключових рішень є вибір мови програмування, яка найкращим чином відповідає потребам проєкту, забезпечуючи ефективне виконання функцій та оптимальне використання ресурсів. Також важливо обрати відповідні технології, що допоможуть реалізувати необхідні функціональні можливості та забезпечити зручний та привабливий користувацький інтерфейс.

Для реалізації сайту було обрано такі мови як:

- PHP (Personal Home Page);
- HTML (HyperText Markup Language);
- CSS (Cascading Style Sheets);
- JavaScript;
- MySQL.

PHP – це популярна універсальна скриптова мова, яка особливо добре підходить для розробки вебсайту [23]. Мова швидка, гнучка та прагматична, PHP

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

також забезпечує роботу всього, від простого блогу до найпопулярніших вебсайтів створених у світі.

Для реалізації проєкту було обрано мову програмування PHP. Вибір мови програмування PHP для розробки проєкту було зумовлено її гнучкістю та можливістю масштабування, що ідеально підходить для створення платформи онлайн-біржі фрілансерів. Простота використання, широке поширення, гнучкість, швидкість, масштабованість та безпека роблять PHP оптимальним вибором для даного проєкту.

Обрана мова програмування PHP відома своєю популярністю і відмінною підтримкою, що робить її ідеальним інструментом для створення та розвитку вебсайтів у майбутньому. Завдяки активній спільноті розробників і широкому спектру готових рішень у вигляді бібліотек, фреймворків та розширень, розробка нових функцій та розширень може відбуватися з високою швидкістю та ефективністю виконання коду. Також PHP славиться своєю простотою використання, широким застосуванням, гнучкістю, швидкістю, масштабованістю та вбудованими засобами безпеки, що робить її ідеальним вибором для подібного проєкту. Зважаючи на специфіку функціоналу та ролей на вебсайті біржі фрілансу, PHP виявляється найбільш оптимальним інструментом для виконання цих завдань. Цей інструмент для використання, сприятиме успішній розробці та ефективному функціонуванню платформи, забезпечуючи надійну та безпечну роботу для користувачів [24].

Таким чином, використання мови програмування PHP є виправданим рішенням для реалізації проєкту вебсайту біржі фрілансу на стороні back-end розробки, оскільки мова володіє усім необхідним спектром інструментів для успішної розробки та експлуатації такого типу вебсайту.

Разом з використанням PHP буде застосовуватись Vitrix Framework, тому що цей фреймворк надає широкі можливості для швидкого та ефективного розроблення вебзастосунків. Також фреймворк має велику кількість вбудованих

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

функцій, які дозволяють легко керувати контентом, забезпечувати безпеку, робити адміністрування та моніторинг системи.

Bitrix Framework забезпечує високу швидкодію та масштабованість, що робить його ідеальним вибором для розробки вебзастосунків будь-якої складності. Цей фреймворк відмінно підходить для великих проєктів, де потрібно обробляти великі обсяги даних та взаємодіяти з великою кількістю користувачів одночасно, які відсилають велику кількість запитів.

Крім того, Bitrix Framework має великий ряд тестів для створених модулів, що дозволяє відслідковувати справність роботи вебсайту та переконатися в надійності розробленого програмного забезпечення. Це робить його дуже привабливим вибором для комерційних проєктів, де надійність та безпека є ключовими аспектами.

HTML – це стандартна мова розмітки, яка використовується для створення структури вебсторінок. Вона надає можливість організувати та структурувати контент на сторінці, включаючи заголовки, параграфи, списки, таблиці та інші елементи. Використання HTML дозволить створити логічну та зрозумілу структуру вебсторінки, що сприятиме зручності навігації користувачів та поліпшить їх взаємодію з сайтом [25].

CSS – це мова опису стилів, яка використовується для оформлення та стилізації сторінок. Вона дозволяє розробникам визначати зовнішній вигляд елементів HTML, включаючи кольори, шрифти, розміри, відступи та інші властивості. Використання CSS дозволяє забезпечити зовнішній вигляд на усьому сайті, що робить його більш привабливим, що може вплинути на сприйняття сервісу користувачем [26].

Разом HTML та CSS дають можливість створення ефективного та зручного користувацького інтерфейсу на вебсайті біржі фрілансу. Тому дані мови дозволяють легко створювати та керувати структурою та виглядом вебсторінки, що робить сайт більш привабливим та зручним для користувачів.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

Вибір JavaScript для сайту біржі фрілансу є стандартним рішенням в розробці сайту, оскільки ця мова програмування забезпечує широкий функціонал та можливості для створення динамічних та інтерактивних елементів вебсайту.

JavaScript – мова програмування, яка використовується для створення вебзастосунків, які взаємодіють з користувачем та маніпулюють веб-сторінками. Вона дозволяє розробникам додавати різноманітні функціональність, таку як анімація, валідація форм, взаємодія з сервером за допомогою AJAX [27].

Використання JavaScript дозволяє створювати динамічний та інтерактивний користувацький інтерфейс, який покращує взаємодію користувачів з сайтом біржі фрілансу [28]. Крім того, JavaScript є стандартом веброзробки і підтримується всіма сучасними браузерами, що робить його ідеальним вибором для реалізації вебзастосунків будь-якого рівня складності [29].

Таким чином, вибір JavaScript для вебсайту біржі фрілансу є доцільним, тому дана мова програмування дозволить виконати доволі не мало функціоналу на стороні front-end розробки.

Обрання MySQL для реалізації бази даних у проєкті обумовлене декількома факторами, які забезпечують його переваги в контексті розробки сайту.

Однією з причин вибору MySQL для реалізації бази даних у проєкті є визнаною популярністю та надійністю серед реляційних систем керування базами даних. Дана система славиться своєю швидкодією, надійністю і розширеним набором функцій, що дозволяє зручно та ефективно обробляти велику кількість даних.

Крім того, MySQL є безкоштовним та має відкритий вихідний код, що робить базу даних доступною для всіх. Це дозволяє зменшити витрати на розробку проєкту, а також забезпечує гнучкість та можливість розширення у майбутньому без значних інвестицій.

Ще однією перевагою MySQL є його широке співпрацю з іншими технологіями, такими як PHP, який також був обраний для цього проєкту. Ця

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

сумісність дозволяє легко і ефективно взаємодіяти між сервером, базою даних та сприяючи швидкому розгортанню та оптимізації роботи вебсайту [30].

Отже, обрання MySQL обгрунтоване його надійністю, швидкістю, безкоштовністю та сумісністю з іншими технологіями, що робить його оптимальним вибором для реалізації бази даних у вебпроектів.

Також потрібно врахувати, що сервер має виконувати поставлені задачі за допомогою операційної системи, тому при виборі операційної системи для сервера необхідно враховувати ряд ключових факторів, серед яких ефективність, надійність та мінімальні витрати на підтримку. Операційна система CentOS 7 відповідає поставленим вимогам. Головною перевагою CentOS 7 є стабільність та надійність, які забезпечують безперебійну роботу сервера протягом тривалого періоду часу. Використання CentOS 7 також гарантує сумісність з великою кількістю програмного забезпечення та інструментів, що спрощує процес налаштування та підтримки сервера. Одним з важливих аспектів вибору CentOS 7 є його оптимізація для використання ресурсів апаратного забезпечення сервера. Операційна система CentOS 7 забезпечує ефективне використання обчислювальної потужності, що дозволяє серверу працювати з високою продуктивністю без помилок [31].

Завдяки цим перевагам, вибір операційної системи CentOS 7 для нашого сервера дозволить забезпечити оптимальну продуктивність та ефективність роботи системи, зменшуючи витрати на підтримку та забезпечуючи стабільну роботу вебсервера.

Використані технології PHP, HTML, CSS, JavaScript і MySQL – відіграють ключову роль у процесі розробки програмного забезпечення, зокрема вебсайтів. Можливості даних інструментів дозволяють створювати динамічні та естетичні вебсайти, забезпечуючи високий рівень інтерактивності та збереження великого обсягу даних. Тому даний обсяг технологій надає можливості для реалізації різноманітних проєктів, дозволяючи створювати продуктивні та ефективні програмні рішення.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

2.5 Висновки проєктування програмного забезпечення

Отже, на основі проведеного аналізу та проєктування архітектури та структури системи можна зробити висновок, що розроблена система готова відповідати вимогам до функціональності та продуктивності. Оптимально побудована архітектура дозволяє забезпечити ефективну роботу системи та з легкістю розширювати її в майбутньому, що створює стійку основу для подальшого розвитку проєкту.

Проєктування архітектури та структури системи було здійснене з урахуванням вимог до функціональності та продуктивності системи. Оптимально побудована архітектура дозволяє забезпечити ефективну роботу системи та легкість її розширення в майбутньому.

Під час розробки логічної моделі бази даних були уважно враховані всі вимоги до зберігання та обробки даних, що дозволило створити чітку структуру для ефективного управління інформацією та сприяє легкості взаємодії з майбутньою базою даних.

Аналіз та вибір технологій та методів реалізації був здійснений з урахуванням вимог проєкту та сучасних тенденцій у сфері розробки програмного забезпечення. Вибрані технології відповідають потребам проєкту та забезпечують його успішної реалізацію в подальшому.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Програмна реалізація модулів

У процесі створення програмного застосунку для біржі фрілансу, ключовим етапом є реалізація функціональності модулів. Цей етап передбачає написання програмного коду, який забезпечить необхідні можливості для ефективної роботи вебсайту. Структурування проєкту за допомогою розміщення файлів у відповідних папках сприяє створенню зрозумілої та організованої архітектури. Кожна папка відповідає певній функціональній групі, що спрощує навігацію та взаємодію з файлами та елементами коду.

Для реалізації проєкту від час виконання використовувались стандарти мови програмування PHP [32]. Тому відображення сторінок використано варіант розділення сторінки на три частини, а саме на header.php, footer.php та index.php. Тому header.php відповідає за верхню частину вебсайту, яка пов'язана з основною навігацією та іншими компонентами, а footer.php відповідає за нижню частину шаблону, а точніше за додаткову інформацію або навігацію, а також за можливість розміщення різних банерів та реклами у майбутньому за необхідності. Верхня і нижня частина динамічно досить не часто змінюється, тому такий спосіб покращить завантаження контенту на вебсайті, що дозволить пришвидшити виконання запитів. Для прикладу виклику шаблону трішки нижче відображений код реалізації:

```
<?php
require($_SERVER["DOCUMENT_ROOT"]."/bitrix/header.php");
$APPLICATION->SetTitle("Freelance");
?>
<br>
<!--input cade index.php -->
<br>
<?php
require($_SERVER["DOCUMENT_ROOT"]."/bitrix/footer.php");?>
```

Для детальнішого прикладу нижче наведений код виклику сторінки реєстрації користувача:

									Арк.
									50
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

```

<?php
require($_SERVER["DOCUMENT_ROOT"]."/bitrix/header.php");
$APPLICATION->SetTitle("Registration");
$APPLICATION->IncludeComponent(
"bitrix:main.register",
"",
Array(
"AUTH" => "Y",
"REQUIRED_FIELDS" => array("EMAIL","NAME"),
"SET_TITLE" => "Y",
"SHOW_FIELDS" => array("EMAIL","NAME","PERSONAL_PHONE"),
"SUCCESS_PAGE" => "",
"USER_PROPERTY" => array(),
"USER_PROPERTY_NAME" => "",
"USE_BACKURL" => "Y"
));
require($_SERVER["DOCUMENT_ROOT"]."/bitrix/footer.php");?>

```

Наведений уривок коду виконує функцію налаштування сторінки реєстрації на вебсайті. Призначення цього коду полягає у створенні інтерфейсу для користувачів, щоб вони могли здійснити реєстрацію на даному вебсайті.

Першим кроком є включення файлу заголовку сторінки та встановлення заголовка сторінки, як «Registration». Подальший код викликає компонент, який відповідає за відображення форми реєстрації. Цей компонент налаштовується з різними параметрами, такими як: встановлення обов'язкових для заповнення полів, визначення сторінки успішного завершення реєстрації, та інші налаштування, спрямовані на зручне використання форми користувачем.

Увесь код реалізації знаходиться в «Додатку Б», через значні розміри коду, через які цілісно не вийде розмістити код на сторінці з поясненням.

Наступним модулем є модуль відображення замовлення на сторінці. Модуль відображення замовлення на сторінці має на меті представити інформацію про різні замовлення у відповідному форматі на сторінці. Даний модуль є важливою складовою системи управління замовленнями, оскільки надає зручний спосіб перегляду та навігації по доступним замовленням.

Реалізація модуля для виведення замовлення:

```

<?if(!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true)die();
$this->setFrameMode(true);
?>

<div class="col-xl-4 col-md-6">
    <div class="pricing-plan-item style-three wow fadeInUp delay-0-2s">

```

										Арк.
										51
Змн.	Арк.	№ докум.	Підпис	Дата						

```

        <span class="badge">Most best</span>
        <?if($arParams["DISPLAY_NAME"]!="N" && $arResult["NAME"]):?>
            <?if(!$arParams["HIDE_LINK_WHEN_NO_DETAIL"] ||
($arResult["DETAIL_TEXT"] && $arResult["USER_HAVE_ACCESS"]):?>
                <h4 class="title"><b><?echo $arResult["NAME"]?></b></h4>
            <?else:??>
                <h4 class="title"><?echo $arResult["NAME"]?></h4>
            <?endif;??>
        <?endif;??>
        <?if($arParams["DISPLAY_DATE"]!="N" &&
$arResult["DISPLAY_ACTIVE_FROM"]):?>
            <span class="price-count"><?echo
$arResult["DISPLAY_ACTIVE_FROM"]?></span>
        <?endif;??>
        <span class="price"><?php echo
$arResult['PROPERTIES']['11']['VALUE']; ?></span>

        <h5>Few detail :</h5><br>
        <?if($arParams["DISPLAY_PREVIEW_TEXT"]!="N" &&
$arResult["PREVIEW_TEXT"]):?>
            <?echo $arResult["PREVIEW_TEXT"];?>
        <?endif;??>

        <a href="<?echo $arResult["DETAIL_PAGE_URL"]?>" class="theme-btn
style-two">Show<i class="fas fa-long-arrow-right"></i></a>
    </div>
</div>

```

Також для реалізації верхнього меню навігації був розроблений модуль, який дозволяє без проблем розмістити навігаційне меню у верхній частині шаблону.

Реалізація навігаційного меню:

```

<div class="navbar-collapse collapse clearfix">
    <ul class="navigation clearfix">
        <? $previousLevel = 0;
        foreach($arResult as $arResult):?>
            <?if ($previousLevel && $arResult["DEPTH_LEVEL"] <
$previousLevel):?>
                <?=str_repeat("</ul></li>", ($previousLevel -
$arResult["DEPTH_LEVEL"]));?>
            <?endif;??>
            <?if ($arResult["IS_PARENT"]):?>
                <?if ($arResult["DEPTH_LEVEL"] == 1):?>
                    <li class="dropdown"><a
href="<?=$arResult["LINK"]?>"><?=$arResult["TEXT"]?></a>
                        <ul>
                            <?else:??>
                                <li class="dropdown"><a href="<?=$arResult["LINK"]?>"
><?=$arResult["TEXT"]?></a></li>
                            <ul>
                                <?endif;??>
                                <?else:??>
                                    <?if ($arResult["PERMISSION"] > "D"):?>
                                        <?if ($arResult["DEPTH_LEVEL"] == 1):?>
                                            <li ><a
href="<?=$arResult["LINK"]?>"><?=$arResult["TEXT"]?></a></li>
                                        <?else:??>

```

									Арк.
									52
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

```

                <li ><a
href="<?=$arItem["LINK"]?>"><?=$arItem["TEXT"]?></a></li>
                <?endif?>
                <?endif?>
                <?endif?>
                <?previousLevel = $arItem["DEPTH_LEVEL"];?>
                <?endforeach?>

        </ul>
</div>

```

Також дане меню знаходиться в header, де реалізований виклик меню, яке повноцінно відображається на сторінці та функціонує.

Реалізація виклику меню в шаблоні:

```

<div class="nav-outer mx-auto clearfix">
<!-- Main Menu -->
<?APPLICATION->IncludeComponent(
    "bitrix:menu",
    "top_multi",
    array(
        "COMPONENT_TEMPLATE" => "top_multi",
        "ROOT_MENU_TYPE" => "top",
        "MENU_CACHE_TYPE" => "N",
        "MENU_CACHE_TIME" => "3600",
        "MENU_CACHE_USE_GROUPS" => "Y",
        "MENU_CACHE_GET_VARS" => array(
        ),
        "MAX_LEVEL" => "2",
        "CHILD_MENU_TYPE" => "left",
        "USE_EXT" => "N",
        "DELAY" => "N",
        "ALLOW_MULTI_SELECT" => "N"
    ),
    false
);?>
<!-- Main Menu End-->
</div>

```

Далі був реалізований модуль для відслідковування та виведення відповідного статусу фрагменти сторінки. Якщо користувач не авторизований, то елементи будуть відповідно від статусу відображатись для навігації до форми реєстрації або авторизації, але якщо користувач авторизований на вебсайті під час його сесії, то відобразиться ім'я та прізвище на місці елементів авторизації і реєстрації. Щоб відобразити, що користувач авторизований в даний момент виводяться його ініціали, але також навпроти ініціалів створюється елемент інтерфейсу, який надає можливість закінчити сесію завчасно і вийти з акаунту.

Фрагмент реалізації даного функціоналу з відстеженням статусу:

										Арк.
										53
Змн.	Арк.	№ докум.	Підпис	Дата						

```

if ($USER->IsAuthorized()) {
    $userName = $USER->GetFullName(); // Отримуємо повне ім'я користувача
    $logoutUrl = $APPLICATION->GetCurPageParam("logout=yes", array("logout"));
    $logoutUrlWithSessid = $APPLICATION-
>GetCurPageParam("logout=yes&sessid=".bitrix_sessid(), array("logout"));
    ?>
    <div class="menu-btns">
        <span class="theme-btn"><?php echo $userName; ?></span>
    </div>

    <div class="menu-btns">
        <a href="<?= $logoutUrlWithSessid; ?>" class="theme-btn">Logout</a>
    </div>
    <?php
        } else {
// Користувач не авторизований, виводимо кнопки Log in i Sing in
        ?>
    <div class="menu-btns">
        <a href="/auth/auth.php" class="theme-btn">Log in</a>
    </div>
    <div class="menu-btns">
        <a href="/auth/registration.php" class="theme-btn">Sing in</a>
    </div>
    <?php
    }
?>

```

Додатково до модуля відображення замовлення на сторінці, у системі також реалізовані модулі реєстрації та авторизації. Ці модулі відіграють важливу роль у взаємодії з користувачами, надаючи їм можливість швидко та безпечно створити особистий акаунт та авторизуватися в системі.

Модуль реєстрації забезпечує можливість новим користувачам створити акаунт, заповнивши необхідні поля з особистою інформацією. Він надає зручний інтерфейс для введення даних, а також перевіряє правильність заповнення обов'язкових полів та відповідність вимогам безпеки.

Модуль авторизації дозволяє зареєстрованим користувачам увійти до свого акаунту, використовуючи введені при реєстрації дані. Це забезпечує доступ до особистих налаштувань, історії замовлень та інших персоналізованих функцій, що доступні лише авторизованим користувачам.

Завдяки цим модулям, користувачам надається можливість легко та просто керувати своїм акаунтом, забезпечуючи безпеку та зручність у взаємодії з вебсайтом. Такий підхід допомагає залучити більше користувачів до системи.

									Арк.
									54
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

3.2 Розробка бази даних

Для реалізації бази даних використовувалась стандарти MySQL. База даних створюється за допомогою мови SQL, що дозволяє легко та просто організувати створення структури розробленої бази даних [33].

Тому для реалізації бази даних, за основу взято логічну модель бази даних (рисунок 2.2), але для роботи бази даних також використовується сервер Nginx, який забезпечує роботу бази даних та дозволяє використовувати базу даних, через айпі адресу сервера. Це допомагає підключити базу даних до любого сервісу при необхідності та забезпечити безпеку даних з використанням прав доступу до неї. Також такий підхід може забезпечити легку реалізацію відкритого API у майбутньому, через використання комплексу перевірок для запитів на отримання або внесення даних за допомогою акаунта, який відслідковуватиметься на основі погодження прав з використання відкритих можливостей використання даних. Ці дані потребуватимуть автоматизацію процесів для специфічних потреб користувачів, але слід зазначити, що особисті дані користувачів завжди закриті для доступу при всіх розкладах використання сервісу у майбутньому.

Тому для реалізації таблиці «User» використовується наступний SQL запит, який реалізовує таблицю:

```
CREATE TABLE User (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(255) NOT NULL,  
    Lastname VARCHAR(255) NOT NULL,  
    Email VARCHAR(255) UNIQUE NOT NULL,  
    PhoneNumber VARCHAR(20) NULL,  
    Login VARCHAR(255) NOT NULL,  
    Password VARCHAR(255) NOT NULL  
);
```

В наведеному запиті відображено реалізацію даної таблиці, яка міститиме необхідні дані для базового акаунту користувача. Також потрібно реалізувати

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

«User_password», що забезпечить надійніше зберігання пароля в зашифрованому виді в окремій таблиці.

Реалізація створення таблиці «User_password»:

```
CREATE TABLE User_password (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    ID_user INT NOT NULL,  
    Password VARCHAR(255) NOT NULL,  
    Date_change DATE NOT NULL,  
    FOREIGN KEY (ID_user) REFERENCES User(ID)  
);
```

У цьому запиті вже відображено не просто створення таблиці, але й також створення зв'язків з полями таблиці «User», що дозволяє забезпечити взаємозв'язок між даними у різних таблицях. Це забезпечує можливість ефективного використання даних та здійснення операцій з ними, які потребують спільного використання інформації з різних джерел. Встановлення зв'язків між таблицями дозволяє виконувати складні запити та отримувати необхідні дані шляхом звернення до пов'язаних таблиць за допомогою ключів. Такий підхід до бази даних допомагає забезпечити безпеку та цілісність інформації, а також зробити систему більш гнучкою та масштабованою.

Наступною таблицею для реалізаціє являється таблиця «Projects». Тому нижче наведений запит на створення таблиці:

```
CREATE TABLE Projects (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(255) NOT NULL,  
    Description TEXT,  
    Preview_text VARCHAR(255),  
    Date_create DATE NOT NULL,  
    Price DECIMAL(10,2) NOT NULL,  
    Created_by INT NOT NULL,  
    FOREIGN KEY (Created_by) REFERENCES User(ID)  
);
```

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

У наведеному фрагменті коду створено таблицю з прив'язкою до таблиці «User», а точніше до поля ID, що дасть можливість відслідковувати яким користувачем був створений проєкт.

Також нижче наведено створення таблиці «Additional_accesses»:

```
CREATE TABLE Additional_accesses (  
    IsActive BOOLEAN,  
    IsAdmin BOOLEAN,  
    ID_user BOOLEAN,  
    IsBlocked BOOLEAN,  
    Date_create DATE,  
    FOREIGN KEY (ID_user) REFERENCES User(ID)  
);
```

Дана таблиця дає можливість отримати користувачу розширені додаткові права для управління контентом на сайті. В залежності від даних в цій таблиці користувач матиме додаткові доступи на сайті, з самого початку у користувача лише самі базові права, які дозволяють користуватись лише дозволеним функціоналом для всіх зареєстрованих користувачів.

Наступною для реалізації є таблиця «Authorization»:

```
CREATE TABLE Authorization (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    ID_user INT NOT NULL,  
    Date_last_authorization DATE NOT NULL,  
    Date_registration DATE NOT NULL,  
    Action BOOLEAN,  
    FOREIGN KEY (ID_user) REFERENCES User(ID)  
);
```

Дана таблиця призначена для моніторингу активності користувача на сайті, що дозволить взнати чи користувач перебуває на вебсайті в даний момент чи в даний момент користувач не активний. Також таблиця зберігає дані про дату реєстрації на сайті та дату останнього відвідування.

										Арк.
										57
Змн.	Арк.	№ докум.	Підпис	Дата						

Також відбудеться реалізація таблиці «Comments», таблиця призначена для зберігання коментарів, пов'язаних з конкретним проектом. Таблиця має наступну створену структуру:

```
CREATE TABLE Comments (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    ID_project INT NOT NULL,  
    Comment TEXT NOT NULL,  
    FOREIGN KEY (ID_project) REFERENCES Projects(ID)  
);
```

Наступною таблицею для створення є «Messages», таблиця призначена для зберігання особистих повідомлень між користувачами. Дана таблиця має наступну структуру запити:

```
CREATE TABLE Messages (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    ID_from INT NOT NULL,  
    ID_to INT NOT NULL,  
    Message TEXT NOT NULL,  
    Date_created DATETIME NOT NULL,  
    FOREIGN KEY (ID_from) REFERENCES User(ID)  
    FOREIGN KEY (ID_to) REFERENCES User(ID)  
);
```

Остання таблиця для реалізації – «Rating», яка створена для зберігання оцінок проектів.

Останній запит на створення таблиці:

```
CREATE TABLE Rating (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    ID_project INT NOT NULL,  
    ID_user INT NOT NULL,  
    IsDone BOOLEAN NOT NULL,  
    Rating INT NOT NULL,  
    FOREIGN KEY (ID_user) REFERENCES User(ID)  
    FOREIGN KEY (ID_project) REFERENCES Projects(ID)  
);
```

									Арк.
									58
Змн.	Арк.	№ докум.	Підпис	Дата					

Тому результатом є створена база даних, яка була спроектована та розглянута в попередніх розділах.

3.3 Керівництво користувача

Реєстрація та вхід:

– реєстрація нового облікового запису: потрібно перейти на сторінку реєстрації та заповнити всі необхідні поля. Після успішної реєстрації відбудеться отримання доступу до всіх функцій сайту;

– вхід в обліковий запис: у разі наявності облікового запису просто введіть свій логін та пароль на сторінці входу.

Пошук та розміщення проєктів:

– пошук проєктів: використовуйте фільтри та категорії для знаходження цікавих проєктів. Також ви можете скористатися розширеним пошуком для точного відбору;

– розміщення проєкту: якщо є проєкт, скористайтесь функцією «Створити проєкт» та детально заповніть всі необхідні дані.

Інформація про комунікацію та обмін повідомленнями. Чат та повідомлення: взаємодіяти можна з іншими користувачами за допомогою вбудованої системи чату та обмінюйтеся повідомленнями для уточнення деталей проєктів.

Інформація про оцінка та відгуки. Оцінка та відгуки: після завершення проєкту ви маєте можливість залишити відгук про роботу фрілансера та оцінити його роботу.

Загальні рекомендації:

– безпека облікового запису: потрібно бути обережним та не розголошувати особисту інформацію, таку як паролі або фінансові дані, іншим користувачам;

– правдивість інформації: потрібно пам'ятати, що правдива інформація про проєкт допоможе залучити більше зацікавлених осіб;

					КВРІПЗ. 200175.01.22.00	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

– підтримка користувачів: у разі будь-яких питань потрібно звернутись до служби підтримки користувачів для отримання допомоги.

3.4 Вимоги до технічних та програмних засобів

Для встановлення вебсайту на сервер потрібно виконати мінімальні потреби або виконати рекомендовані в залежності від обсягу та характеристики запитів, які очікується, що ресурс буде обробляти, а також кількість користувачів, які відвідуватимуть сайт.

Мінімальні потреби для сервера:

- SSD диск розміром 128 гб;
- операційна система: CentOS 7;
- оперативна пам'ять розміром 2 гб;
- процесор 5ht Intel Core i3 четвертого покоління і вище;

Рекомендовані потреби для сервера:

- SSD диск розміром 480 гб;
- операційна система: CentOS 7;
- оперативна пам'ять розміром 4 гб та вище;
- процесор 11ht Intel Core i7 і вище;

Для сторони користувача поставлення відповідні мінімальні вимоги при умові, що пристрій користувача має підтримувати браузер, який буде відкривати вебресурс.

Мінімальні потреби для користувача:

- системний блок;
- монітор;
- клавіатура;
- маніпулятор (сенсора миш);
- операційна система: Windows 10 – 11 або інші системи;
- оперативна пам'ять розміром 2 гб;

									Арк.
									60
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

– процесор Intel.

3.5 Вибір та обґрунтування методів тестування додатку

Тестування програмного забезпечення у сучасному світі виступає, як необхідний та обов'язковий елемент в процесі розробки та підтримки програмних продуктів. Актуальність цього підходу зумовлена швидким технологічним прогресом, високою конкуренцією та постійними змінами вимог користувачів.

Тестування, як процес, має за мету виявлення та усунення помилок, недоліків та непередбачуваних витоків у функціональності програмного забезпечення на ранніх етапах розробки. Це дозволяє забезпечити високу якість та надійність програми, зменшуючи ймовірність виникнення проблем під час експлуатації та тестуванні [34].

Сучасні підходи до тестування включають в себе різноманітні методики та практики, такі як: автоматизоване тестування, тестування контрольованого середовища та використання спеціалізованих інструментів тестування [35].

Для тестування роботи вебсайта буде використовуватись віртуальна машина VMware Workstation. Даний підхід є поширеним у практиці для тестування та доволі відомий своєю ефективністю. Встановлення програмного забезпечення VMware Workstation дозволяє керувати віртуальними машинами [36].

Після запуску VMware Workstation створюється нова віртуальна машина, на яку буде встановлено операційну систему CentOS 7. Для цього обирається опція встановлення операційної системи з образу, після чого завантажується образ CentOS 7. Під час налаштування віртуальної машини обираються параметри, такі як: розмір пам'яті, кількість процесорів і обсяг диска, а також встановлюються мережеві налаштування для забезпечення з'єднання з мережею. Після завершення налаштування віртуальна машина запускається, і розпочинається процес встановлення CentOS 7. Після встановлення операційної системи налаштовується сервер, встановлюються та налаштовуються необхідні серверні компоненти, такі

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

як: вебсервер Nginx, база даних MySQL або PHP. Після цього на віртуальний сервер завантажується сайт, який потрібно протестувати. Після розгортання сайту проводиться тестування для перевірки його функціональності, продуктивності, безпеки та сумісності з різними браузерами [37].

Наступним методом є Unit-test, який допоможе протестувати програмне забезпечення. Він спрямований на перевірку окремих компонентів або модулів програми. Цей підхід передбачає ізольоване тестування кожного компонента з метою виявлення можливих помилок або неполадок у його роботі. Також Unit-test буде виконуватись на віртуальній машині VMware Workstation [38].

Основними характеристиками Unit-test є автоматизація процесу виконання тестів, повторна виконуваність, швидкість виконання та можливість виявлення помилок під час тестування [39].

До інших методів тестування програмного забезпечення також відноситься тестування системи в цілому, включаючи в себе усі складові програмного продукту. В Bitrix Framework, наприклад, передбачено можливість проведення комплексного тестування всієї системи, що дозволяє перевірити роботу сайту на наявність помилок та недоліків. Такі тести допомагають забезпечити високу якість та надійність сайту перед його впровадженням в експлуатацію.

У підсумку, тестування програмного забезпечення є важливим етапом у процесі розробки та підтримки програмних продуктів. Воно спрямоване на виявлення помилок та недоліків, забезпечуючи високу якість та надійність програми. Unit-test є одним з ефективних методів тестування, що дозволяє перевіряти окремі компоненти або модулі програми на коректність їх роботи. Використання віртуальних машин, таких як VMware Workstation, сприяє проведенню тестування в контрольованому середовищі, що забезпечує оптимальну якість та безпеку програмного продукту. Також, проведення комплексного тестування усієї системи в Bitrix Framework допомагає забезпечити безперебійну роботу вебсайту у реальних умовах експлуатації.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

Тому для тестування відбудеться не малий обсяг роботи. Такий підхід дозволить ефективно перевірити роботу сайту і виявити можливі проблеми або недоліки на ранніх етапах розробки.

3.6 Тестування додатка

У початкових етапах процесу була створена нова віртуальна машина в системі VMware Workstation з обраною операційною системою CentOS 7. Далі були налаштовані параметри віртуальної машини, включаючи: обсяг пам'яті, кількість процесорів, обсяг диска та мережеві налаштування [40]. Після цього був запуснений процес встановлення операційної системи, під час якого був вибраний відповідний образ та виконані інструкції на екрані для завершення процесу. Після встановлення CentOS 7 було налаштоване серверне середовище, включаючи встановлення та налаштування вебсервера Nginx, бази даних MySQL та PHP [41]. Це забезпечило створення середовища для роботи вебзастосунка. Наступним кроком було розгортання сайту (рисунк 3.1), який підлягав тестуванню, на віртуальній машині.

```
Bitrix UM appliance 7.5.3
*****: System Data :*****
Hostname: diplomserver
Kernel: 3.10.0-1160.95.1.el7.x86_64
OS: CentOS Linux release 7.9.2009 (Core)
Memory: 1882128 kB
CPU: AMD Ryzen 5 5500H with Radeon Graphics x1
*****: Interface ens33 :*****
MAC: 00:0c:29:37:29:23
IP4: 192.168.1.105
*****: Options :*****
bitrix url: http://192.168.1.105/
diplomserver login:
```

Рисунок 3.1 – Результат розгортання

Після успішного розгортання сайту було проведено тестування його функціональності, продуктивності, безпеки та сумісності з різними браузерами.

Наступним етапом в процесі тестування є реалізація Unit-test для виведення відкритих проєктів на вебсторінку. Даний тест спрямований на перевірку

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

правильності функціонування механізму виведення інформації про відкриті проекти на веб-сторінці, що включає в себе правильне форматування та відображення даних.

Реалізація Unit-test для виводу створених відкритих проектів на вебсторінку наведено нижче:

```
<?php
use PHPUnit\Framework\TestCase;

class PricingPlanItemTest extends TestCase
{
    public function testRenderPricingPlanItem()
    {
        // Arrange
        $arParams = [
            "DISPLAY_NAME" => "Y",
            "HIDE_LINK_WHEN_NO_DETAIL" => false,
            "DISPLAY_DATE" => "Y",
            "DISPLAY_PREVIEW_TEXT" => "Y"
        ];

        $arItem = [
            "NAME" => "Test Plan",
            "DISPLAY_ACTIVE_FROM" => "2024-04-28",
            "PROPERTIES" => [
                "11" => [
                    "VALUE" => "$10"
                ]
            ],
            "PREVIEW_TEXT" => "Test preview."
        ];
        ob_start();
    ?>
```

Після створення образу тестових даних та запуску початку тесту нижче наведено вже роботу тесту:

```
<div class="col-xl-4 col-md-6">
    <div class="pricing-plan-item style-three wow fadeInUp delay-0-2s">
```

										Арк.
										64
Змн.	Арк.	№ докум.	Підпис	Дата						

КВРІПЗ. 200175.01.22.00

```

<span class="badge">Most best</span>
<?php if ($arParams["DISPLAY_NAME"] != "N" && $arResult["NAME"]) : ?>
    <?php if (!$arParams["HIDE_LINK_WHEN_NO_DETAIL"] || ($arResult["DETAIL_TEXT"] && true)) : ?>
        <h4 class="title"><b><?php echo $arResult["NAME"] ?></b></h4>
    <?php else : ?>
        <h4 class="title"><?php echo $arResult["NAME"] ?></h4>
    <?php endif; ?>
<?php endif; ?>
<?php if ($arParams["DISPLAY_DATE"] != "N" && $arResult["DISPLAY_ACTIVE_FROM"]) : ?>
    <span class="price-count"><?php echo $arResult["DISPLAY_ACTIVE_FROM"] ?></span>
<?php endif ?>
<span class="price"><?php echo $arResult["PROPERTIES"]['11']['VALUE']; ?></span>
<h5>Few detail :</h5><br>
<?php if ($arParams["DISPLAY_PREVIEW_TEXT"] != "N" && $arResult["PREVIEW_TEXT"]) : ?>
    <?php echo $arResult["PREVIEW_TEXT"]; ?>
<?php endif; ?>
<a href="<?php echo $arResult["DETAIL_PAGE_URL"] ?>" class="theme-btn style-two">Show<i class="fas fa-
long-arrow-right"></i></a>
</div>
</div>
<?php
$output = ob_get_contents();
ob_end_clean();
}
}
?>

```

Після виконання на основі отриманого результату (рисунок 3.2), можна сказати, що тест був проведений успішно.

```

PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.                                                                    1 / 1 (100%)

Time: 00:00.046, Memory: 8.00 MB

OK (1 test, 4 assertions)

```

Рисунок 3.2 – Результати тесту

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

Таким чином, потрібно протестувати увесь основний функціонал сайту. Існують можливі сценарії функціонального тестування сайту, які наведено у таблиці 3.1. для відображення існуючого тестування.

Таблиця 3.1 – Тестові сценарії функціонального тестування

№	Функціональна вимога	Вихідні дані	Очікуваний результат
1	Авторизація в системі	Користувач вводить свій логін і пароль	При коректних даних користувач отримує доступ до акаунту, при неправильному вводі виведеться повідомлення про помилку
2	Реєстрація в системі	Користувач заповнює форму з даним	Якщо всі дані введено коректно, в систему додається новий користувач
3	Додання відкритого проєкту	Користувач заповнює форму з інформацією про проєкт	За умови коректного заповнення всіх полів додається новий проєкт в списку проєктів
4	Редагування проєкту	Користувач відкриває інформацію про проєкт та змінює частину інформації	За умови коректного введення змінюється інформація про проєкт
5	Оновлення стадії проєкту	Користувач обирає проєкту і визначає чи даний проєкт завершений	На графіку прогресу проєкту відображається доданий прогрес задачі в відсотковому форматі
6	Написання повідомлень	Користувач заповнює форму з інформацією для отримувача даних	Користувач відправляє дані, після чого користувач, якому адресувалось повідомлення буде передана структура повідомлення за його ідентифікатором
7	Фільтрація виведення інформації	Користувач вписує параметри по яким хоче вивести дані	Виводиться інформація яка відповідна обраному параметру

Таким чином, після успішного завершення тестування цих функцій можливо перейти до наступного етапу тестування.

Наступним етапом тестування є застосування Bitrix Framework для проведення тестування усієї системи сайту. Використання функціоналу тестування в межах цього фреймворку дозволяє ефективно перевірити працездатність різних модулів та компонентів в єдиній інтегрованій системі.

Під час цього тестування використовуються вбудовані засоби Bitrix Framework, які дозволяють автоматизувати процеси тестування та аналізу роботи

									Арк.
									66
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

сайту. Зокрема, використання функціоналу тестування усієї системи сайту дозволяє проводити інтеграційні тести, які допоможуть перевірити взаємодію різних модулів та компонентів між собою.

Під час виконання тестів за допомогою Vitrix Framework здійснюється перевірка роботи ключових функцій сайту, включаючи обробку запитів користувачів, відображення інформації на веб-сторінках, роботу форм зворотного зв'язку та інші аспекти функціональності сайту.

Тому перший проведеним тестом виступає тест головного функціоналу вебсайту (рисунок 3.3) для отриманні інформації функціональності системи.

General website features		
Required PHP Modules	✓ All required modules are installed.	?
PHP Required Parameters	✓ Settings are correct	?
Web server modules	✓ No conflicts.	?
Server Variables	✓ Correct	?
Session retention	✓ Success	?
UTF configuration parameters (mbstring and BX_UTF)	✓ Correct. The website runs in UTF encoding	?
Service scripts in the site root	✓ None	?
Using sockets	✓ Success	?
Use cron to run agents	✓ Success	?

Рисунок 3.3 – Результати тесту основної функціональності

На підставі наданих даних можна зробити висновок, що базовий функціонал системи функціонує стабільно і без порушень. Це вказує на те, що ключові елементи функціональності системи, які перевірялися під час тестування, працюють належним чином і відповідають очікуванням користувачів.

Наступним тестування буде тестування додаткових можливостей для вебсайту, які повинні бути присутні при розробці вебсайту на Vitrix Framework.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

Advanced features			
Database connection parameters	✓	Success	?
Redundant output in configuration files	✓	Success	?
Session retention without UserAgent	✓	Success	?
Website Parameters	✓	Correct	?
Stack size; pcre.recursion_limit	✓	Success	?
File upload	✓	Success	?
Upload files over 4MB	✓	Success	?
Upload file using php://input	✓	Success	?
POST requests with many parameters	✓	Success	?
Check for unsent messages	✓	Success	?
Local redirects (LocalRedirect function)	✓	Success	?
Using cache files	✓	Success	?
Access to update server	✓	Success	?
HTTP authorization	✓	Success	?
File creation and execution	✓	Success	?
getimagesize support for SWF	✓	Success	?

Рисунок 3.3 – Результати тесту додаткових можливостей

Під час тестування додаткових можливостей в рамках Vitrix не було виявлено жодних помилок. Це свідчить про те, що процес встановлення системи на віртуальну машину пройшов успішно і безперебійно. Усі компоненти та модулі, які використовуються в системі, встановлені та налаштовані коректно.

Виявлення відсутності помилок у додаткових можливостях Vitrix підтверджує правильність конфігурації віртуальної машини та належне функціонування всіх додаткових програмних компонентів. Це є важливим кроком у забезпеченні стабільної та ефективної роботи системи, оскільки додаткові можливості часто відображають важливі функціональні та можуть впливати на користувацький досвід.

Наступним етапом у послідовності тестування є перевірка коректності функціонування бази даних. Це тест крок, оскільки база даних є основою для зберігання та обробки інформації, яка використовується в системі. Тому для тестування було виконано тестування бази даних (рисунок 3.4) на помилки в роботі всієї системи.

Database test		
MySQL version	✓ Success	?
Database and web server times	✓ Success	?
MySQL Mode	✓ Success	?
Connection charset	✓ Success	?
Database charset	✓ Success	?
Database table charset	✓ Success	?
Database structure	✓ Success	?

Рисунок 3.4 – Результати тестування бази даних

Після проведення тестування виявилось, що база даних не має відхилень у виконанні процесів.

3.7 Аналіз результатів тестування

У початкових етапах процесу створення нової віртуальної машини у системі VMware Workstation з обраною операційною системою CentOS 7 було успішно завершено. Налаштування параметрів віртуальної машини, таких як: обсяг пам'яті, кількість процесорів, обсяг диска та мережеві налаштування, було здійснено відповідно до вимог. Під час процесу встановлення операційної системи CentOS 7 був вибраний відповідний образ та виконані відповідні інструкції для завершення процедури виконання.

Після успішного встановлення операційної системи на віртуальну машину, було налаштоване серверне середовище, включаючи встановлення та налаштування вебсервера Nginx, бази даних MySQL та PHP. Це дозволило створити середовище для подальшої роботи з веб-застосунком. Особлива увага була зосереджена на перевірці функціональності, продуктивності, безпеки та сумісності з різними браузерами.

У ході виконання тестування було успішно пройдено визначені сценарії. Для кожного із цих сценаріїв було розроблено відповідний набір PHPUnit тестів, які відтворювали умови та перевіряли отримані результати на відповідність очікуваному вихідному стану. Отримані під час функціонального тестування

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

результати, зазначені в попередньому розділі, представлені у Таблиці 3.2. Дана таблиця містить опис тестованої вимоги, введені дані для кожного тесту, отриманий результат та загальну оцінку його коректності.

Таблиця 3.2 – Тестові сценарії функціонального тестування

№	Функціональна вимога	Вхідні дані	Отриманий результат	Результат
1	Авторизація в системі	Логін і пароль	Користувач отримує доступ до акаунту або повідомлення про помилкову	Правильно
2	Реєстрація в системі	Дані полів користувача	Користувач отримує доступ до акаунту, який був створений на основі введених даних	Правильно
3	Додання відкритого проєкту	Дані полів проєкту	Проект доданий до бази даних	Правильно
4	Редагування проєкту	Заміна даних полів обраним користувачем	Заміна даних проєкту відповідно до відредагованих полів	Правильно
5	Оновлення стадії проєкту	Значення статусу проєкту	Заміна стуса проєкту	Правильно
6	Написання повідомлень	Дані з полів	Повідомлення заноситься до бази даних	Правильно
7	Фільтрація виведення інформації	Отримання даних з поля для параметрів	Виводиться інформація яка відповідає обраному параметру	Правильно

Аналізуючи процес тестування проєкту, слід відзначити, одним з головних етапів було використання Vitrix Framework для проведення комплексного тестування усієї системи сайту. Використання цього фреймворку дозволило ефективно перевірити різні модулі та компоненти в єдиній інтегрованій системі. Під час цього тестування були використані вбудовані засоби Vitrix Framework, що автоматизували процеси тестування та аналізу роботи сайту.

Під час перевірки головного функціоналу вебсайту, а також додаткових можливостей за допомогою Vitrix Framework, не було виявлено жодних помилок. Це свідчить про правильність конфігурації віртуальної машини та належне функціонування всіх компонентів та модулів системи. Зокрема, відсутність

помилку у додаткових можливостях Vitrix підтверджує стабільність та ефективність системи в цілому.

Наступним етапом у послідовності тестування було перевірка коректності функціонування бази даних. Результати цього тестування свідчать про відсутність відхилень у роботі бази даних, що є ключовим аспектом для забезпечення стабільності та надійності системи.

Отже, аналізуючи процес тестування проекту, можна зробити висновок про його успішність та стабільність у роботі, а також про відсутність помилок та відхилень у функціонуванні основних компонентів та модулів системи.

3.8 Висновки програмної реалізації та тестування

Програмна реалізація модулів була успішно проведена, що дозволило реалізувати необхідний функціонал вебзастосунку відповідно до вимог технічного завдання. Використання потужних інструментів розробки дозволило забезпечити якісний та ефективний код, який підтримується та розширюється з легкістю.

Розробка бази даних була проведена професійно та з урахуванням всіх вимог до зберігання та обробки інформації. Структура бази даних добре організована і дозволяє ефективно взаємодіяти з програмним забезпеченням, забезпечуючи оптимальну продуктивність та безпеку даних.

Керівництво користувача надає докладну та зрозумілу інформацію щодо використання вебзастосунку. Чіткі інструкції та пояснення дозволяють користувачам швидко ознайомитися з функціоналом та зручно користуватися системою.

Вимоги до технічних та програмних засобів були виконані в повному обсязі, що дозволило забезпечити стабільну та ефективну роботу вебзастосунку. Обрані технології та інструменти відповідають вимогам проекту та забезпечують його успішне функціонування.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

Вибір методів тестування був обґрунтований та здійснений з урахуванням специфіки проєкту. Використання різноманітних методів та інструментів дозволило забезпечити високу якість програмного забезпечення та виявити та усунути потенційні помилки та недоліки.

Тестування додатка було проведено на високому рівні і дозволило перевірити всі аспекти його роботи. Результати тестування свідчать про стабільність та надійність системи.

Аналіз результатів тестування підтвердив відповідність розробленого веб-застосунку вимогам та очікуванням. Виявлені недоліки та проблеми були виправлені, що дозволило забезпечити оптимальний рівень якості продукту.

					КВРІПЗ. 200175.01.22.00	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Метою даної кваліфікаційної роботи є ретельний аналіз технічних аспектів функціонування вебсайтів бірж фрілансу та визначення можливостей для покращення їх ефективності. Основна увага приділяється розкриттю різноманітних технічних аспектів інфраструктури таких платформ з метою надання корисної інформації для розробників, адміністраторів та користувачів щодо оптимізації та покращення їхньої роботи.

Задачами дослідження було наступне:

- провести аналіз предметної області, враховуючи особливості її реалізації та функціональні можливості;
- провести огляд існуючих рішень на ринку, визначити їх переваги, недоліки та мінімальний функціонал;
- розробити систему, яка буде відповідати вимогам та потребам користувачів у цій галузі.

Тому у роботі було проведено дослідження предметної області, а також постановку завдань. Проаналізовано наявне програмне забезпечення та визначено вимоги до даного вебсайту. Під час проєктування була обрана відповідна архітектура клієнт-серверна архітектура з використанням архітектурного шаблону MVC та спроектована логічна структура бази даних, яка потрібна була для реалізації бази даних для проєкту, розроблено інтерфейс користувача та вибрано технології для реалізації.

Наступним етапом була програмна реалізація вебсайту та його тестування. Були розроблені модулі вебсайту за допомогою обраних інструментів для реалізації проєкту, створена база даних за допомогою SQL запитів, додано керівництво користувача, яке допоможе користувачу орієнтуватись на вебсайті та вимоги до технічних засобів, що вказують потреби для забезпечення роботи сервера та програмного забезпечення на сервері.

									Арк.
									73
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

Після вибору методів тестування, було проведення тестів, що стали ключовим етапом в розробці нашого вебсайту біржі фрілансу. Використані методи тестування були націлені на ретельну перевірку всіх функціональних можливостей та аспектів вебсайту з метою виявлення потенційних помилок і недоліків. Отримані результати тестування підтвердили високу якість розробленого вебсайту та його готовність до використання в реальних умовах. Виявлені та виправлені помилки стали важливим етапом у забезпеченні безперебійної роботи вебсайту.

Розроблений вебсайт біржі фрілансу має дружній інтерфейс та відповідає потребам користувачів у даній галузі біржі фріланса. Сайт підтримує різноманітні функціональні можливості та відповідає системним вимогам.

Отримані результати демонструють, що розроблений сайт може відмінно виконувати поставлені задачі і функціональні вимоги користувачів. Такий підхід дозволить підтримувати високу ефективність та забезпечувати комфортні умови для користувачів, сприяючи подальшому успіху та конкурентоспроможності платформи на ринку бірж фрілансу.

У майбутньому варто розглянути різноманітні можливості розвитку вебсайту, спрямовані на підвищення його ефективності та зручності для користувачів. Однією з таких можливостей є використання системи дата-аналітики, що дозволить отримувати важливі дані про активність користувачів на вебсайті. Збираючи статистику виконання проєктів або виконаних завдань фрілансерами, можна буде здійснювати аналіз та вдосконалення робочого процесу, а також пропонувати користувачам більш ефективні інструменти для досягнення їх цілей.

Застосування штучного інтелекту для покращення проєкту в цьому контексті може бути особливо корисним. Наприклад, на основі зібраної аналітики та використання алгоритмів машинного навчання можна розробити систему, що автоматично формує резюме фрілансерів на основі їхньої діяльності на вебсайті. Це сприятиме підвищенню ефективності пошуку проєктів для фрілансерів та допоможе залучити нових клієнтів для замовників, які шукають кваліфікованих спеціалістів для виконання проєктів.

									Арк.
									74
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tinashe B. The Freelance Educator: Practical Advice for Starting your Educational Consulting Business. London : Routledge, 2022. 120 p.
2. Alexandra F. Freelance Your Way to Freedom: How to Free Yourself from the Corporate World and Build the Life of Your Dreams. Hoboken : Wiley, 2022. 272 p.
3. Хміль О.О, Праворська Н.І. Веб-сайт біржі фрілансу. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». Хмельницький, 2023. 304-307 с.
4. Офіційний сайт Freelancer [Електронний ресурс] – Режим доступу: <https://www.freelancer.com> (дата звернення 11.02.2024)
5. Tom A., Sophie C. The Freelancer's Business Brain: Work smart, win clients, make money (Freelance Writing Essentials). New York : ABC Business Communications Ltd, 2024. 341 p.
6. Tom M. Marketplace Best Practices: Transforming Commerce in the Platform Economy. Chicago : Independently published, 2021. 477 p.
7. Офіційний сайт Fiverr [Електронний ресурс] – Режим доступу: <https://www.fiverr.com> (дата звернення 11.02.2024)
8. Офіційний сайт Upwork [Електронний ресурс] – Режим доступу: <https://www.upwork.com> (дата звернення 11.02.2024)
9. Visual Paradigm [Електронний ресурс] – Режим доступу: <https://www.visual-paradigm.com> (дата звернення 16.02.2024)
10. Ashley P. Creating Software with Modern Diagramming Techniques: Build Better Software with Mermaid. North Carolina : Pragmatic Bookshelf, 2023. 158 p.
11. Mark R., Neal F. Fundamentals of Software Architecture: An Engineering Approach. Sebastopol : O'Reilly Media, 2020. 419 p.
12. John G. Software Architecture Patterns for Serverless Systems: Architecting for innovation with events, autonomous services, and micro frontends. Birmingham : Packt Publishing, 2021. 436 p.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

13. Nico L. Software Engineering for Absolute Beginners: Your Guide to Creating Software Products. New York : Apress, 2021. 360 p.

14. Matthew S. API's for the Complete Beginner: Understand API's at a foundational level (Information Technology for the Complete Beginner). Chicago : Independently published, 2023. 688 p.

15. Victor H. G. PHP MVC: Create your own MVC framework with PHP (and learn a lot in the process). Chicago : Independently published, 2023. 379 p.

16. Sikha S. B., Richard W. E. Database Design Using Entity-Relationship Diagrams (Foundations of Database Design). Boca Raton : Auerbach Publications, 2022. 388 p.

17. Carlos C., Steven M. Database Systems: Design, Implementation, & Management (MindTap Course List). Boston : Cengage Learning, 2022. 816 p.

18. Alex P. Database Internals: A Deep Dive into How Distributed Data Systems Work. Sebastopol : O'Reilly Media, 2019. 370 p.

19. Luca P. UI + UX: web design simply explained (Manuals for web designers). Chicago : Independently published, 2022. 182 p.

20. Figma. URL: <https://www.figma.com> (дата звернення: 28.02.2024).

21. Jenifer T., Charles B., Aynne V. Designing Interfaces: Patterns for Effective Interaction Design. Sebastopol : O'Reilly Media, 2020. 599 p.

22. Jason B., Alex W., James G. The Principles of Beautiful Web Design. Melbourne : SitePoint, 2020. 282 p.

23. Офіційний сайт PHP. URL: <https://www.php.net/manual/en/index.php> (дата звернення: 20.03.2024).

24. Eric M. PHP Cookbook: Modern Code Solutions for Professional Developers. Sebastopol : O'Reilly Media, 2023. 431 p.

25. Jürgen W. HTML and CSS: The Comprehensive Guide. Bonn : Rheinwerk Computing, 2023. 814 p.

26. Ben F. Responsive Web Design with HTML5 and CSS - Fourth Edition: Build future-proof responsive websites using the latest HTML5 and CSS techniques. Birmingham : Packt Publishing, 2023. 498 p.

									Арк.
									76
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРІПЗ. 200175.01.22.00

27. Laurence L. S. JSON and AJAX with JavaScript for beginners: Learn how to use JSON data within your web applications. Chicago : Independently published, 2022. 116 p.

28. Philip A. JavaScript: The Comprehensive Guide to Learning Professional JavaScript Programming (The Rheinwerk Computing). Bonn : Rheinwerk Computing, 2022. 982 p.

29. Paul M. HTML, CSS, & JavaScript All-in-One For Dummies. New York : For Dummies, 2023. 848 p.

30. Jon D. PHP & MySQL: Server-side Web Development. Hoboken : Wiley, 2022. 672 p.

31. John T., Russell O. Administer and Secure Enterprise Linux: Red Hat and CentOS versions 7 and 8. Chicago : Independently published, 2020. 230 p.

32. Документація PHP. URL: <https://www.php.net/manual/en/index.php> (дата звернення: 03.04.2024).

33. Документація MySQL. URL: <https://dev.mysql.com/doc/> (дата звернення: 20.03.2024).

34. What is Data-Driven Testing? URL: https://www.tutorialspoint.com/software_testing_dictionary/data_driven_testing.htm (дата звернення: 28.04.2024).

35. Matthew H., Michael L. Software Testing Strategies: A testing guide for the 2020s. Birmingham : Packt Publishing, 2023. 378 p.

36. James B. VMware Workstation Made Easy: Virtualization for Everyone (Computers Made Easy). Chicago : CME Publishing, 2022. 155 p.

37. Tony R. Building Virtual Machine Labs: A Hands-On Guide (Second Edition): Volume I (Color Print) (Building Virtual Machine Labs: A Hands-On Guide (Second Edition) - Color Print). Chicago : Independently published, 2021. 590 p.

38. Simon A. Software Test Design: Write comprehensive test plans to uncover critical bugs in web, desktop, and mobile apps. Birmingham : Packt Publishing, 2022. 426 p.

					КВРІІЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

39. Документація PHPUnit. URL: <https://phpunit.de/documentation.html> (дата звернення: 28.04.2024).

40. Wale S. Linux Administration: A Beginner's Guide, Eighth Edition. New York : McGraw Hill, 2020. 688 p.

41. Cybellium L., Kris H. Mastering NGINX: A Comprehensive guide to Learn NGINX. Chicago : Independently published, 2023. 291 p.

					КВРІПЗ. 200175.01.22.00	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

ДОДАТОК А
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Розробка виконується згідно з процесом створення вебсайту біржі фрілансу. Вебсайт біржі фрілансу є сучасним рішенням для полегшення взаємодії між фрілансерами та замовниками послуг. Це рішення забезпечить зручність, ефективність та безпеку для обох сторін, допомагаючи знайти та замовити послуги найкращих фахівців.

1 Підстава для розробки

Підставою для розробки даного веб-застосунку є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри програмного забезпечення.

Найменування розробки: «Вебсайт біржі фрілансу».

2 Призначення розробки

2.1 Функціональне призначення

Функціональне призначення – надати клієнтам можливість перегляду та придбання музичних інструментів та обладнання за допомогою персонального комп'ютера чи смартфона. Додаток надасть користувачам можливість оплачувати товари в електронному вигляді та виступатиме посередником між клієнтами та працівниками магазину.

2.2 Експлуатаційне призначення

Веб застосунок має використовуватись звичайними людьми без жодної професійної підготовки за допомогою браузера на ПК, ноутбуках, планшетах та смартфонах за операційними системами Windows, Mac Android, IOS.

3 Вимоги до програмного продукту

3.1 Вимоги до функціональних характеристик

Система має надавати користувачам такі можливості:

- перегляд профілів фрілансерів та детальної інформації про їхні послуги.
- реєстрація та подальший вхід і вихід із системи.
- створення та публікація замовлень.
- спілкування між замовниками та фрілансерами через систему повідомлень.
- оцінювання та відгуки про виконані роботи, що допоможе формувати репутацію користувачів.

3.2 Вимоги до надійності

Надійність вебзастосунку біржі фрілансу є критичним аспектом, який включає в себе захист даних користувачів, обробку помилок, а також забезпечення безперебійної роботи системи. Детальні вимоги до надійності включають наступне:

Захист даних користувачів:

- всі персональні дані користувачів (включаючи ім'я, адресу електронної пошти, номер телефону та платіжні реквізити) повинні зберігатися в зашифрованому вигляді.
- використання сучасних алгоритмів шифрування (наприклад, AES-256) для захисту даних.
- передача даних між клієнтськими пристроями та сервером повинна здійснюватися через захищений протокол HTTPS з використанням SSL/TLS сертифікатів.

Аутентифікація та авторизація:

- використання ролей та дозволів для обмеження доступу до критичних функцій та даних лише авторизованим користувачам.

Інтерфейс користувача:

- відображення чітких та зрозумілих повідомлень про помилки для користувачів, що пояснюють проблему та пропонують можливі рішення.
- перенаправлення користувачів на відповідні функціонуючі сторінки у разі виникнення критичних помилок.

Автоматичне відновлення:

- використання механізмів автоматичного відновлення у разі збою, таких як резервне копіювання даних та відновлення з резервних копій.
- впровадження стратегій відновлення після збоїв для забезпечення швидкого відновлення системи після критичних несправностей.

Тестування та оновлення:

- регулярне проведення тестування на проникнення для виявлення вразливостей.
- впровадження автоматизованого тестування та контролю якості для запобігання помилок під час оновлень та змін у системі.

Резервне копіювання:

- регулярне створення резервних копій всіх критичних даних та зберігання їх у безпечних місцях.
- періодичне тестування процесу відновлення з резервних копій для забезпечення готовності до надзвичайних ситуацій.

Ці вимоги спрямовані на забезпечення високого рівня надійності вебзастосунку, захисту даних користувачів та стабільної роботи системи навіть у разі виникнення непередбачених ситуацій.

3.3 Вимоги до складу та параметрів технічних засобів та програмної сумісності

Для коректної роботи сайту визначено мінімальні вимоги для технічних характеристик комп'ютера:

- 32 (x86) чи 64(x64) -розрядний процесор;
- 4 гігабайти RAM;
- 4 ГБ вільного місця на диску;
- Відеокарта, що підтримує технологію DirectX 9;
- Windows 7+;
- Інтернет.

Вимоги до смартфонів:

- Android 4;
- IOS 10;
- 2 гігабайти RAM.

3.4 Вимоги до транспортування та зберігання

Програмний продукт може поставлятися через хмарні сховища та фізичні носії. Документація надається у двох варіантах – електронному та паперовому.

Умови експлуатації не виходять за межі умов експлуатації технічних засобів.

3.5 Спеціальні вимоги

Для коректної функціональності ПЗ спеціальні вимоги відсутні.

4 Вимоги до програмної документації

Замовнику має бути надано текст та опис програми, технічне завдання. Також замовнику має бути надано власне програма з інструкцією користувача.

5 Техніко-економічне обґрунтування

Продуктивний економічний ефект при використанні ПЗ полягає у оптимізації наступних процесів:

- зменшення кількості працівників магазину, що працюють з клієнтами;
- можливість знаходити цільову аудиторію не значно більшій території, що дозволить масштабувати бізнес;
- спрощенні ведення документації та оплати товарів.

6 Стадії розробки

Стадія та період	Етап	Зміст
Технічне завдання, січень	Обґрунтування необхідності розробки	Вибір теми ДП, коротка характеристика ПЗ, вимоги до ПЗ, стадії розробки ПЗ
Ескізний проєкт, січень-лютий	Створення ескізного проєкту	Початкова розробка структури системи, вибір середовища та технологій
Технічний проєкт, лютий-березень	Створення технічного проєкту	Уточнення структури та вибір остаточних технологій розробки, проєктування дизайну ПЗ
Робочий проєкт, квітень	Створення робочого проєкту	Програмна реалізація ПЗ, створення застосунку, що відповідає усім нормам
Розробка документації, травень	Створення документації	Створення супровідної документації для проєкту, розробка ілюстративних матеріалів
Тестування, травень,	Проведення тестування	Проведення тестування модулів та системи цілком, виправлення виявлених невідповідностей ПЗ
Впровадження, червень	Отримання документів від курівників	Підготовка документації, необхідної для затвердження роботи: відгуки, рецензії.
Захист, червень	Розробка документації	Підготовка до захисту та захист КвР

7 Порядок контролю та приймання

Контроль і приймання здійснюється юзерами вебзастосунку та керівниками проекту. При цьому має бути проведено перевірку функціональності ПЗ та перевірку відповідності наданої документації.

ДОДАТОК Б (обов'язковий)

КОД ПРОГРАМИ

Код класу Component:

```
<?php
if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true) die();
$this->setFrameMode(false);

if (CModule::IncludeModule("iblock"))
{
    if($arParams["IBLOCK_ID"] > 0)
        $bWorkflowIncluded = CIBlock::GetArrayByID($arParams["IBLOCK_ID"],
"WORFLOW") == "Y" && CModule::IncludeModule("workflow");
    else
        $bWorkflowIncluded = CModule::IncludeModule("workflow");

    if (!$bWorkflowIncluded)
    {
        if ($arParams["STATUS_NEW"] != "N" && $arParams["STATUS_NEW"] !=
"NEW") $arParams["STATUS_NEW"] = "ANY";
    }

    if(!is_array($arParams["STATUS"]))
    {
        if($arParams["STATUS"] === "INACTIVE")
            $arParams["STATUS"] = array("INACTIVE");
        else
            $arParams["STATUS"] = array("ANY");
    }

    $arGroups = $USER->GetUserGroupArray();

    // check whether current user has access to view list
    if ($USER->IsAdmin() || is_array($arGroups) &&
is_array($arParams["GROUPS"]) && count(array_intersect($arGroups,
$arParams["GROUPS"])) > 0)
    {
        $bAllowAccess = true;
    }
    elseif ($USER->GetID() > 0 && $arParams["ELEMENT_ASSOC"] != "N")
    {
        $bAllowAccess = true;
    }
    else
    {
        $bAllowAccess = false;
    }

    // if user has access
    if ($bAllowAccess)
    {
        $arResult["CAN_EDIT"] = $arParams["ALLOW_EDIT"] == "Y" ? "Y" :
"N";
    }
}
```

```

$arResult["CAN_DELETE"] = $arParams["ALLOW_DELETE"] == "Y" ? "Y" :
"N";

if ($USER->GetID())
{
    $arResult["NO_USER"] = "N";

    // get list of iblock properties and list of iblock property
    ids
    $rsIBlockPropertyList =
    CIBlockProperty::GetList(array("sort"=>"asc", "name"=>"asc"),
    array("ACTIVE"=>"Y", "IBLOCK_ID"=>$arParams["IBLOCK_ID"]));
    $arIBlockPropertyList = array();
    $arPropertyIDs = array();
    $i = 0;
    while ($arProperty = $rsIBlockPropertyList->GetNext())
    {
        $arIBlockPropertyList[] = $arProperty;
        $arPropertyIDs[] = $arProperty["ID"];
    }

    // set starting filter value
    $arFilter = array("IBLOCK_TYPE" => $arParams["IBLOCK_TYPE"],
    "IBLOCK_ID" => $arParams["IBLOCK_ID"], "SHOW_NEW" => "Y");
    // check type of user association to iblock elements and add
    user association to filter

    if ($arParams["ELEMENT_ASSOC"] == "PROPERTY_ID" &&
    intval($arParams["ELEMENT_ASSOC_PROPERTY"]) > 0 &&
    in_array($arParams["ELEMENT_ASSOC_PROPERTY"], $arPropertyIDs))
    {
        $arFilter["PROPERTY_". $arParams["ELEMENT_ASSOC_PROPERTY"]] = $USER-
        >GetID();
    }
    else
    {
        $arFilter["CREATED_BY"] = $USER->GetID();
    }

    // deleting element
    if (check_bitrix_sessid() && $_REQUEST["delete"] == "Y" &&
    $arResult["CAN_DELETE"])
    {
        $arParams["ID"] = intval($_REQUEST["CODE"]);

        // try to get element with id, for user and for iblock
        $rsElement = CIBlockElement::GetList(array(),
        array_merge($arFilter, array("ID" => $arParams["ID"])));
        if ($arElement = $rsElement->GetNext())
        {
            // delete one
            $DB->StartTransaction();
            if(!CIBlockElement::Delete($arElement["ID"]))
            {
                $DB->Rollback();
            }
            else
            {
                $DB->Commit();
            }
        }
    }
}

```

```

        if ($bWorkflowIncluded)
        {
            $rsWFStatus = CWorkflowStatus::GetList("c_sort",
"asc", array("ACTIVE" => "Y"));
            $arResult["WF_STATUS"] = array();
            while ($arStatus = $rsWFStatus->GetNext())
            {
                $arResult["WF_STATUS"][$arStatus["ID"]] =
$arStatus["TITLE"];
            }
        }
        else
        {
            $arResult["ACTIVE_STATUS"] = array("Y" =>
 GetMessage("IBLOCK_FORM_STATUS_ACTIVE"), "N" =>
 GetMessage("IBLOCK_FORM_STATUS_INACTIVE"));
        }

        // get elements list using generated filter
        $rsIBlockElements = CIBlockElement::GetList(array("SORT" =>
"ASC"), $arFilter);

        $arResult["ELEMENTS_COUNT"] = $rsIBlockElements-
>SelectedRowCount();
        //$page_split = intval(COption::GetOptionString("iblock",
"RESULTS_PAGEN"));
        $arParams["NAV_ON_PAGE"] = intval($arParams["NAV_ON_PAGE"]);
        $arParams["NAV_ON_PAGE"] = $arParams["NAV_ON_PAGE"] > 0 ?
$arParams["NAV_ON_PAGE"] : 10;

        $rsIBlockElements->NavStart($arParams["NAV_ON_PAGE"]);

        // get paging to component result
        if ($arParams["NAV_ON_PAGE"] < $arResult["ELEMENTS_COUNT"])
        {
            $arResult["NAV_STRING"] = $rsIBlockElements-
>GetPageNavString(GetMessage("IBLOCK_LIST_PAGES_TITLE"), "", true);
        }

        // get current page elements to component result
        $arResult["ELEMENTS"] = array();
        $bCanEdit = false;
        $bCanDelete = false;
        while ($arElement = $rsIBlockElements->NavNext(false))
        {
            $arElement = htmlspecialchars($arElement);
            if ($bWorkflowIncluded)
            {
                $PREVIOUS_ID = $arElement['ID'];
                $LAST_ID =
CIBlockElement::WF_GetLast($arElement['ID']);
                if ($LAST_ID != $arElement["ID"])
                {
                    $rsElement =
CIBlockElement::GetByID($LAST_ID);
                    $arElement = $rsElement->GetNext();
                }

                $arElement["ID"] = $PREVIOUS_ID;

                $arElement["CAN_EDIT"] = $arResult["CAN_EDIT"]
== "Y" ? (in_array($arElement["WF_STATUS_ID"], $arParams["STATUS"]) == true ?
"Y" : "N") : "N";

```

```

                $arResult["CAN_DELETE"] =
$arResult["CAN_DELETE"] == "Y" ? (in_array($arResult["WF_STATUS_ID"],
$arParams["STATUS"]) == true ? "Y" : "N") : "N";
            }
            elseif (in_array("INACTIVE", $arParams["STATUS"]) ===
true)
            {
                $arResult["CAN_EDIT"] = $arResult["CAN_EDIT"]
== "Y" ? ($arResult["ACTIVE"] == "Y" ? "N" : "Y") : "N";
                $arResult["CAN_DELETE"] =
$arResult["CAN_DELETE"] == "Y" ? ($arResult["ACTIVE"] == "Y" ? "N" : "Y") :
"N";
            }
            else
            {
                $arResult["CAN_EDIT"] = $arResult["CAN_EDIT"];
                $arResult["CAN_DELETE"] =
$arResult["CAN_DELETE"];
            }

            if (!$bCanEdit && $arResult["CAN_EDIT"] == "Y" &&
$arResult["CAN_EDIT"] == "Y")
            {
                $bCanEdit = true;
            }

            if (!$bCanDelete && $arResult["CAN_DELETE"] == "Y" &&
$arResult["CAN_DELETE"] == "Y")
            {
                $bCanDelete = true;
            }

            $arResult["ELEMENTS"][] = $arResult;
        }

        if ($arResult["CAN_EDIT"] == "Y" && !$bCanEdit)
$arResult["CAN_EDIT"] = "N";
        if ($arResult["CAN_DELETE"] == "Y" && !$bCanDelete)
$arResult["CAN_DELETE"] = "N";
    }
    else
    {
        $arResult["NO_USER"] = "Y";
    }

    $arResult["MESSAGE"] =
htmlspecialchars($_REQUEST["strIMessage"]);

    $this->IncludeComponentTemplate();
}
else
{
    $APPLICATION->AuthForm("");
}
}
?>

```

Код класы Parameters:

```

<?php
if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true)
{

```

```

        die();
    }

    /** @var array $arCurrentValues */
    use Bitrix\Main\Loader;

    if (!Loader::includeModule("iblock"))
    {
        return;
    }

    $iblockExists = (!empty($arCurrentValues['IBLOCK_ID']) &&
        (int)$arCurrentValues['IBLOCK_ID'] > 0);

    if ($iblockExists)
    {
        $arIBlock = CIBlock::GetArrayByID($arCurrentValues["IBLOCK_ID"]);

        $bWorkflowIncluded = ($arIBlock["WORKFLOW"] == "Y") &&
        Loader::includeModule("workflow");
        $bBizproc = ($arIBlock["BIZPROC"] == "Y") &&
        Loader::includeModule("bizproc");
    }
    else
    {
        $bWorkflowIncluded = Loader::includeModule("workflow");
        $bBizproc = false;
    }

    $arIBlockType = CIBlockParameters::GetIBlockTypes();

    $arIBlock=array();
    $iblockFilter = [
        'ACTIVE' => 'Y',
    ];
    if (!empty($arCurrentValues['IBLOCK_TYPE']))
    {
        $iblockFilter['TYPE'] = $arCurrentValues['IBLOCK_TYPE'];
    }
    $rsIBlock = CIBlock::GetList(Array("sort" => "asc"), $iblockFilter);
    while($arr=$rsIBlock->Fetch())
    {
        $arIBlock[$arr["ID"]] = "[".$arr["ID"]."] ".$arr["NAME"];
    }

    $arProperty_LNSF = array(
        "NAME" => GetMessage("IBLOCK_ADD_NAME"),
        "TAGS" => GetMessage("IBLOCK_ADD_TAGS"),
        "IBLOCK_SECTION" => GetMessage("IBLOCK_ADD_IBLOCK_SECTION"),
        "PREVIEW_TEXT" => GetMessage("IBLOCK_ADD_PREVIEW_TEXT"),
        "PREVIEW_PICTURE" => GetMessage("IBLOCK_ADD_PREVIEW_PICTURE"),
        "DETAIL_TEXT" => GetMessage("IBLOCK_ADD_DETAIL_TEXT"),
        "DETAIL_PICTURE" => GetMessage("IBLOCK_ADD_DETAIL_PICTURE"),
    );

    $arProperty = [];
    if ($iblockExists)
    {
        $rsProp = CIBlockProperty::GetList(
            [
                "SORT" => "ASC",
                "NAME" => "ASC",
            ],
            [

```

```

        "ACTIVE" => "Y",
        "IBLOCK_ID" => $arCurrentValues["IBLOCK_ID"],
    ]
);
while ($arr = $rsProp->Fetch())
{
    $arProperty[$arr["ID"]] = "[" . $arr["CODE"] . "]" . " " .
$arProperty["NAME"];
    if (in_array($arr["PROPERTY_TYPE"], ["L", "N", "S", "F"]))
    {
        $arProperty_LNSF[$arr["ID"]] = "[" . $arr["CODE"] . "]" . " " .
$arProperty["NAME"];
    }
}

$arGroups = array();
$rsGroups = CGroup::GetList("c_sort", "asc", Array("ACTIVE" => "Y"));
while ($arGroup = $rsGroups->Fetch())
{
    $arGroups[$arGroup["ID"]] = $arGroup["NAME"];
}

if ($bWorkflowIncluded)
{
    $rsWFStatus = CWorkflowStatus::GetList("c_sort", "asc", Array("ACTIVE"
=> "Y"));
    $arWFStatus = array();
    while ($arWFS = $rsWFStatus->Fetch())
    {
        $arWFStatus[$arWFS["ID"]] = $arWFS["TITLE"];
    }
}
else
{
    $arActive = array("ANY" => GetMessage("IBLOCK_STATUS_ANY"), "INACTIVE"
=> GetMessage("IBLOCK_STATUS_INCATIVE"));
}

$arAllowEdit = array("N" => GetMessage("IBLOCK_ALLOW_N"), "CREATED_BY" =>
GetMessage("IBLOCK_CREATED_BY"), "PROPERTY_ID" =>
GetMessage("IBLOCK_PROPERTY_ID"));

$arComponentParameters = array(
    "GROUPS" => array(
        "PARAMS" => array(
            "NAME" => GetMessage("IBLOCK_PARAMS"),
            "SORT" => "200"
        ),
        "ACCESS" => array(
            "NAME" => GetMessage("IBLOCK_ACCESS"),
            "SORT" => "400",
        ),
    ),
    "PARAMETERS" => array(
        "SEF_MODE" => Array(
            /*"edit" => array(
                "NAME" => GetMessage("IBLOCK_SEF_EDIT"),
                "DEFAULT" => "?edit=Y",
            ),*/
        ),
        "IBLOCK_TYPE" => array(

```

```

        "PARENT" => "DATA_SOURCE",
        "NAME" => GetMessage("IBLOCK_TYPE"),
        "TYPE" => "LIST",
        "ADDITIONAL_VALUES" => "Y",
        "VALUES" => $arIBlockType,
        "REFRESH" => "Y",
    ),
    "IBLOCK_ID" => array(
        "PARENT" => "DATA_SOURCE",
        "NAME" => GetMessage("IBLOCK_IBLOCK"),
        "TYPE" => "LIST",
        "ADDITIONAL_VALUES" => "Y",
        "VALUES" => $arIBlock,
        "REFRESH" => "Y",
    ),
    "GROUPS" => array(
        "PARENT" => "ACCESS",
        "NAME" => GetMessage("IBLOCK_GROUPS"),
        "TYPE" => "LIST",
        "MULTIPLE" => "Y",
        "ADDITIONAL_VALUES" => "N",
        "VALUES" => $arGroups,
    ),
    "STATUS" => array(
        "PARENT" => "ACCESS",
        "NAME" => $bWorkflowIncluded?
        GetMessage("CP_BIEAL_IBLOCK_STATUS") : GetMessage("IBLOCK_S_ACTIVE"),
        "TYPE" => "LIST",
        "MULTIPLE" => $bWorkflowIncluded ? "Y" : "N",
        "VALUES" => $bWorkflowIncluded ? $arWFStatus : $arActive,
    ),
    "EDIT_URL" => array(
        "PARENT" => "PARAMS",
        "TYPE" => "TEXT",
        "NAME" => GetMessage("IBLOCK_ADD_EDIT_URL"),
    ),
    "ELEMENT_ASSOC" => array(
        "PARENT" => "ACCESS",
        "NAME" => GetMessage("IBLOCK_ELEMENT_ASSOC"),
        "TYPE" => "LIST",
        "ADDITIONAL_VALUES" => "N",
        "VALUES" => $arAllowEdit,
        "REFRESH" => "Y",
        "DEFAULT" => "CREATED_BY",
    ),
),
);

if (($arCurrentValues["ELEMENT_ASSOC"] ?? 'CREATED_BY') === "PROPERTY_ID")
{
    $arComponentParameters["PARAMETERS"]["ELEMENT_ASSOC_PROPERTY"] = array(
        "PARENT" => "ACCESS",
        "NAME" => GetMessage("IBLOCK_ELEMENT_ASSOC_PROPERTY"),
        "TYPE" => "LIST",
        "MULTIPLE" => "N",
        "VALUES" => $arProperty,
        "ADDITIONAL_VALUES" => "Y",
    );
};

```

```

}
if (($arCurrentValues["ELEMENT_ASSOC"] ?? 'CREATED_BY') !== "N")
{
    $arComponentParameters["PARAMETERS"]["ALLOW_EDIT"] = array(
        "PARENT" => "ACCESS",
        "NAME" => GetMessage("IBLOCK_ALLOW_EDIT"),
        "TYPE" => "CHECKBOX",
        "DEFAULT" => "Y",
    );

    $arComponentParameters["PARAMETERS"]["ALLOW_DELETE"] = array(
        "PARENT" => "ACCESS",
        "NAME" => GetMessage("IBLOCK_ALLOW_DELETE"),
        "TYPE" => "CHECKBOX",
        "DEFAULT" => "Y",
    );
}

$arComponentParameters["PARAMETERS"]["NAV_ON_PAGE"] = array(
    "PARENT" => "PARAMS",
    "NAME" => GetMessage("IBLOCK_NAV_ON_PAGE"),
    "TYPE" => "TEXT",
    "DEFAULT" => "10",
);

$arComponentParameters["PARAMETERS"]["MAX_USER_ENTRIES"] = array(
    "PARENT" => "PARAMS",
    "NAME" => GetMessage("IBLOCK_MAX_USER_ENTRIES"),
    "TYPE" => "TEXT",
    "DEFAULT" => "100000",
);

```

Код класу Template:

```

<?
if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true) die();
$this->setFrameMode(false);

$colspan = 2;
if ($arResult["CAN_EDIT"] == "Y") $colspan++;
if ($arResult["CAN_DELETE"] == "Y") $colspan++;
?>
<?if ($arResult["MESSAGE"] <> ''):?>
    <?ShowNote($arResult["MESSAGE"])?>
<?endif?>
<table class="data-table">
<?if($arResult["NO_USER"] == "N"):?>
    <thead>
        <tr>
            <td<?=$colspan > 1 ? " colspan=\\". $colspan. \\" :
            ""?>><?=GetMessage("IBLOCK_ADD_LIST_TITLE")?></td>
        </tr>
    </thead>
    <tbody>
        <?if (count($arResult["ELEMENTS"]) > 0):?>
            <?foreach ($arResult["ELEMENTS"] as $arElement):?>
                <tr>
                    <td><!--a href="detail.php?CODE=<?=$arElement["ID"]?>"--
                    ><?=$arElement["NAME"]?><!--/a--></td>
                    <td><small><?=is_array($arResult["WF_STATUS"]) ?
                    $arResult["WF_STATUS"][$arElement["WF_STATUS_ID"]] :
                    $arResult["ACTIVE_STATUS"][$arElement["ACTIVE"]]?></small></td>

```

```

                <?if ($arResult["CAN_EDIT"] == "Y"):?>
                <td><?if ($arResult["CAN_EDIT"] == "Y"):?><a
href="<?=$arParams["EDIT_URL"]?>?edit=Y&CODE=<?=$arResult["ID"]?>"><?=Ge
tMessage("IBLOCK_ADD_LIST_EDIT")?><?else:??&nbsp;<?endif?></a></td>
                <?endif?>
                <?if ($arResult["CAN_DELETE"] == "Y"):?>
                <td><?if ($arResult["CAN_DELETE"] == "Y"):?><a
href="?delete=Y&CODE=<?=$arResult["ID"]?>&&<?=bitrix_sessid_get()?>"
onClick="return confirm('<?echo CUtil::JSEscape(str_replace("#ELEMENT_NAME#",
$arResult["NAME"],
GetMessage("IBLOCK_ADD_LIST_DELETE_CONFIRM")))?>')"><?=GetMessage("IBLOCK_ADD
_LIST_DELETE")?></a><?else:??&nbsp;<?endif?></td>
                <?endif?>
            </tr>
        <?endforeach?>
    <?else:??>
    <tr>
        <td<?=$colspan > 1 ? " colspan=\\". $colspan. \\" :
""?>><?=GetMessage("IBLOCK_ADD_LIST_EMPTY")?></td>
    </tr>
    <?endif?>
</tbody>
</table>
<tfoot>
    <tr>
        <td<?=$colspan > 1 ? " colspan=\\". $colspan. \\" : ""?>><?if
($arParams["MAX_USER_ENTRIES"] > 0 && $arResult["ELEMENTS_COUNT"] <
$arParams["MAX_USER_ENTRIES"]):?><a
href="<?=$arParams["EDIT_URL"]?>?edit=Y"><?=GetMessage("IBLOCK_ADD_LINK_TITLE
")?></a><?else:??<?=GetMessage("IBLOCK_LIST_CANT_ADD_MORE")?><?endif?></td>
    </tr>
</tfoot>
</table>
<?if ($arResult["NAV_STRING"] <> ''):?><?=$arResult["NAV_STRING"]?><?endif?>

```

Код класы Description:

```

<?
if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true) die();
$arComponentDescription = array(
    "NAME" => GetMessage("IBLOCK_ELEMENT_ADD_LIST_NAME"),
    "DESCRIPTION" => GetMessage("IBLOCK_ELEMENT_ADD_LIST_DESCRIPTION"),
    "ICON" => "/images/eaddlist.gif",
    "PATH" => array(
        "ID" => "content",
        "CHILD" => array(
            "ID" => "iblock_element_add",
            "NAME" => GetMessage("T_IBLOCK_DESC_ELEMENT_ADD"),
        ),
    ),
);
?>

```

Код класы Header:

```

<? global $USER;
if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true) die();?>
<?
IncludeTemplateLangFile(__FILE__);

```

```

?>
<!DOCTYPE html>
<html lang="<?=LANGUAGE_ID;?>">
<head>
  <?=$APPLICATION->ShowHead();?>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="description" content="">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
  <!-- Title -->
  <title>IT</title>
  <!-- Favicon Icon -->
  <link rel="shortcut icon"
href="/local/templates/.default/images/favicon.png" type="image/x-icon">
  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@500;600&family=Ro
boto:wght@400;500;700&display=swap" rel="stylesheet">

</head>
<body>
<?=$APPLICATION->ShowPanel();?>
<div class="page-wrapper">

  <!-- main header -->
  <header class="main-header header-one menu-white">

    <div class="header-top-wrap bgc-gray">
      <div class="container">
        <div class="header-top">
          <ul>
            <li><i class="far fa-envelope"></i> <a
href="mailto:support@gmail.com">support@gmail.com</a></li>
            <li><i class="far fa-phone"></i> <a
href="callto:+000(123)45688">+000 (123) 456 88</a></li>
            <li class="for-none" style="color: white"><i
class="far fa-clock"></i> Working Hours Support : Monday - Sunday, 9 am to 6
pm</li>
            <li>
              <div class="social-style-one">
                <a href=""><i class="fab fa-facebook-
f"></i></a>
                <a href=""><i class="fab fa-twitter"></i></a>
                <a href=""><i class="fab fa-
instagram"></i></a>
                <a href=""><i class="fab fa-linkedin-
in"></i></a>
              </div>
            </li>
          </ul>
        </div>
      </div>
    </div>

  <!--Header-Upper-->
  <div class="header-upper bgc-black">
    <div class="container clearfix">

      <div class="header-inner rel d-flex align-items-center">
        <div class="logo-outer">

```

```
        <div class="logo"><a href="index.html"></a></div>
    </div>
```

```
<div class="nav-outer mx-auto clearfix">
    <!-- Main Menu -->
    <?APPLICATION->IncludeComponent(
        "bitrix:menu",
        "top_multi",
        array(
            "COMPONENT_TEMPLATE" => "top_multi",
            "ROOT_MENU_TYPE" => "top",
            "MENU_CACHE_TYPE" => "N",
            "MENU_CACHE_TIME" => "3600",
            "MENU_CACHE_USE_GROUPS" => "Y",
            "MENU_CACHE_GET_VARS" => array(
                ),
            "MAX_LEVEL" => "2",
            "CHILD_MENU_TYPE" => "left",
            "USE_EXT" => "N",
            "DELAY" => "N",
            "ALLOW_MULTI_SELECT" => "N"
        ),
        false
    );?>
    <!-- Main Menu End-->
</div>
```

```
<!-- Nav Search -->
<div class="nav-search py-10">
    <button class="far fa-search"></button>
    <form action="#" class="hide">
        <input type="text" placeholder="Search"
class="searchbox" required="">
        <button type="submit" class="searchbutton far fa-
search"></button>
    </form>
</div>
```

```
<!-- Menu Button -->
<?php
if ($USER->IsAuthorized()) {
    // Користувач авторизований, виводимо його ім'я
    $userName = $USER->GetFullName(); // Отримуємо повне
ім'я користувача
    $logoutUrl = $APPLICATION-
>GetCurPageParam("logout=yes", array("logout"));
    $logoutUrlWithSessid = $APPLICATION-
>GetCurPageParam("logout=yes&sessid=".bitrix_sessid(), array("logout"));
    ?>
    <div class="menu-btns">
        <span class="theme-btn"><?php echo $userName;
?></span>
    </div>

    <div class="menu-btns">
        <a href="<?= $logoutUrlWithSessid; ?>"
class="theme-btn">Logout</a>
    </div>
    <?php
} else {
    // Користувач не авторизований, виводимо кнопки Log
in i Sing in
```

```

        ?>
        <div class="menu-btns">
            <a href="/auth/auth.php" class="theme-btn">Log
in</a>
        </div>
        <div class="menu-btns">
            <a href="/auth/registration.php" class="theme-
btn">Sing in</a>
        </div>
        <?php
    }
    ?>
</div>

</div>
<!--End Header Upper-->
</header>

```

Код класу Footer:

```

<!-- footer area start -->
<footer class="main-footer bgc-gray footer-white rel z-1">

    <div class="container">
        <div class="row medium-gap">
            <div class="col-xl-3 col-sm-6">
                <div class="footer-widget widget_about wow fadeInUp delay-0-
2s">
                    <div class="footer-logo mb-30">
                        <a href="index.html"></a>
                    </div>

                    <a href="about.html" class="read-more">Learn More Us <i
class="fas fa-long-arrow-right"></i></a>
                </div>
            </div>
            <div class="col-xl-3 col-sm-6 order-xl-2">
                <div class="footer-widget widget_newsletter wow fadeInUp
delay-0-6s">
                    <h4 class="footer-title">Newsletter</h4>
                    <p>Sed perspiciatis unde omnis iste natus error voluptatem
accusante.</p>

                    <form action="#">
                        <label for="email"><i class="far fa-
envelope"></i></label>
                            <input id="email" type="email"
placeholder="support@gmail.com" required>
                            <button>Sign Up</button>
                    </form>
                    <h5>Follow Us</h5>
                    <div class="social-style-one">
                        <a href="#"><i class="fab fa-facebook-f"></i></a>
                        <a href="#"><i class="fab fa-twitter"></i></a>
                        <a href="#"><i class="fab fa-instagram"></i></a>
                        <a href="#"><i class="fab fa-linkedin-in"></i></a>
                    </div>
                </div>
            </div>
            <div class="col-xl-6">
                <div class="row">
                    <div class="col-md-4 col-6 col-small">

```

```

        <div class="footer-widget widget_nav_menu wow
fadeInUp delay-0-3s">
        <h4 class="footer-title">Quick Links</h4>
        <ul class="list-style-two">
        <li><a href="services.html">Latest
Services</a></li>
        <li><a href="project-grid.html">Popular
Works</a></li>
        <li><a href="faqs.html">Asked
Questions</a></li>
        </ul>
        </div>
    </div>
    <div class="col-md-4 col-6 col-small">
        <div class="footer-widget widget_nav_menu wow
fadeInUp delay-0-4s">
        <h4 class="footer-title">Services</h4>
        <ul class="list-style-two">
        <li><a href="service-details.html">Product
Design</a></li>
        <li><a href="service-details.html">Design &
Development</a></li>
        <li><a href="service-details.html">UX/UI
Strategy</a></li>
        <li><a href="service-details.html">Search
Engine</a></li>
        <li><a href="service-details.html">IT
Consulting</a></li>
        <li><a href="service-details.html">Software
Development</a></li>
        <li><a href="service-details.html">Business
Analysis</a></li>
        </ul>
        </div>
    </div>
    <div class="col-md-4 col-6 col-small">
        <div class="footer-widget widget_nav_menu wow
fadeInUp delay-0-5s">
        <h4 class="footer-title">Support</h4>
        <ul class="list-style-two">
        <li><a href="contact.html">Start
Here</a></li>
        <li><a href="contact.html">Password
Protected</a></li>
        <li><a href="contact.html">Licenses</a></li>
        <li><a href="contact.html">Changelog</a></li>
        </ul>
        </div>
    </div>
    </div>
    </div>
    </div>
    </div>
    <div class="footer-bottom bgc-black mt-20 pt-20">
        <div class="container">
            <div class="row align-items-center">
                <div class="col-lg-8">
                    <div class="footer-bottom-menu mb-10 wow fadeInRight
delay-0-2s">
                        <ul>
                            <li><a href="#">Version 1.2</a></li>
                            <li><a href="#">Image Licenses</a></li>
                            <li><a href="#">Instructions</a></li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
    <div class="col-lg-4">
        <div class="copyright-text text-lg-end wow fadeInLeft
delay-0-2s">
            <p>© 2024. All rights reserved.</p>
        </div>
    </div>
</div>
</div>
<div class="footer-shapes">
    
    
    
</div>
</footer>
<!-- footer area end -->

<!-- Scroll Top Button -->
<button class="scroll-top scroll-to-target" data-target="html"><span
class="fas fa-angle-double-up"></span></button>
</div>
<!--End pagewrapper-->
<script src="/local/templates/.default/js/jquery-3.6.0.min.js"></script>
<!-- Bootstrap -->
<script src="/local/templates/.default/js/bootstrap.min.js"></script>
<!-- Appear Js -->
<script src="/local/templates/.default/js/appear.min.js"></script>
<!-- Slick -->
<script src="/local/templates/.default/js/slick.min.js"></script>
<!-- Magnific Popup -->
<script src="/local/templates/.default/js/jquery.magnific-
popup.min.js"></script>
<!-- Nice Select -->
<script src="/local/templates/.default/js/jquery.nice-
select.min.js"></script>
<!-- Image Loader -->
<script src="/local/templates/.default/js/imagesloaded.pkgd.min.js"></script>
<!-- Circle Progress -->
<script src="/local/templates/.default/js/circle-progress.min.js"></script>
<!-- Isotope -->
<script src="/local/templates/.default/js/isotope.pkgd.min.js"></script>
<!-- WOW Animation -->
<script src="/local/templates/.default/js/wow.min.js"></script>
<!-- Custom script -->
<script src="/local/templates/.default/js/script.js"></script>
</body>
</html>

```

ДОДАТОК В
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

ВЕБСАЙТ БІРЖИ ФРІЛАНСУ

Виконав: студент гр. ПЗ-20-1 Олег Хміль

Керівник: канд. пед. наук, доцент Наталія Праворська

Актуальність теми

Вебсайти біржі фрілансу на сьогоднішній день стали не лише важливим елементом економіки, але й відображенням розвитку сучасного ринку праці.

Онлайн-платформи відкривають віртуальний простір, де фрілансери та клієнти з різних країн можуть легко знаходити одне одного для виконання різноманітних проектів та послуг. За допомогою біржам фрілансу, талановиті фахівці отримують можливість знайти роботу у будь-якій галузі для виконання найрізноманітніших завдань, тоді як компанії та підприємці можуть знаходити професіоналів для реалізації своїх проектів.



Мета роботи

Мета роботи – це розробка вебсайту, що дозволить опубліковувати проєкти та взаємодіяти з ними і забезпечити зручний та зрозумілий інтерфейс для користувачів.

Предметом дослідження є процес дослідження та розроблення вебсайту біржі фрілансу.

Об'єктом дослідження є структура вебсайту біржі фрілансу, яка забезпечує функціональність для публікації проєктів та взаємодії з ними користувачів.

Завдання:

- дослідити предметну область, з урахування особливостей її реалізації і функціоналу;
- виконати аналіз існуючих рішень на ринку, визначити їх переваги та недоліки і мінімальний функціонал;
- розробити систему, яка буде задовольняти вимоги користувачів;

Основні завдання та вимоги

Завдання:

- Реалізувати перегляд профілів фрілансерів та детальної інформації про їхні послуги.
- Реалізувати реєстрацію та подальший вхід і вихід із системи.
- Реалізувати створення та публікація замовлень.
- Реалізувати спілкування між замовниками та фрілансерами через систему повідомлень.
- Реалізувати оцінювання та відгуки про виконані роботи, що допоможе формувати репутацію користувачів.

Вимоги:

- Надійність: Система повинна бути стійкою до збоїв і забезпечувати безперервну роботу з мінімальними простоями.
- Безпека: Забезпечення захисту персональних даних користувачів, використання шифрування для передачі даних та автентифікації користувачів.
- Масштабованість: Можливість обробляти зростаючу кількість користувачів і транзакцій без втрати продуктивності.
- Продуктивність: Швидкий час відгуку на запити користувачів і оптимізована робота бази даних.
- Підтримка та обслуговування: Легкість у підтримці та оновленні системи, зокрема, можливість додавання нових функцій без порушення роботи наявних.

Аналіз існуючих рішень

Для аналізу було обрано три популярні фріланс біржі:



Freelancer – це платформа для фрілансу, яка займає високі позиції у світі. На цьому сайті зібрана велика кількість фахівців із різних галузей, які готові виконувати завдання та проекти для клієнтів з усього світу.



Fiverr – це вебплатформа для фрілансу, що отримала значну популярність у останні роки. На Fiverr можна знайти фахівців різних напрямків, які готові виконати різноманітні завдання за різними цінами, починаючи від п'яти доларів



Upwork є популярною онлайн-платформою, яка забезпечує зв'язок між фрілансерами та клієнтами з усього світу. Завдяки широкому спектру послуг, включаючи: веброзробку, дизайн, маркетинг, переклад, написання текстів та адміністрування, також дана платформа виконує потреби різноманітних бізнесів та індивідуальних користувачів

Аналіз існуючих рішень

ПОРІВНЯННЯ ПЛАТФОРМ БІРЖ ФРІЛАНСУ:

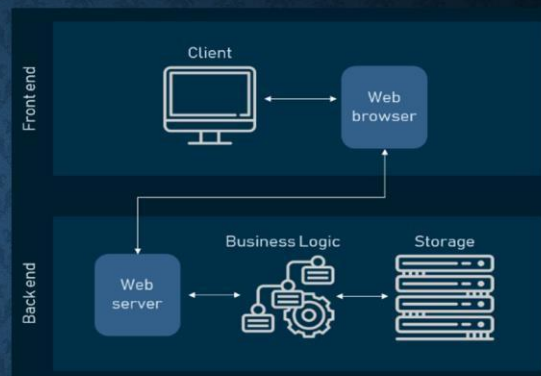
Критерій	Freelancer	Upwork	Fiverr
Кількість користувачів	50+ мільйонів	12+ мільйонів	3+ мільйони
Різноманітність проектів	Широкий спектр	Широкий спектр	Більше орієнтовано на невеликі проекти
Ціни	Різноманітні	Різноманітні, загалом вищі	Фіксовані ціни за «гігі»
Комісія	3-10% (з фрілансера)	10-20% (з фрілансера)	20% (з фрілансера)
Рівень конкуренції	Високий	Високий	Дуже високий
Складність отримання проекту	Складний	Складний	Дуже складний
Способи оплати	Різноманітні	Різноманітні	PayPal, кредитні картки
Підтримка	Базова	Розширена	Базова
Інтерфейс	Сучасний, складний для навігації	Сучасний, зручний	Сучасний, зрозумілий, інтуїтивно
Переваги	Велика спільнота, багато проектів	Багато проектів, високі ціни	Прості у використанні, фіксовані ціни
Недоліки	Висока конкуренція, низькі ціни	Висока конкуренція, високі комісії	Дуже висока конкуренція, низькі ціни

ВИКОРИСТАНІ ТЕХНОЛОГІЇ



Архітектура програмного забезпечення

Для реалізації була обрана клієнт-серверна архітектура та з обраним архітектурним шаблоном MVC для розробки програмного забезпечення

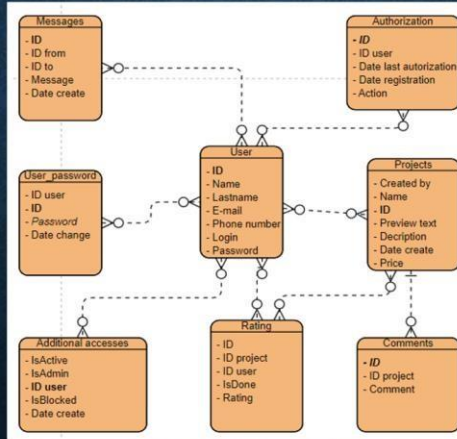


Даний вебсайт буде працювати наступним чином, користувач буде відвідувати вебсторінку через браузер та за допомогою браузера користувач буде відсилати запити, тому за це буде відповідати front-end частина. Після надіслання запиту з сторони користувача надійде запит на дані, які має відправити сервер.

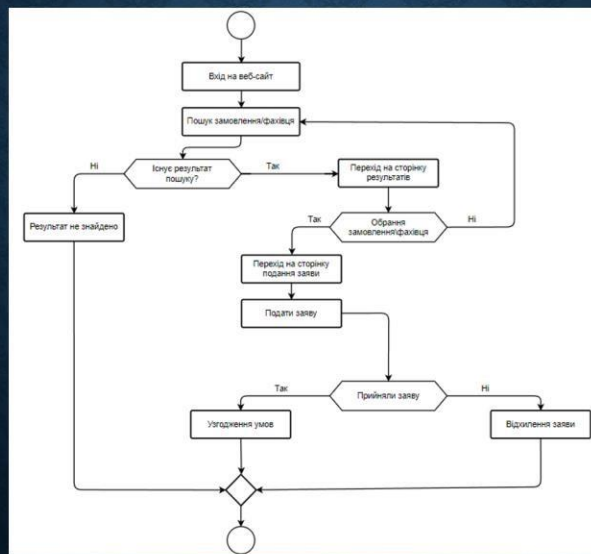
Для роботи сервера відповідає back-end частина, яка буде обробляти отримані запити від користувача. В свою чергу сервер проводить перевірку запиту за допомогою розробленої бізнес-логіки. Сервер оброблятиме запит в залежності від типу запиту та відправить відповідні дані з сервера.

Проектування Бази Даних

На рисунку відображена логічна модель бази даних з необхідними зв'язками та структурою побудована на основі структури необхідних таблиць та їх полів:

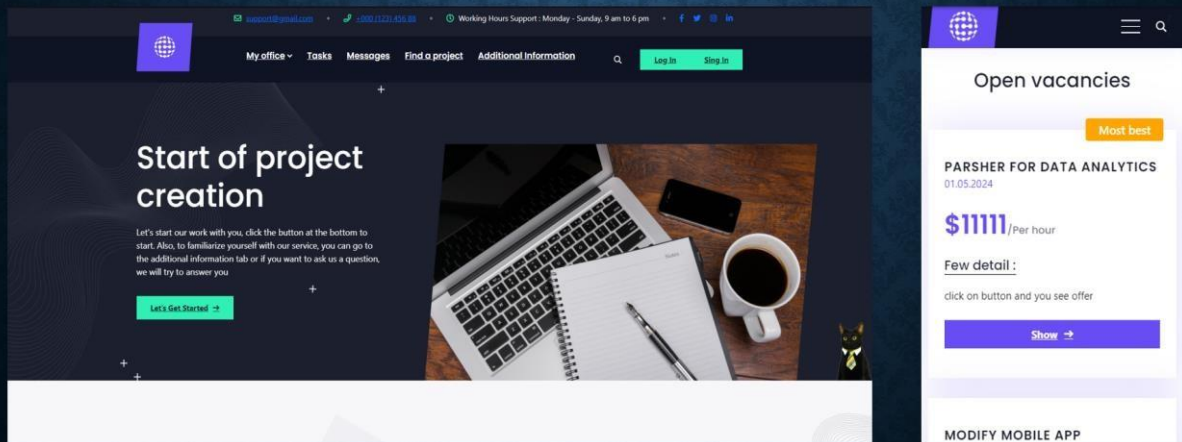


ПРИНЦИП РОБОТИ



Інтерфейс

Дизайн вебсайту виконано в доволі простому стилі, щоб не відволікати користувача від роботи та весь зроблений інтерфейс в збалансованих кольорах:



Тестування

Для перевірки роботи вебсайту було виконано три різних тестування: Створення образу віртуальної машини для тестів програмного забезпечення на локальному сервері, Unit Test, тестування за допомогою влаштованих інструментів Bitrix Framework

Advanced features		
Database connection parameters	✓ Success	
Redundant output in configuration files	✓ Success	
Session retention without UserAgent	✓ Success	
Website Parameters	✓ Connect	
Stack size; pure recursion_limit	✓ Success	
File upload	✓ Success	
Upload files over IMB	✓ Success	
Upload file using php://input	✓ Success	
POST requests with many parameters	✓ Success	
Check for unsent messages	✓ Success	
Local redirects (LocalRedirect function)	✓ Success	
Using cache files	✓ Success	
Access to update server	✓ Success	
HTTP authorization	✓ Success	
File creation and erasation	✓ Success	
gravatarize support for SWF	✓ Success	

PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

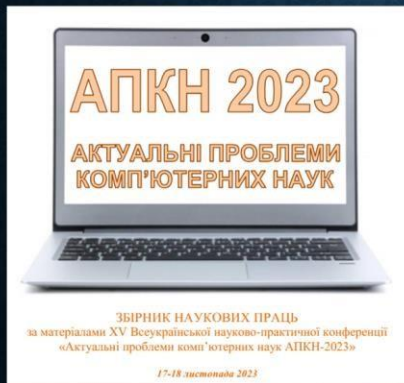
1 / 1 (100%)

Time: 00:00.046, Memory: 8.00 MB

OK (1 test, 4 assertions)

№	Функціональна вимога	Вхідні дані	Отриманий результат	Результат
1	Авторизація в системі	Логін і пароль користувача	Користувач отримує доступ до акаунту або повідомлення про помилку	Правильно
2	Реєстрація в системі	Дані полів користувача	Користувач отримує доступ до акаунту, який був створений на основі введених даних	Правильно
3	Додавання відкритого проекту	Дані проекту полів	Проект доданий до бази даних	Правильно
4	Редагування проекту	Заміна даних полів обраним користувачем	Заміна даних проекту відповідно до відредагованих полів	Правильно
5	Оновлення стадії проекту	Значення статусу проекту	Заміна стуса проекту	Правильно
6	Написання повідомлень	Дані з полів	Повідомлення заноситься до бази даних	Правильно
7	Фільтрація виведення інформації	Отримання даних з поля для параметрів	Виводиться інформація яка відповідає обраному параметру	Правильно

Участь у наукових конференціях



Розробка також була висвітлена в тезах всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023».

Посилання:

Хміль О.О, Праворська Н.І. Веб-сайт біржі фрілансу. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». Хмельницький. 2023. – 304-307 с.

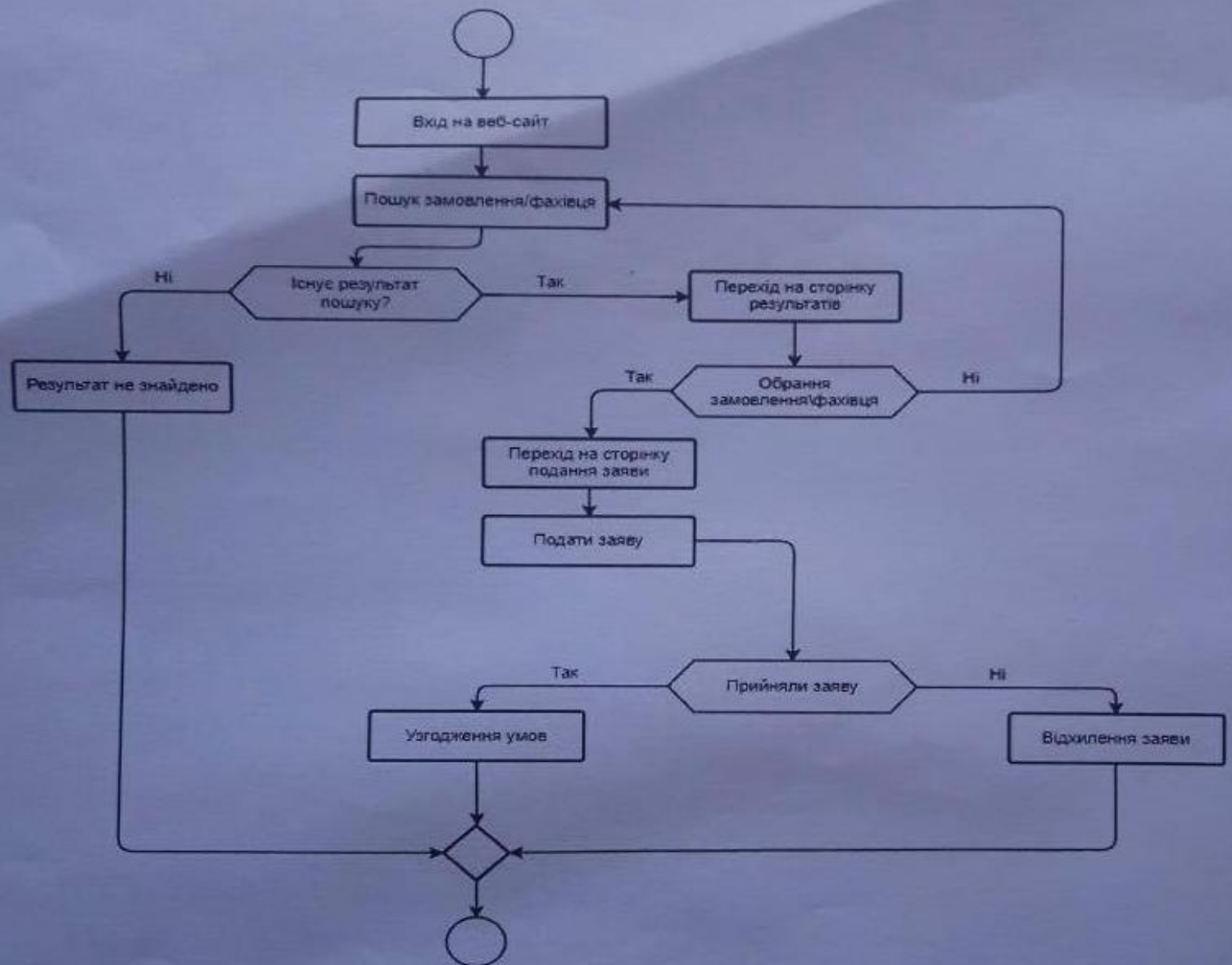


Висновки

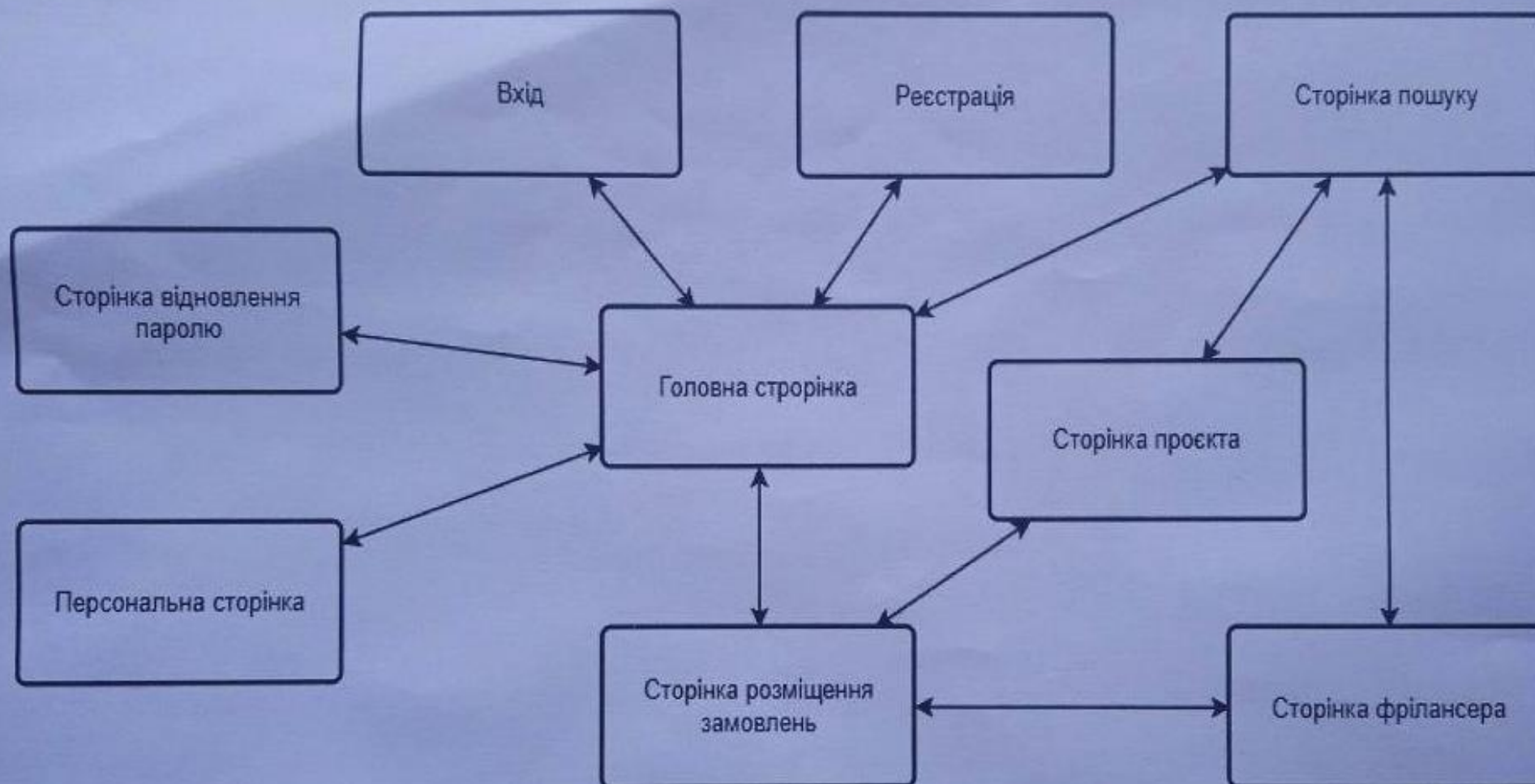
- Виконання кваліфікаційної роботи мало на меті розкрити аспекти проєктування та розробки вебсайта біржі фрілансу.
- Тому у роботі було проведено дослідження предметної області, а також постановку завдань, проаналізовано наявне програмне забезпечення та визначено вимоги до даного вебсайту, спроектовано програмне забезпечення, розроблено і протестовано застосунок та підведено підсумки.
- В роботі було використано фреймворк **Bitrix**, мови програмування **PHP**, **JavaScript** і мова розмітки **HTML** та мову стилю **CSS** з використанням бази даних **MySQL**.
- Підсумовуючи, розроблений вебсайт біржі фрілансу має дружній інтерфейс та відповідає потребам користувачів у даній галузі біржі фріланса. Сайт підтримує різноманітні функціональні можливості та відповідає поставленим системним вимогам.
- Отримані результати демонструють, що розроблений сайт може відмінно виконувати поставлені задачі і функціональні вимоги для користувачів. Такий підхід дозволить підтримувати високу ефективність та забезпечувати комфортні умови для користувачів.

Дякую за увагу!

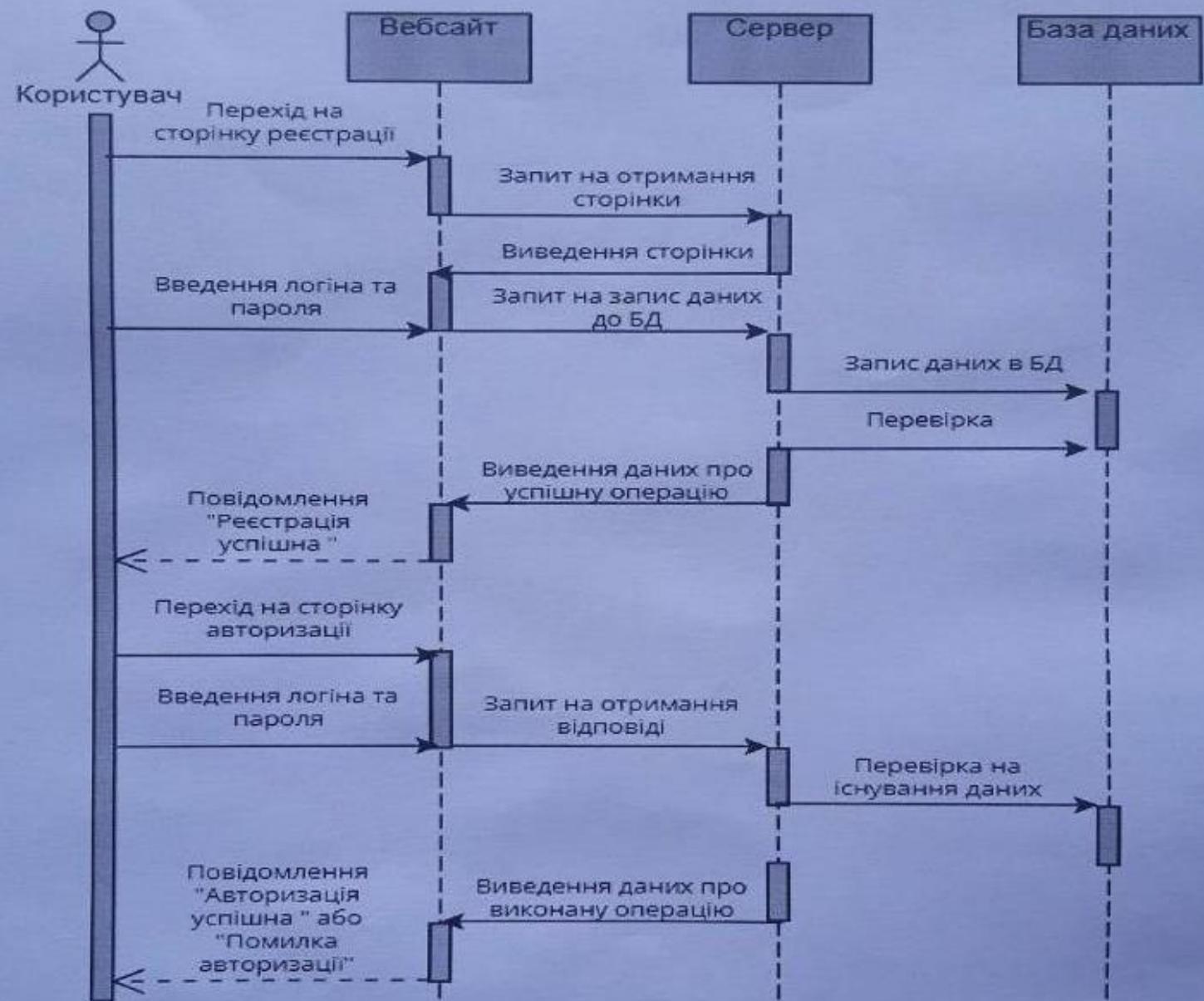
ГРАФІЧНА ЧАСТИНА



					КвРІПЗ. 200175.01.22.E8		
					Вебсайт біржі фрілансу		
Зм.	Арк.	№ докум.	Підпис	Дата	Літера	Маса	Масштаб
Розробив		Хміль О.О	<i>[Signature]</i>	22.05.20			
Керівник		Праворська Н	<i>[Signature]</i>				
Консульт.					Аркуш 1	Аркушів 3	
Н. Контр.		Яшина О.М	<i>[Signature]</i>	22.05			
Зав. каф.		Бедратюк Л.В	<i>[Signature]</i>	22.05			



					КвРІПЗ. 200175.01.22.E8		
					Вебсайт біржі фрілансу		
Зм.	Арк.	№ докум.	Підпис	Дата	Літера	Маса	Масштаб
Розробив		Хміль О.О.	<i>[Signature]</i>	10.08.18			
Керівник		Праворська Н.	<i>[Signature]</i>				
Консульт.					Аркуш	2	Аркушів
							3
Н. Контр.		Яшина О.М.	<i>[Signature]</i>	10.08			
Зав. каф.		Бедратюк Л.П.	<i>[Signature]</i>	2018			



				КвРІПЗ. 200175.01.22.Е8					
				Вебсайт біржі фрілансу			Літера	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата					
Розробка		Хміль О.О.	<i>[Signature]</i>	12.05.16					
Керівник		Праворська Н.І.	<i>[Signature]</i>	12.05.16					
Консульт.					UML-діаграма послідовності авторизації			Аркуш 2	Аркушів 3
Н. Контр.		Яшина О.М.	<i>[Signature]</i>	12.05					
Зав. каф.		Бедратюк Л.В.	<i>[Signature]</i>	12.05					

СУПРОВІДНІ ДОКУМЕНТИ

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Хміль О. О.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-20-1

ЗАЯВА

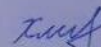
З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті», згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів у роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

29.05.2024

дата



підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 11%

ID: 128468 Назва: БКР Вебсайт біржі фрілансу Додано в БД: 2024-06-05 Автора: Хміль Олег Керівники: Праворська Н.І., канд. пед. наук, доцент Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	98017	785	3687 (4%)	50 (6%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми



Ім'я користувача:
ІПЗ

Дата перевірки:
05.06.2024 15:11:00 EEST

Дата звіту:
06.06.2024 11:25:08 EEST

ID перевірки:
1016322651

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100012953

Назва документа: БКР_Вебсайт_біржі_фрілансу_Хмель_О_О_Праворська_Н_І_

Кількість сторінок: 77 Кількість слів: 14501 Кількість символів: 116915 Розмір файлу: 4.13 MB ID файлу: 1016121087

8.96% Схожість

Найбільша схожість: 3.36% з джерелом з Бібліотеки (ID файлу: 1016114514)

5.21% Джерела з Інтернету 564

Сторінка 79

5.77% Джерела з Бібліотеки 198

Сторінка 83

0.48% Цитат

Цитати 5

Сторінка 84

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 1

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «Бакалавр»

Дипломник Хміль Олег Олегович

Тема Вебсайт біржі фрілансу

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 ; кількість сторінок записки 109

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі досліджено та проаналізовано предметну область, розглянуто існуючі рішення з їхніми перевагами та недоліками, на основі чого визначено вимоги до програмного забезпечення. Розглянуто різні технології, розроблено структуру програмного продукту та прототип користувацького інтерфейсу, що було використано під час розробки програми. Після завершення розробки проведено ряд тестувань для перевірки відповідності вимогам та коректності роботи, за результатами якого можна стверджувати, що програмне забезпечення відповідає поставленим вимогам, працює стабільно і готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до усіх поставлених вимог та завдань.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі було доведено актуальність теми роботи, встановлено мету та основні завдання. У першому розділі було детально проаналізовано предметну область, розглянуто все існуючі рішення з наведенням їхніх переваг та недоліків на основі чого було здійснено постановку задач та вимог. У другому розділі було проведено аналіз, на основі якого для виконання даної роботи було обрано Vitrix Framework з використанням клієнт-серверної архітектури та додаткових технологій. Також було створено прототип інтерфейсу користувача та розроблено структуру проєкту. У третьому розділі було виконано програмну реалізацію з наведенням коду основних модулів у результаті чого було розроблено програмний продукт. Також було проведено ряд тестувань для перевірки на коректність роботи та відповідність вимогам.

4. Позитивні сторони роботи Актуальність теми даної кваліфікаційної роботи було досить детально описано. При аналізі вже існуючих рішень на ринку програмного забезпечення було наведено статистичні данні. При виборі засобів розробки було виконано детальний опис технологій з принципами їхньої роботи.

5. Негативні сторони роботи Реалізована система пошуку проєктів або фрілансерів може бути покращена за рахунок додання додаткових параметрів пошуку. Також доцільно реалізувати систему різноманітних фільтрів для кращого та зручнішого перегляду статистичних даних.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано у вигляді різноманітних рисунків та діаграм та відповідають темі кваліфікаційної роботи. Пояснювальна записка виконана відповідно до вимог та чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивну оцінку. Матеріал пояснювальної сторінки структурований, послідовний та детально описаний, що дозволяє зрозуміти увесь викладений матеріал відповідної тематики, а наведений графічний матеріал наочно демонструє усі аспекти виконаної роботи.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі відповідно до усіх поставлених вимог та завдань і заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ Мартинюк Валерій Володимирович, доктор технічних наук, професор зав. кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки ХНУ.

“ 6 ” 06 2024 р.



(підпис)

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Вебсайт біржі фрілансу»

Автор: Хміль Олег Олегович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Праворська Наталія Іванівна, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) запозичення, виявлені в тексті роботи, є фрагментарними.

Максимальний обсяг запозичень, визначений системою Anti-Plagiarism, складає 2.0%. Обсяг запозичень, визначений системою Unichек виявлення збігів ідентичності/схожості, складає 8.96% і адресується до 564 джерел з Інтернету і 198 джерел з бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 4.06.2024

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Наталія ПРАВОРСЬКА