

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

ДИПЛОМНИЙ ПРОЕКТ

Інтернет-платформа для надання перукарських послуг

Назва теми

Рівень вищої освіти Перший(бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

Шифр ДППЗ.190153.19.06.ПЗ

Виконав студент III курсу група ПЗс-19-1



О.В. Лисюк
Ініціали, прізвище

Керівник к. т. н., доцент
Науковий ступінь, звання



Ю.В. Форкун
Ініціали, прізвище

Нормоконтролер к.т.н. доцент



І.В. Гурман
Ініціали, прізвище

До захисту допускаю:
Завідувач кафедри інженерії
програмного забезпечення



Л. П. Бедратюк
Ініціали, прізвище

7 06 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри 193
Л. П. Бедратюк
05 02 2022 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)**

Лисюку Олегу Валентиновичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інтернет-платформа для надання перукарських послуг

Керівник проекту (роботи) Форкун Юрій Вікторович, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, місце роботи

Затверджена наказом ректора університету від 05.02.2022 р. № 11

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2022 р.

3. Вихідні дані до проекту (роботи) Матеріали переддипломної практики





4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація, тестування програмного забезпечення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Презентаційні матеріали (слайди, 14шт.)

6. Консультанти розділів дипломного проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання
Нормоконтроль	Гурман І.В., доцент кафедри ПЗ		
Антиплагіат	Гурман І. В., доцент кафедри ПЗ		

7. Дата видачі завдання « 05 » лютого 2022р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	При
1 Ознайомлення з тематикою дипломного проектування (ДП), визначення та узгодження індивідуальних тем ДП	01.12 – 30.12.2021	
2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2022	
3 Проектування програмного забезпечення	01.02 – 28.02.2022	
4 Програмна реалізація	01.03 – 10.04.2022	
5 Тестування програмного забезпечення	11.04 – 30.04.2022	
6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів	01.05 – 25.05.2022	
7 Попередній захист ДП	Травень 2022 (згідно графіка)	
8 Перевірка ДП на плагіат, нормоконтроль, отримання відгуків та рецензій. Брошування (зшиття) пояснювальної записки	26.05 – 30.05.2022	
9 Підготовка до захисту та захист ДП	з 01.06.2022	

Студент


Підпис

О.В. Лисюк
Ініціали, прізвище

Керівник проекту (роботи)


Підпис

Ю.В. Форкун
Ініціали, прізвище

АНОТАЦІЯ

Тема дипломного проекту: Інтернет-платформа для надання перукарських послуг.

Автор проекту: Лисюк Олег Валентинович.

Керівник проекту: Форкун Юрій Вікторович.

Пояснювальна записка: 96 с., 28 рис., 2 табл., 3 дод., 11 джерел.

Графічна частина: 15 слайдів.

TELEGRAM, BOT, E-COMMERCE, PHP, LARAVEL, MYSQL, SQL, API, HOOK, MVC.

Метою проекту є розробка програмного забезпечення, яке дозволить його користувачам полегшити роботу в перукарському бізнесі а саме онлайн запис у перукарню, а також полегшить облік замовлень для адміністратора системи. Система має сучасний, інтуїтивно-зрозумілий користувачам інтерфейс.

Під час розробки веб-сайту:

- проведений аналіз предметної області;
- проаналізовані методики та засоби створення веб-сайтів;
- розроблено веб-сайт, що виконує поставлені задачі.

Для розробки додатку була використана мова програмування php, фреймворк Laravel та база даних MySQL.

В результаті було розроблено веб-сайт для певної перукарні, у якій можна переглядати різноманітну інформацію про заклад, а саме – ціни, послуги, фото і т.д; а також онлайн запис на візит.

30.05.22
Дата


Підпис

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	ДПШЗ.190153.19.06.ВД	Пояснювальна записка	96		
2	A4		Завдання на дипломний Проект	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4		Презентаційні матеріали	15		

					ДПШЗ.190153.19.06.ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Інтернет-платформа для надання перукарських послуг	Літ.	Арк.	Аркуша
Виконав		Лоско О.В.		5.06			1	96
Керівник		Фархун Ю.В.		2.06	Пояснювальна записка	ХНУ, ПТЗс-19-1		
Н. Контр.		Гурман І.В.		2.06				
Зав. Каф.		Бедрачок Л.П.		8.06				

ЗМІСТ

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 21

1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання..... 26

2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ 29

2.1 Архітектура та функціональна структура сайту 29

2.3 Проектування серверної частини веб-сайта..... 41

2.4 Проектування клієнтської частини веб-сайта 45

2.5 Аналіз та вибір технологій і методів реалізації веб-сайту..... 50

3 ПРОГРАМНА РЕАЛІЗАЦІЯ 53

3.1 Розробка бази даних..... 53

3.2 Розробка програмних модулів 54

3.3 Керівництво користувача 62

3.4 Технічні характеристики інтернет-платформи 65

3.5 Розгортання та встановлення системи..... 66

4 ТЕСТУВАННЯ ВЕБ-ДОДАТКА..... 67

4.1 Аналіз методів тестування веб-додатка..... 67

4.2 Розробка тестів 69

ВИСНОВКИ 72

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 73

ДОДАТОК А..... 75

ДОДАТОК В..... 80

ДОДАТОК Г 89

					ДППЗ.190153.19.06.ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Інтернет-платформа для надання перукарських послуг	Літ.	Арк.	Аркуші
Виконав		Лисок О.В.	<i>[Підпис]</i>	5.06			1	96
Керівник		Фархун Ю.В.	<i>[Підпис]</i>	7.06	Пояснювальна записка	ХНУ, ІПЗс-19-1		
Н. Контр.		Гурман І.В.	<i>[Підпис]</i>	7.06				
Зав. Каф.		Бедратюк Л.П.	<i>[Підпис]</i>	8.06				

ВСТУП

У сучасному світі програмування починає відігравати надважливу роль, так як модернізація триває і суспільство переходить на новий рівень життя. Розвивається приватне підприємництво, в людей змінюються перспективи, спокійний спосіб життя замінюється швидким і виникає проблема постійного браку особистого часу.

Перукарська справа виходить на передній план, так як передбачає зручність, швидкість і якість обслуговування для постійно зайнятого населення, адже потреба в перукарських послугах була є і буде завжди. І тому потрібно надавати увагу тому, що буде актуальним у майбутньому.

На сьогоднішній день вже написано немало сайтів для полегшення праці в даному бізнесі, але немає такого сайту, який зміг би замінити декілька етапів в обслуговуванні і надати клієнту можливість самостійно робити вибір виду полуги, робити вибір барбера та створювати замолення, не роблячи при цьому жодних зусиль.

Саме тому виникла ідея написати даний сайт.

Метою даного проекту є розробка програмного забезпечення, яке дозволить його користувачам полегшити роботу в перукарському бізнесі а саме онлайн запис у перукарню, а також полегшить облік замовлень для адміністратора системи.

Виходячи з поставленої мети, були поставлені такі завдання: аналіз обраної тематичної області; порівняння доступних аналогів веб-сайтів; визначення програмної платформи, а саме фреймворків та інтерфейсу прикладних рішень та бібліотек API з великою кількістю доступних інструментів: функцій, класів, структур, які підходять для розробки сучасних веб-сайтів з інтеграцією телеграм.

Ціль даного розроблення – зробити зрозумілий і доступний для кожного користувача сайт для роботи із перукарськими послугами.

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					6

Він надає нові можливості як для гостя перукарні, так і для адміністрації.

Зі сторони відвідувача програма максимально спростила процес створення замовлення, надала повну інформацію про послугу і ціну про нього та можливість замінити дане замовлення у телеграм-боті.

Сайт буде містити декілька вкладок таких як:

- головна;
- послуги;
- про нас;
- ціни;
- барбери;
- контакти;
- галерея;

На кожній сторінці буде відображена різна інформація стосовно названою сторінки. Послуги та ціни розміщені на окремих сторінках. В свою чергу ці сторінки розділені для зручності у пошуку. Користувачу буде дуже зручно на початку подивитись усі послуги які має ця перукарня, почитати різноманітну інформацію про них, а також після того дізнатись усі можливі ціни які пропонує дана перукарня.

При створенні замовлення великим плюсом є те, що користувач бачить суму даного замовлення і може контролювати її.

Описана вище процедура повністю звільняє від прийому замовлення адміністратором чи перукарем, й звільняє від пояснень про послуги та ціни, якщо вони виникають.

Це спрощує процес планування роботи перукарів.

Коли до візиту у перукарню залишається 12 годин, користувачу приходить сповіщення у телеграм бот, про візит, що він був записаний на певну дату і певний час у перукарню.

Робота перукаря зводиться до того, що він просто проводить перукарські послуги й приводить до порядку робоче місце, як гість покинув цей заклад.

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					7

При звичному для нас обслуговуванні, коли замовлення приймає сам перукар інколи спрацьовує «людський фактор» і відбуваються помилки у пропонованій ціні. В нашому випадку програма знімає з перукаря частину відповідальності, так як він більше не є пунктом передачі інформації, а є просто тим, хто виконує певне замовлення на певну послугу тобто зачіску.

Зі сторони адміністрації також можна виділити безліч корисностей. Просто тримаючи в руках телефон, директор, перукар, чи адміністратор може змінювати, передивлятис, редагувати різноманітну інформацію на всьому сайті, а самим великим плюсом є те, що директор зможе переглядати, всі замовлення і підбивати підсумок скільки виконав перукар замовлень на день.

Якщо буде необхідність змінити щось в послугах, чи цінах, легко можна редагувати всю різноманітну інформацію про певну послугу та ціни, не потрібно передруковувати масу паперу - гігантська економія і часу, і коштів.

Директор / адміністратор матиме постійний доступ до сайту та адмін частини. В любий час можна буде додати щось нове у сайт, не прикладаючи до цього великих зусиль. І після цього користувач зможе побачити якусь новинку яку пропонує ця перукарня.

І при цьому не обов'язково знаходитись на своєму робочому місці.

Тому було вирішено використати дану тему як тему дипломного проекту, тому що це актуально, необхідно і при написанні данного сайту показані навички, які були освоєні за процес навчання.

									Арк.
									8
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					

декількох веб-сторінок, що містить основну і найважливішу інформацію про організацію чи людину. На такий вид сайту зазвичай не ставиться завдання створити платформу для спілкування користувачів або замовлення послуг, а покладається завдання про інформування потенційних клієнтів. Як правило в сайтах візитках не зустрічаються форми для реєстрації, форми замовлення товару і інше. Зазвичай там є такі блоки як: інформація про компанію, її послуги, портфолію, адреса, контакти, посилання на соціальні мережі, форма відгуку та форма зворотнього зв'язку. Основною відмінністю сайту-візитки є його призначення, оскільки він призначений для того, щоб надати потрібну інформацію о компанії, а також пропозиції які вона виставляє. Для того щоб створити дійсно якісний та привабливий сайт-візитку необхідно брати до уваги особливості галузі в якій компанія здійснює свою діяльність, чітку позицію фірми щодо свого бренду, а також логотип, унікальну крилату фразу і найголовніше – унікальний дизайн проекту.

– Інтернет каталог - Основне призначення інтернет-каталогу – зручна і доступна подача конкретної інформації у великому обсязі. Зазвичай має вигляд каталогу. На відміну від переліку товарів, розташованих на інших інтернет-ресурсах, характеризується більш детальною розробкою, з обов'язковою функцією пошук по сайту. За іншими підрозділами, в основному не дуже відрізняється від корпоративного веб-проекту. Тут можуть бути – новинний блок, статті по основній сфері діяльності, відгуки і так далі. Дизайн інтернет-каталогу максимально націлений на вирішенні бізнес-завдань і задоволення інтересів цільової аудиторії.

– Інтернет магазини – це веб ресурс, сайт з певним каталогом, за допомогою якого відбувається безпосередній продаж товарів споживачеві з урахуванням доставки. При цьому розміщення інформації, замовлення товару і договір відбуваються прямо на сайті магазину. Користувач може прямо на сайті вибрати і замовити потрібні йому товари з каталогу. Тобто обов'язковою ознакою інтернет- магазину є можливість оформлення онлайн замовлення.

										Арк.
										10
Зм.Арк		№ докум.	Підпис	Дата						

Важливі елементи інтернет-магазину:

- оновлення наявного асортименту (продукти/товари та їх кількість);
- можливість додавати товари до «кошику»;
- вхід для зареєстрованих користувачів.

– Новинні та пошукові портали - такими сайтами ми користуємось щодня, коли хочемо дізнатися щось нове або бути в курсі останніх новин. Як правило, такі сайти є дуже великими й, відповідно, вимагають багато часу на розробку та просування. Зазвичай, це комерційні проекти.

– Інформаційні портали - це сайти, на яких зібрані різного роду статті на будь-які теми. Сайт може містити безліч інформаційних статей на всі тематики або ж декілька та розповідати про новинки в світі технологій. Також вони можуть об'єднувати та структурувати в собі інформацію, зібрану з інших ресурсів. Найчастіше інформаційні портали існують за рахунок реклами у вигляді банерів або спонсорів.

– Веб-системи обліку товарів, бухгалтерії - подібні сайти призначені виключно для внутрішнього використання будь-якою фірмою й доступ до них для звичайних користувачів закритий. Вони дуже важливі для структурування даних та використовуються зазвичай дуже великими компаніями, які мають безліч філій в різних містах. Такі сайти містять досить багато інформації й коштують дуже дорого, проте їх зручність в разі окупає ціну.

– Landing page – це новий формат сайту веб-сторінка на вашому піддомені або домені, основною метою якої є спонукання користувача зробити цільову дію. Це може бути оформлення заявки на:

- покупку товару;
- безкоштовну консультацію;
- оформлення броні;
- завантажити прайс;
- прорахувати кошторис або вартість якихось послуг;
- участь в акції;

									Арк.
									11
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					

– передплату на розсилку і так далі.

Вперше landing page почали використовувати в США, і з тих пір популярність такого веб-ресурсу тільки зростає. Це обумовлено його високою ефективністю. Характерною особливістю такого інтернет-ресурсу є відсутність складної навігації, об'ємних текстів, зайвих меню, постійно спливаючих банерів.

На рисунку 1.1 зображено класифікацію веб-ресурсів.



Рисунок 1.1–Вебресурси

Зм.Арк	№ докум.	Підпис	Дата	

документ повертається на сервер і після цього відображається в браузері. Java, PHP і інші мови програмування використовуються для інтерпретації сторінок на серверній стороні.

MVC – ключ до розуміння розробки динамічних веб-додатків. Тому розробнику необхідно знати цю модель.

MVC розшифровується як Model-View-Controller. Це дозволяє організувати свій код, вибираючи блоки, що відповідають за вирішення різних завдань. Один блок відповідає за дані додатку, інший - за зовнішній вигляд, а третій контролює роботу додатка. Компоненти MVC:

- модель, що відповідає за дані, а також визначає структуру програми;
- перегляд, який відповідає за взаємодію з користувачем, тобто код компонента уявлення визначає зовнішній вигляд програми і те, як воно використовується взагалі;
- контролер, що відповідає за відносини між моделлю і представленням, визначає, як сайт реагує на дії користувача і по суті є мозком додатку MVC.

Прикладами динамічних сайтів є сторінки, створені на основі систем управління контентом (CMS). З них найбільш поширені інтернет-магазини, а також форуми, сторінки з відгуками і інші ресурси з можливістю розміщення контенту відвідувачами.

Досліджуючи веб-ресурси за видовими ознаками, їх можна поділити на:

- комерційні;
- корпоративні;
- інформаційні.

Ресурси комерційного типу є найбільш поширеними серед інших видів ресурсів. Основна мета їх створення – продаж товарів користувачам мережі.

Ресурси корпоративної спрямованості створені для виконання одного з наступних завдань: підвищити імідж компанії в очах існуючих і потенційних клієнтів або зміцнення зв'язку між відділеннями та філіями організації.

Інформаційні ресурси призначені для надання користувачам

									Арк.
									14
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					

Навігація – це інструмент веб-дизайну, який дозволяє користувачеві легко і ефективно переміщатися по розділах веб-сайту. Навігація позбавляє користувача від необхідності знати URL-адресу окремих розділів – саме тому сайтів без навігації майже не буває, адже такі портали неможливо використовувати хоч скільки-небудь продуктивно. Навігація може бути реалізована як у вигляді меню, так і у вигляді гіперпосилання.

Меню – це сгруперовані гіперпосилання, які зазвичай визначають окремі розділи сайту. Кожне посилання – це назва такого розділу. Для сайту меню є одним з найважливіших елементів веб-ресурсу, який безпосередньо впливає на популярність сайту в пошуковій системі.

Меню може містити додаткові меню, в яких також вказують на стосункам розділів – ієрархію сайту. Щоб правильно скласти меню для сайту, важливо враховувати особливості типу цього сайту, його тематики, обсягу та ін. Зазвичай для меню розробляється окремий дизайн. На різних сайтах меню реалізовано по-різному: десь це звичне для всіх вертикальне або горизонтальне текстове меню, а десь – нестандартне флеш-меню або меню в картинках. Чинники, що впливають на вибір того чи іншого рішення, дуже різноманітні.

Види меню сайту:

- головне меню, яке відображається в будь-яких розділах ресурсу;
- другорядне меню, що використовується у випадку великих інтернет-проектів з багатьма сторінками, де одного рівня навігації недостатньо;
- випадаюче меню, що використовується для економії простору екрану та з'являється при наведенні курсору;
- меню, що розкривається при натисканні на категорію;
- вертикальне меню, яке знаходиться в лівій або правій частині сайту.
- горизонтальне, яке знаходиться у верхній частині сайту, тобто в шапці.

Навігація за гіперпосиланнями – це тип навігації який забезпечує перехід від одного розділу до іншого. В цьому випадку посиланням може бути слово з головної частини сайту, значок або зображення, яке тематично пов'язане з

іншим розділом сайту. Для гіперпосилань також передбачено окремий дизайн та оформлення, але він, як правило, простий, інакше текст, в який вставлене посилання, буде виглядати негармонійно.

Також час від часу відвідуючи різноманітні сайти, я потрапляв на сторінки у які є під'єднані телеграм боти, але у цих телеграм ботів, немає ніякого функціоналу, а просто опис різноманітних продуктів.

Телеграм боти – це спеціальні акаунти в Telegram, які можуть автоматично обробляти і відправляти повідомлення. Вони створюються програмістами і працюють у них на сервері.

Боти можуть виконувати практично будь-які завдання, які може робити кожен користувач акаунту Telegram з онлайн-сервісами. Роботи можуть вчити, розважати, шукати, транслювати, нагадувати, з'єднувати та підключатися до інтернету речей. По суті, боти – це такий собі зручний для людини інтерфейс роботи з різноманітними веб-службами.

Натомість на сьогоднішній день, Телеграм боти дуже популярні, але нажаль їх не так часто використовують у будь яких сайтах. Тому я і вирішив зробити щось цікаве та обрав саме створення сайту та підключення до нього телеграм бота.

Зваживши переваги і недоліки кожного типу вебсайтів я вирішив обрати саме сайт компанії з каталогом продукції, адже він забезпечує найкращий варіант щодо моєї теми дипломного проекту.

Тепер, після вибору класифікації сайту для реалізації, потрібно дізнатись все про перукарські послуги. Перукарські послуги були важливі з давніх давен, тому що людина хоче виглядати гарно, а натомість їй потрібно йти в перукарню та приводити себе в порядок. Кожна людина раз в місяць обов'язково має відвідати салон перукарських послуг. Але нажаль є дуже мало сайтів які пов'язані особливо з перукарськими послугами.

Отже найдоречніше створити сайт який буде містити різноманітні послуги та ціни, і в любий момент користувач зможе зайти, та подивитись що йому

										Арк.
										20
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ						

найбільш підходить, і при тому йому не потрібно буде телефонувати в перукарню, та дізнаватися прайс, а також, в нього буде можливість записатись онлайн на візит в перукарню, це набагато полегшить роботу користувача

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Ресурси сайтів перукарських послуг є обширними. Тому у цьому розділі будуть відображені сайти які є подібними до проекту реалізованого під час роботи над дипломним проектом.

BarbershopKhm (Рисунок 1.4) - Це односторінковий лендінг сайт, в якому вся інформація розміщена на одній сторінці. Це неповноцінний продукт а лише логічне продовження рекламного оголошення або посилання, які часто пов'язані з соціальними медіа, розсилками електронною поштою або маркетинговими компаніями, з метою підвищення ефективності реклами. Це ефективний вид маркетингу та інструмент для рекламних кампаній. Посадкова сторінка підштовхує користувачів до певної дії – оформити передплату на безкоштовний пробний період, забронювати послугу або зробити покупку. Для цього використовують спеціальні кнопки із закликом до дії: наприклад, «Почати», «Підписатися» або «Купити».

Зазвичай клієнти дізнаються про посадкову сторінку з розсилки або онлайн реклами, знаходять її в соцмережах або в результатах пошуку в Google або Yandex.

Лендінг це інструмент генерування лідів та охоплення потрібних клієнтів. Використовуючи лендінг, можна отримати уявлення про результативність маркетингової стратегії. Він містить привабливий та адаптивний дизайн, на який приємно дивитись користувачу, коротку інформацію про послуги, та контактні дані мережі.

Тому можна зробити висновок, що лендінг сайт, не надає усієї повноти

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					21

гортати вміст не тільки вниз, але і вбік. Адаптивний дизайн виправляє цю проблему, роблячи сайт однаково зручним для перегляду з будь-яких пристроїв.

Сьогодні більше 50% всіх покупок в інтернеті роблять зі смартфонів і планшетів. Все тому, що мобільний пристрій завжди під рукою.

Ігнорувати таку кількість мобільного трафіку недоцільно – так можна втратити половину всіх своїх покупців. Але потрібно подбати, щоб їм було зручно вивчати каталог, додавати товари в кошик, оплачувати замовлення Для цього і потрібно адаптивний дизайн сайту – щоб він автоматично підлаштовувався під розмір вікна браузера на будь-якому смартфоні, планшеті або персональному компютері.

Актуальні новини люди сьогодні отримують не з газет, а знову-таки зі своїх смартфонів. Адже інших пристроїв під рукою зазвичай немає. Тому, якщо на сайті міститься новинна або інша термінова інформація – він обов'язково повинен мати адаптивний дизайн.

Google давно вважає наявність адаптивного дизайну одним з ключових чинників ранжирування. Тобто, адаптивні веб-ресурси отримують більш високі позиції, ніж майданчика зі звичайним дизайном. Тому його наявність – це вже не рекомендація, а обов'язкова умова.

Всього існує 5 основних типів адаптивного веб-дизайну:

- гумовий макет - цей спосіб набув найбільшого поширення. Сайт адаптується за рахунок стиснення блоків по ширині до розмірів екрану мобільного пристрою. А блоки й які не стискаються, шикуються вниз;

- перенесення блоків - такий варіант підходить для сайтів з многоколончатою структурою. Якщо блоки не поміщаються на екрані – вони просто переносяться вниз;

- розробка макетів - під кожний окреме дозвіл екрана розробляються макети. Метод дуже трудомісткий, тому для простих сайтів не використовується (для них досить просто переносити блоки). Зате він застосовується для бізнес-сайтів, корпоративних сайтів, лендінгів і особливо для інтернет-магазинів. В

даному випадку розробляються макети під всі основні дозволи екрану, так як тут потрібно акцентувати увагу користувача на важливих блоках і вести його до потрібного нам цільовим дії;

- масштабування тексту і картинок - відмінний варіант для найпростіших сайтів. Масштабується не весь веб-ресурс, а тільки деякі блоки. Дане рішення особливою гнучкістю не володіє, тому теж популярністю не користується;

- панелі - цей варіант прийшов з мобільних додатків. Додаткове меню тут з'являється при горизонтальному або вертикальному натиску. І це теж не завжди зручно, оскільки вимагає додаткових дій від користувача;

При проектуванні дипломного проекту буде врахований цей недолік, який був знайдений у аналогах, тому буде закладений адаптивний інтерфейс із методом перенесення блоків.

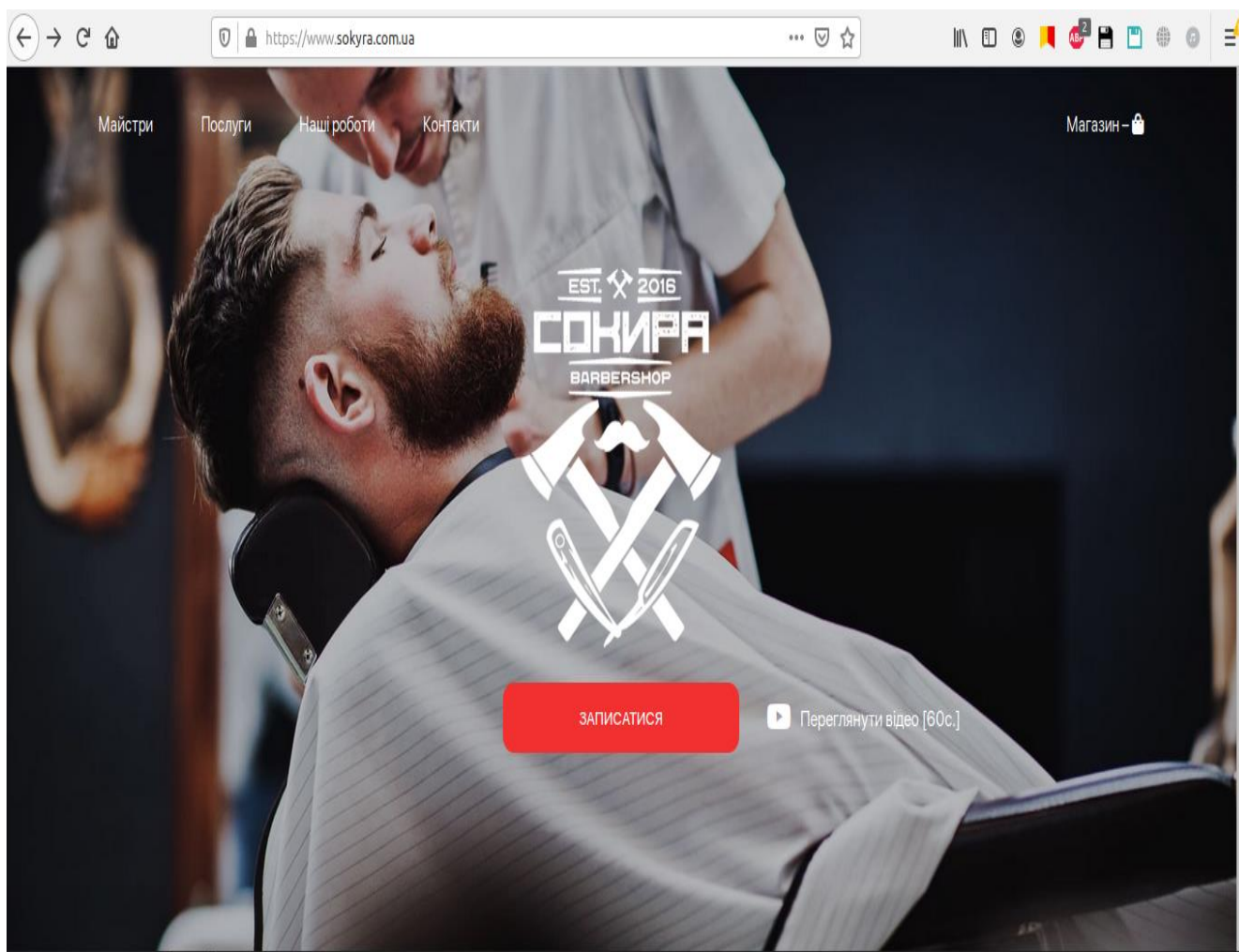


Рисунок 1.5 – інтерфейс сайту Сокира Barbershop.

									Арк.
									24
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					

цей сайт на перший погляд більш функціональніший, а насправді це тільки здається. При взаємодії із сайтом стає зрозуміло, що при розробці ui і ux було упущено багато важливих дрібниць.

UI Design (User Interface) перекладається як «призначений для користувача інтерфейс». Інтерфейс являє собою графічну структуру програми. Він складається з кнопок, натиснених користувачами, текстів, які вони читають, зображень, полів введення тексту і всіх інших елементів, з якими взаємодіє користувач. Крім того, він включає в себе макет екрану, переходи, анімацію інтерфейсу і кожен мікро-взаємодію. Будь-який візуальний елемент, взаємодія або анімація повинні бути розроблені в процесі UI-дизайну.

Ця робота лягає на дизайнерів користувацького інтерфейсу (UI-дизайнер). Вони вирішують, як візуально буде виглядати додаток. Саме вони повинні вибирати кольорні схеми і форми кнопок - ширину ліній і шрифти, які використовуються для тексту. Якщо узагальнити, то дизайнери користувацького інтерфейсу створюють зовнішній вигляд інтерфейсу програми.

UX (User Experience) перекладається як «користувацький досвід». Призначений для користувача досвід визначається тим, як користувачі взаємодіють з додатком/сервісом. Чи є досвід користувача інтуїтивно-зрозумілим або, навпаки, заплутує його? Навігація в додатку здається логічною чи не піддається логіці? Чи дає людям взаємодія з додатком відчуття, що вони ефективно виконують завдання, які вони поставили перед собою, або це схоже на блукання по замкненому колу і відчуття безвиході?

Досвід користувача визначається тим, наскільки легко чи складно взаємодіяти з елементами призначеного для користувача інтерфейсу, створеними дизайнерами.

При цьому UX-дизайнери також дбають про користувацький

										Арк.
										25
Зм. Арк		№ докум.	Підпис	Дата						

інтерфейс програми, і саме тому люди часто плутаються, або ж взагалі не знаходять різниці між ними.

При аналізі помилок було прийнято рішення більш детально дослідити дизайн сайтів які пропонують перукарські послуги для побудови максимально зручного користувацького інтерфейсу.

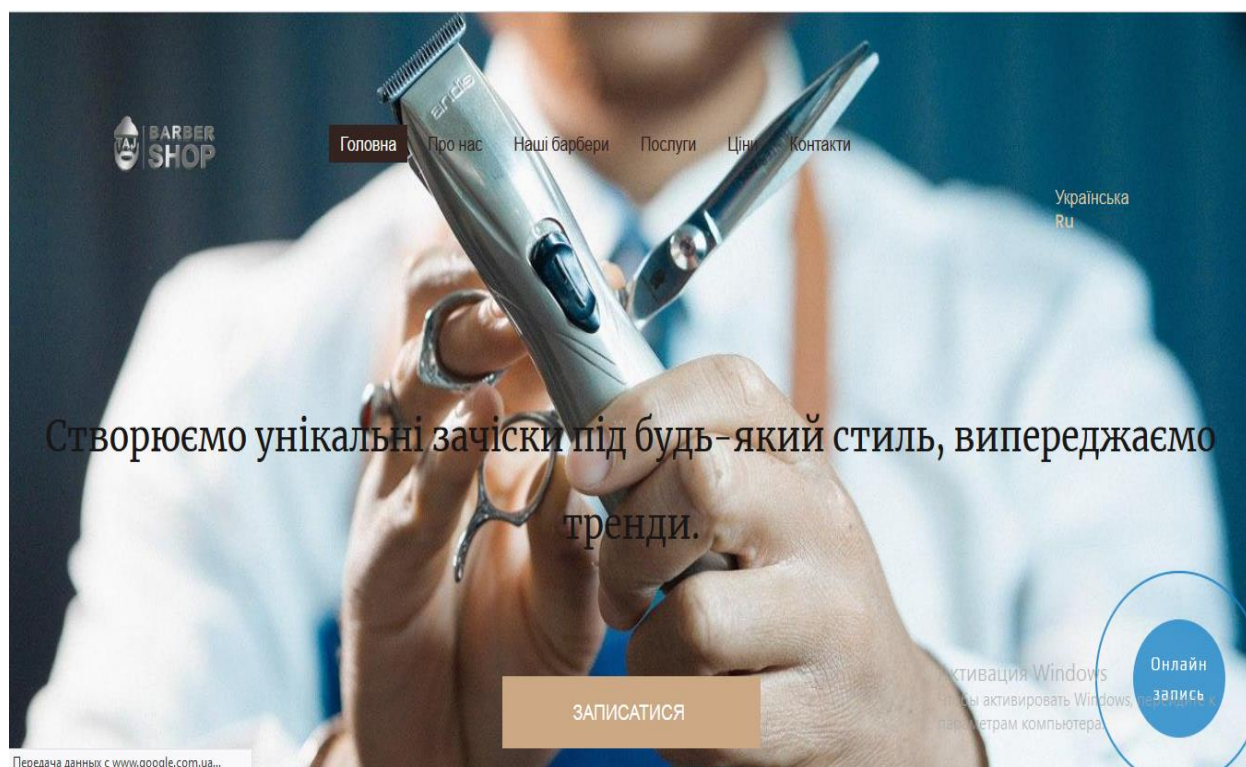


Рисунок 1.6 – інтерфейс сайту TejBarbershop.

Проаналізувавши ринок сайтів, які надають перукарські послуги, зрозуміло, що він обширний та має немалу кількість конкурентів. Відвідавши багато сайтів, було зрозуміло, що не вистачає простоти та функціоналу у більшості сайтах.

Тому було прийнято рішення реалізувати дипломний проект на тему “Інтернет платформа перукарських послуг з підключенням телеграм бота”, який буде містити вище описаний функціонал. Саме сайт буде корисним для такої теми, тому що не потрібно буде скачувати різні додатки, які ще до всього будуть займати ваше місце на дивайсі.

1.3 Визначення вимог до програмного забезпечення тарозробкатехнічного завдання

Після аналізу предметної області та наявного програмного забезпечення було на відміну від наявних на ринку продуктів, розроблюваний вебсайт має унікальну можливість запису онлайн та подальше його опрацювання та відслідковувати його перебіг у телеграм боті. Користувач на сам перед може переглянути всі послуги та ціни на них, також може обрати любого барбера який йому до вподоби, подивитись різноманітні фото, після чого користувач може зробити запис у перукарню, обираючи певну послугу, потрібного барбера, дату та час який буде зручний як для користувача так і для працівника. Після чого на його пошту прийде повідомлення з детальною інформацією про його замовленням, та силкою і кодом для переходу у телеграм бот.

У свою чергу телеграм бот міститиме можливість опрацювати раніше зроблений заказ. Це спеціальні акаунт в Telegram, який може автоматично обробляти і відправляти повідомлення, в якого серверна частина міститься на стороні клієнта, а інтерфейс виступатиме усім відомий додаток Telegram.

Він може виконувати практично будь-які завдання, які може робити кожен користувач акаунту Telegram з онлайн-сервісами. Роботи можуть вчити, розважати, шукати, транслювати, нагадувати, з'єднувати та підключатися до інтернету речей, проте в даному випадку, користувач зможе редагувати замовлення, а саме (дату, час, послугу та барбера). Також на серверній частині сайту буде реалізовано планувальник команд, який дозволяє плавно й виразно визначати розклад команд у самій програмі. Це допоможе реалізувати завдання з нагадування про прийдешній візит у певну період часу.

Інтерфейс користувача має бути максимально зручний та інтуїтивним.

Також невідомою частиною кожного сайту є адмін панель. Також відома як адмінка сайту або бекенд – це графічна панель правління сайтом. За допомогою адмін панелі в дипломному проекті буде реалізовано:

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					27

- a) налаштування структури сайту;
- b) зміна зовнішнього оформлення сайту;
- c) додавання та редагування контенту, а саме:
 - 1) додавання видалення та редагування цін;
 - 2) додавання видалення та редагування барберів;
 - 3) додавання видалення та редагування послуг;
 - 4) додавання видалення та редагування фото;
 - 5) налаштування з'єднання з TelegramAPI а саме:
 - створення веб хука на користувацький домен;
 - видалення раніш створеного хука;
 - отримання детальної інформації про вебхук;
 - збереження та налаштування поточного домену сайту;
- b) перегляд та редагування замовлень;

Невідомою частиною роботи в адмін панелі є захист доступу до неї, в даному дипломному проєкті, буде реалізована авторизація через JWT токен, а також система ролей з можливістю менеджменту користувачів та їх доступами до окремих модулів сайту.

Враховуючи всі вищезазначені особливості, можна скласти перелік функціональних можливостей, які має надавати програма:

- перегляд послуг;
- перегляд цін;
- перегляд фото по різних категоріях;
- перегляд барберів та короткої інформації про них;
- онлайн запис з відправкою детальної інформації на пошту користувача;
- редагування замовлення у телеграм боті;
- нагадування про замовлення у телеграм боті;

Також було сформовано такі вимоги до вебсайту:

- стримані та базові кольори (сірий, білий, темно сірий), які лежать в основі дизайну сайту.

									Арк.
									28
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					

розуміння архітектури, бізнес-процесів клієнта і багато чому іншого. При цьому, створюючи сайт з нуля, клієнт отримує унікальний і персоналізований продукт, який буде вирішувати його завдання і не витратити час на зайві процеси. Самостійна розробка дозволяє створювати проекти будь-якої складності і з будь-яких побажань клієнта.

Кожна мова має свої особливості, сфери застосування і вирішує свої задачі.

На рисунку 2.1 зображено рейтинг мов написання веб-сайту.

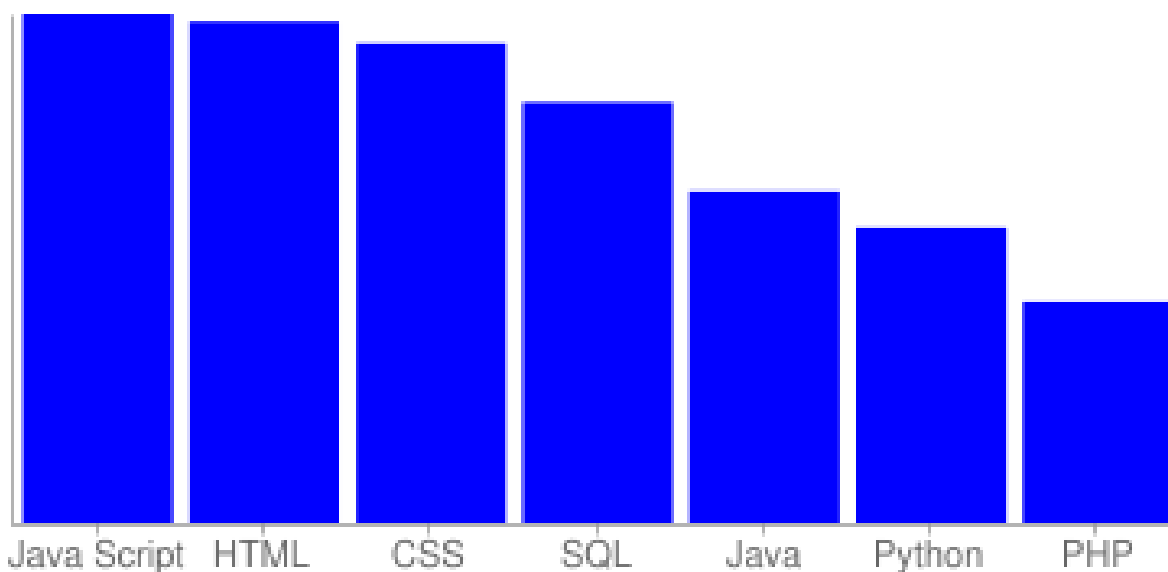


Рисунок 2.1– Рейтинг мов програмування веб-сайтів

Для створення зовнішнього вигляду сайту, який буде видно кінцевим користувачам, на сьогоднішній день існують дві базові мови: HTML і CSS.

HTML – це мова розмітки гіпертексту. Вона дозволяє користувачеві створювати і структурувати розділи, параграфи, заголовки, посилання і блоки для веб-сторінок і додатків.

HTML-сторінка – це файл, який містить в собі код, створений з використанням мови HTML.

HTML надає наступні можливості:

– видавати мережеві документи з заголовками, текстом, таблицями, списками, фотографіями;

відносяться бізнес-об'єкти, які містять в собі логіку збереження й отримання інформації, із якою працює додаток. Іноді його ототожнюють із шаром доступу до даних (DAL).

Представлення (View) – це компонент, який відповідає за відображення користувацького інтерфейсу. Представлення можуть відображати дані отримані від моделі.

Контролер (Controller) – це компонент, який містить всю логіку конкретного екрану (сторінки додатку). Контролер обробляє дії користувача і потім оновлює модель і/або представлення.

Даний шаблон ще називають пасивним представленням MVC. Існує також варіація цього шаблону, яка називається MassiveViewController. Основна відмінність полягає у тому, що діяльність (екран додатку) разом із її логікою потрібно розглядати як шар відображення, а для контролера необхідно створювати окремий клас.

Загальна схема шаблону MVC показана на рисунку 2.3.

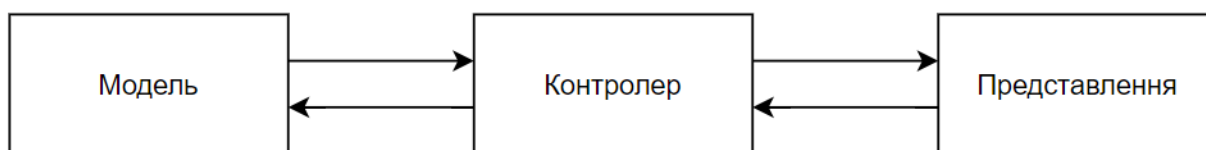


Рисунок 2.3 – Схема шаблону MVC

Отже мною було прийняте рішення розробляти проект на мови програмування PHP а особливо на фреймворку Laravel, тому що цей фреймворк є дуже зручним у роботі, особливо написання коду, а самим головним плюсом є те що у цьому фреймворку використовується архітектура MVC.

2.2Опис структури даних та моделі бази даних

На сьогоднішній день, дуже важливою частиною у більшості додатках та веб-додатках є бази даних у яких зберігають різноманітну інформацію та дані.

Бази даних є найефективнішим способом постійного зберігання та отримання оперативних та аналітичних даних у цифровому вигляді.

Бази даних бувають двох видів:

- реляційні;
- нереляційні.

Реляційна база даних — це набір інформації, яка організовує точки даних із визначеними зв'язками для легкого доступу. У моделі реляційної бази даних структури даних - включаючи таблиці даних, індекси та представлення даних - залишаються відокремленими від структур фізичного зберігання, що дозволяє адміністраторам баз даних редагувати фізичне сховище даних, не впливаючи на логічну структуру даних.

На підприємстві реляційні бази даних використовуються для упорядкування даних і виявлення зв'язків між ключовими точками даних. Вони дозволяють легко сортувати та знаходити інформацію, що допомагає організаціям приймати бізнес-рішення ефективніше та мінімізувати витрати. Вони добре працюють зі структурованими даними.

Таблиці даних, які використовуються в реляційній базі даних, зберігають інформацію про пов'язані об'єкти. Кожен рядок містить запис з унікальним ідентифікатором, відомим як ключ, і кожен стовпець містить атрибути даних. Кожен запис призначає значення кожній ознаці, завдяки чому зв'язки між точками даних легко ідентифікувати.

Стандартним інтерфейсом користувача та прикладної програми (API) реляційної бази даних є мова структурованих запитів. Оператори коду SQL використовуються як для інтерактивних запитів на інформацію з реляційної бази даних, так і для збору даних для звітів. Для забезпечення точності та доступності реляційної бази даних необхідно дотримуватися визначених правил цілісності даних.

Бази даних мають різні реалізації. Вони також відрізняються версією SQL, яку вони підтримують. Хоча існує стандарт, який описує мову SQL, між SQL,

Програмне забезпечення, яке використовується для зберігання, отримання, керування та зміни даних, що зберігаються в реляційній базі даних, називається системою управління реляційною базою даних (RDBMS).

СУБД забезпечує інтерфейс між користувачами, програмами та базою даних. Система керування базами даних також надає вам доступ до адміністративних функцій для спрощення керування зберіганням даних, доступом та продуктивністю.

Після введення даних у базу даних можна порівнювати значення з різних стовпців, зазвичай також з різних таблиць, і об'єднувати рядки, коли їх значення збігаються.

Це дозволяє виконувати зв'язування даних і відносно складні операції в базі даних. Усі операції логічні, незалежно від положення рядка таблиці. Рядки в реляційній базі даних зберігаються в абсолютно довільному порядку він не повинен відображати порядок, в якому вони введені, або порядок, у якому вони зберігаються.

Типи відносин:

- один до одного (якщо один запис з першої таблиці містить не більше одного запису з другої таблиці, і навпаки);
- один до багатьох (якщо один запис з першої таблиці може мати один або кілька записів з другої, але один запис з другої таблиці має не більше одного запису з першої таблиці);
- багато-до-багатьох (відбувається, коли один запис з першої таблиці містить багато записів з другої таблиці, а один запис з другої таблиці містить багато записів з першої таблиці).

Нереляційна база даних, або база даних NoSQL, працює інакше. Він має працювати з напівструктурованими даними. Кожен запис міститься в одному записі JSON. Отже, база даних NoSQL виглядає як папка з файлами, а не таблиця.

В результаті він може обробляти будь-який тип даних без необхідності

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					37

змінювати архітектуру. Таким чином, створення та підтримка бази даних NoSQL є швидшим і дешевшим.

Однак є і недоліки. За цю гнучкість доведеться заплатити додатковими зусиллями щодо обробки. Крім того, адміністрування бази даних NoSQL є більш складним. Через неточну структуру даних передача записів на сильно типізовані мови програмування стає проблемою.

Коли корисна нереляційна база даних? Соціальні медіа, аналітичне програмне забезпечення – чим більше різних типів даних має обробляти програма, тим очевиднішим стає використання бази даних NoSQL.

Приклади нереляційних баз даних:

– MongoDB - справжнім лідером на ринку є база даних MongoDB з відкритим вихідним кодом. Записи зберігаються як документ BSON (модифікація JSON). MongoDB керує веб-сайтами Craigslist, eBay, Foursquare і The New York Times. Подібна до документа структура ідеально підходить для обробки тисяч текстів і коментарів, які ці веб-сайти зберігають щодня;

– DynamoDB - це популярний продукт Amazon Web Services, який використовується для зберігання та обробки даних. DynamoDB цінується за вбудовану безпеку (шифрування документів) і кешування в пам'яті для швидшої взаємодії. Samsung, Netflix і Lyft є одними з відомих користувачів DynamoDB;

– Elasticsearch - це повнотекстовий пошуковий механізм, який також часто використовується як нереляційна база даних. Перевагою Elastic є простота його архітектури даних. Крім того, реакція набагато швидша в порівнянні з базами даних SQL;

Існують також бази даних ключ-значення, також відомі як словники або хеш-таблиці. Записи в них витягуються за допомогою унікального ключа, який допомагає швидко знайти необхідні дані в базі даних. Бази даних "ключ-значення" більш гнучкі та легші, ніж класичні реляційні, що дозволяє їм позитивно впливати на продуктивність веб-додатків.

Вибір між реляційною та нереляційною базою даних може здатися

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					38

складним, але насправді все, що вам потрібно зробити, це визначити свої потреби та проаналізувати переваги та недоліки обох типів.

Якщо ви працюєте з добре структурованими, більш-менш постійними даними – кращим вибором буде база даних SQL. Це впорядковує вашу інформацію та заощадить місце для зберігання. Однак, якщо вам доводиться обробляти різні типи даних, будьте готові до складних оновлень та обслуговування бази даних.

В іншому випадку ви можете вирішити використовувати базу даних NoSQL, яка зберігає всі доступні дані у вигляді файлів. Так, вона менш структурована, але для створення та підтримки такої бази потрібно набагато менше часу.

Отже мною була вибрана реляційна база даних, тому що вона найбільш підходить до вимог мого проекту.

Для реалізації збереження всіх введених даних, отримання даних за ключем, їх оновлення та видалення було використано phpMyAdmin. Були створені таблиці для коректної роботи програми відповідно до необхідних вхідних даних.

Відповідно до предметної області було створено такі головні таблиці як:

- Users;
- Prices;
- Services;
- Barbers.

Users - таблиця потрібна для того, щоб зберігати всю інформацію про користувачів. Ця таблиця містить такі поля як:

- id – унікальний ідентифікатор;
- first_name – Імя користувача;
- last_name – Прізвище користувач;
- username – логін користувача;
- email - Емаїл;

– telegram_id – ідентифікатор, який буде з'єднувати телеграм користувача з юзером у веб-сайті;

– role_id – ідентифікатор ролі користувача.

Orders - таблиця потрібна для того, щоб зберігати всю інформацію про різноманітні записи на візит до перукаря. Ця таблиця містить такі поля як:

– id – унікальний ідентифікатор;

– first_name – Імя користувача;

– last_name – Прізвище користувач;

– email - Емайл;

– phone – номер телефону який вводить користувач при запису на візит;

– date – дата;

– status – статус замовлення (виконане/не виконане);

– price_id – зв'язок з таблицею ціни;

– barber_id - зв'язок з таблицею барбери;

– user_id - зв'язок з таблицею користувачів;

– workingtime_id - зв'язок з таблицею робочих часів.

Prices - таблиця потрібна для того, щоб зберігати всю інформацію про всі ціни які привязані до послуг. Ця таблиця містить такі поля як:

– id – унікальний ідентифікатор;

– title – Назва послуги;

– image – картинка;

– price - ціна;

– position – позиція на якій буде показана ціна;

– visible – статус який забороняє або дозволяє показувати цей запис на сторінці.

Barbers - таблиця потрібна для того, щоб зберігати всю інформацію про всіх барберів які працюють у перукарні. Ця таблиця містить такі поля як:

– id – унікальний ідентифікатор;

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					40

- name – Назва послуги;
- last_name – картинка;
- working_position - ціна;
- image – позиція на якій буде показана ціна;
- visible – статус яки забороняє або дозволяє показувати це запис на сторінці.

Схему бази даних можна подивитись на рисунку 2.4.

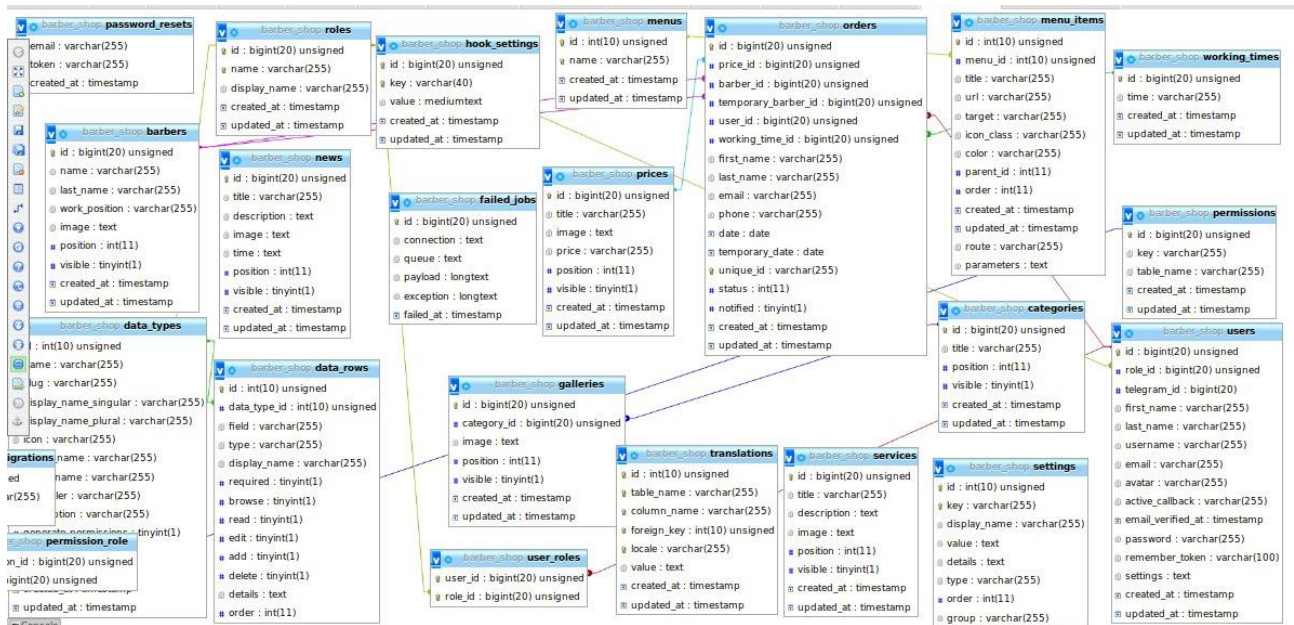


Рисунок 2.4 – Схема бази даних

2.3 Проектування серверної частини веб-сайта

Код, який створюють розробники бекенду, гарантує, що все, що буде розробник frontend, є повністю функціональним. Головне завдання розробника серверної частини – переконатися, що сервер, програма та база даних взаємодіють між собою. Для вирішення такої складної задачі розробники

використовують серверні мови, такі як PHP, Ruby, Python та Java для створення програми.

Веб-технології стрімко розвиваються, і щоб встигати за швидкістю розвитку нових технологій, у веб-програмістів часто бракує часу для реалізації

Розглянемо також не менш популярний фреймворк Laravel 7. Laravel – це фреймворк для веб-додатків з виразним і елегантним синтаксисом. Він дозволить спростити вирішення основних наболілих завдань, таких як автентифікація, маршрутизація, сесії і кешування. Laravel – це спроба об'єднати все найкраще, що є в інших PHP фреймворк, а також Ruby on Rails, ASP.NET MVC і Sinatra.

Laravel – доступний, але потужний. Має безліч відмінних інструментів для великих, надійних додатків:

Як і багато інших PHP фреймворків, для організації коду Laravel використовує архітектурний патерн MVC;

– Laravel дотримується філософії виразного, красивого синтаксису. Він призначений для людей, які цінують елегантність, простоту і читаність;

– Функціонал Laravel є простим для розуміння і використання;

– Більшість функцій Laravel чудово працюють, не вимагаючи додаткових налаштувань. Вони спираються на загальноприйняті стандарти написання коду, роблячи його інтуїтивно зрозумілим;

– Документація Laravel закінчена і постійно оновлюється;

– Чудова IoC (Інверсія управління);

– Зручна система міграцій;

– Інтегрована система модульного тестування.

Обидва фреймворка для організації коду використовують архітектурний патерн MVC. При детальному розгляді ці фреймворки складаються з трьох основних компонентів:

– Model – це сам додаток який нічого не знає про HTTP запитих;

– View – це HTTP запит / відповідь і уявлення даних, які вимагає клієнт від сервера;

– Controller – це кілька і більше класів, чиє завдання – абстрагування моделі від HTTP;

За архітектурою дані фреймворки ідентичні, так як реалізують один і той же

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					43

архітектурний патерн. Він дозволяє швидко реалізувати проект, а також легко супроводжувати і масштабувати його в подальшому.

Розширення важлива частина фреймворку, тому розглянемо їх докладніше. Розширення – це можливість підключати додаткові модулі або бібліотеки для фреймворка, які дозволять розширити його можливості. Yii і Laravel підтримують таку можливість. На даний момент, розширень набагато більше, звичайно ж, у Yii, так як фреймворк живе набагато довше, ніж Laravel, однак другий, в свою чергу, розвивається дуже великими темпами і включає в себе багато всього для роботи відразу «з коробки».

Розширення є абсолютно для будь-якого завдання, будь то панель адміністратора або ж платіжна система.

Чимало важливим фактором при виборі фреймворка є його документація. Через малу кількість років розробки або особистих переконань авторів документація Laravel дуже мізерна і неінформативна. Тому для навчання доведеться дивитися вихідний код, і як наслідок, не завжди буде зрозуміло, що робить та чи інша функція. В Yii ж є велика і корисна документація, де описана кожна функція із зазначенням її призначення і приклад використання. Так що, безсумнівно, це величезний плюс фреймворка Yii.

Для належної роботи сайту важлива підтримка REST API. REST – це архітектурний стиль взаємодії між компонентами розподілених додатків у мережі. REST – це набір послідовних обмежень, які слід враховувати при проектуванні розподіленої системи гіпермедіа. У деяких випадках (інтернетмагазини, пошукові системи, інші системи на основі даних) це може підвищити продуктивність та спростити архітектуру. Загалом, компоненти REST взаємодіють із клієнтами та серверами у всесвітній мережі.

В Інтернеті віддалені процедурні виклики можуть бути звичайними HTTP-запитами (зазвичай "GET" або "POST"; це називається "REST-запит"), а необхідні дані передаються як параметри запиту.

Для веб-служб, які відповідають REST тобто, не порушуючи жодних

										Арк.
										44
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ						

обмежень, що застосовуються до нього, використовується термін "RESTful". На відміну від веб-служб на основі SOAP, веб-API RESTful не має "офіційного" стандарту. Справа в тому, що REST – це архітектурний стиль, а SOAP – протокол. Хоча сам REST не є стандартом, більшість реалізацій RESTful використовують стандарти HTTP, URL, JSON та XML.

В кінцевому рахунку, в даний час всі сучасні PHP фреймворки зобов'язані підтримувати REST архітектуру. Нижче наведено список можливостей фреймворків для роботи з REST API.

Yii 2:

- Підтримка JSON, JSONP і XML;
- Роутинг відповідно до REST-запитами;
- Підтримка HATEAOS;
- Кешування запитів;
- Обмеження швидкості.

Laravel:

- Налаштування роутінга REST-запитів;
- Підтримка JSON, JSONP.

На основі проведеного аналізу був зроблений вибір на сторону, що активно розвивається Laravel у цій сфері, його структура дозволяє легко масштабувати веб-додаток.

2.4 Проектування клієнтської частини веб-сайта

Для створення дизайну даного веб сайту спочатку було придумано весь інтерфейс користувача згідно вимог. Таким чином було розроблено макети інтерфейсу користувача.

Завдання розробника frontend – кодувати інтерфейс веб-сайту чи програми; тобто частину веб-сайту, яку бачить і взаємодіє користувач. Вони працюють над розробками, наданими веб-дизайнером, і реалізують їх за

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					45

допомогою HTML, JavaScript та CSS. Мови розмітки використовуються для визначення форматування текстового файлу. Іншими словами, мова розмітки повідомляє програмне забезпечення, яке відображає текст, як слід форматувати текст. Мови розмітки є повністю розбірливими для читання – вони містять стандартні слова, але теги розмітки приховані від користувача в кінцевому варіанті роботи.

Дві найпопулярніші мови розмітки – HTML та XML. HTML розшифровується як мова розмітки HyperText та використовується для створення веб-сайтів. Додані до звичайного текстового документа, всі теги HTML описують, як цей документ повинен відобразитися веб-браузером.

Невідмінна частина будь-якої веб сторінки – каскадні стилі розмітки сторінки або CSS. Стилi – це в основному сукупність стилістичних правил. Мови аркушів стилів використовуються, доволі буквально, для оформлення документів, написаних мовами розмітки.

CSS розшифровується як каскадні таблиці стилів з акцентом на «стиль». Хоча HTML використовується для структури веб-документа, визначає такі речі, як заголовки та абзаци, і дозволяє вставляти зображення, відео та інші медіа, CSS визначає стиль створеного документа.

Насамперед розглянемо головну сторінку. На якій розташований хедер, який в собі містить такі пункти як:

- Головна;
- Про нас;
- Послуги;
- Ціни;
- Барбери;
- Галерея;
- Контакти.

На головній сторінці окрім пункту меню також є коротка розповідь про саму перукарню рисунок 2.5, інформація про декілька послуг та цін, інформація

про барберів, та контактна інформація. При натиску на кнопку Контакти, сторінку буде плавно проскролюватись в низ, і в низу буде блок з відображенням контактної інформації рисунок. 2.6. Цю контактну інформацію можна буде переглядати на різних сторінках.

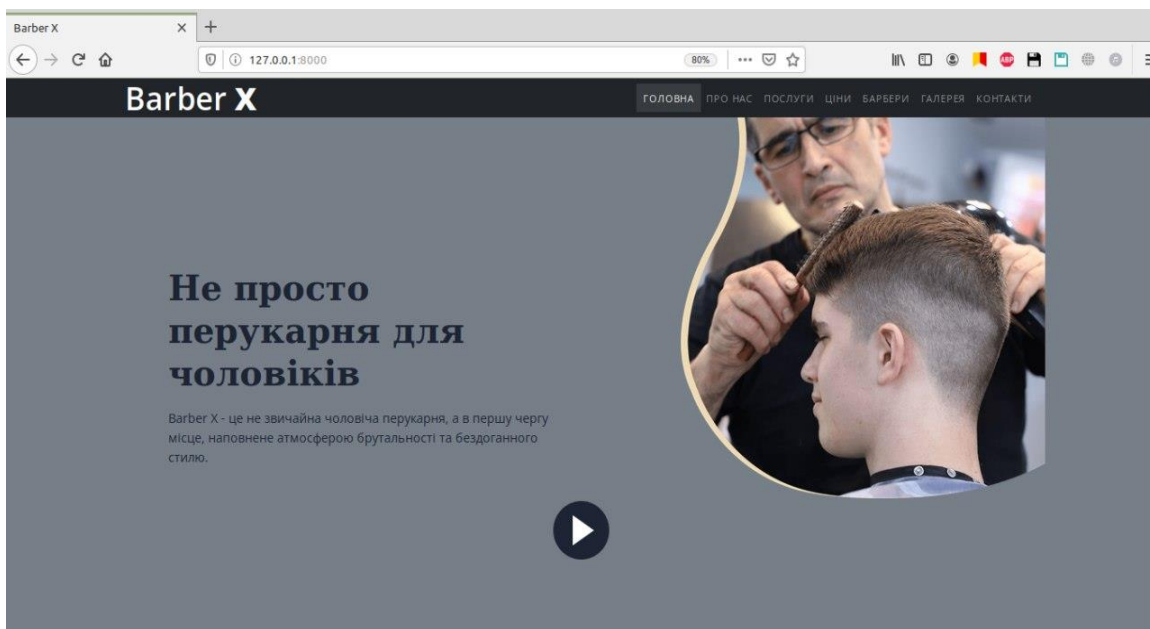


Рисунок 2.5 — Головна сторінка.

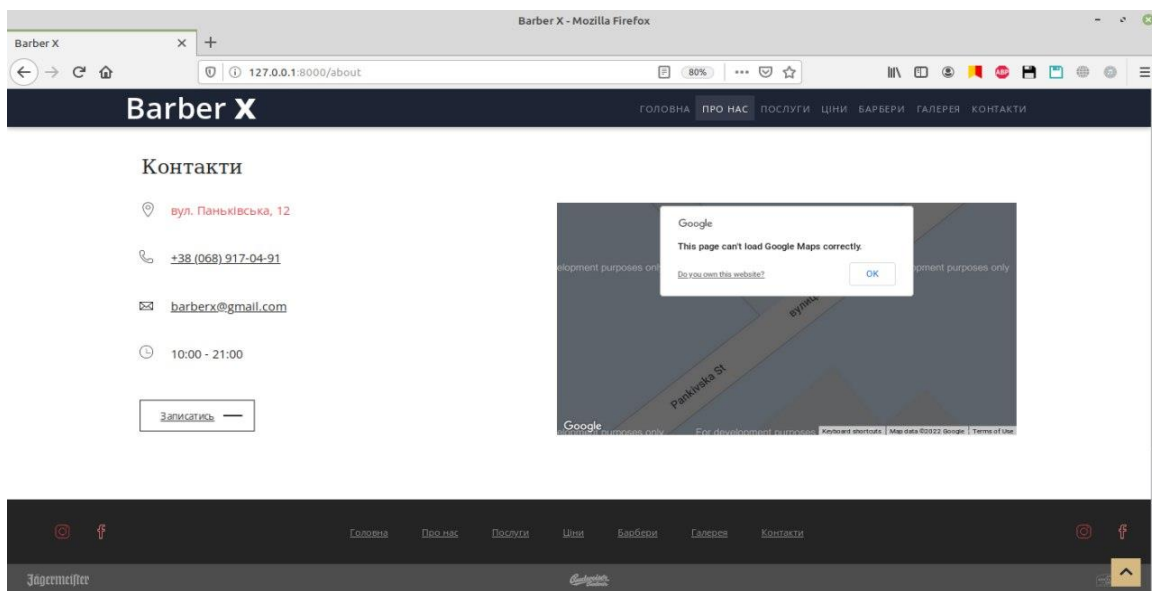


Рисунок 2.6 — Контактна інформація.

Перейшовши на сторінку Про нас, ми можемо побачити всю детальну інформацію про нашу перукарню, її особливості, а також, короткий опис, як

										Арк.
										47
Зм.Арк	№ докум.	Підпис	Дата							

було створено цю перукарню, хто її започаткував і т.д. рисунок 2.7.



Рисунок 2.7 — Сторінка про нас.

Перейшовши на сторінку послуги, ми можемо роздивитись різноманітні послуги, які існують у цій перукарні, детальну інформацію про певну послугу, що в неї входить, що мусить зробити барбер, і який кінцевий результат чекає на користувача рисунок 2.7.

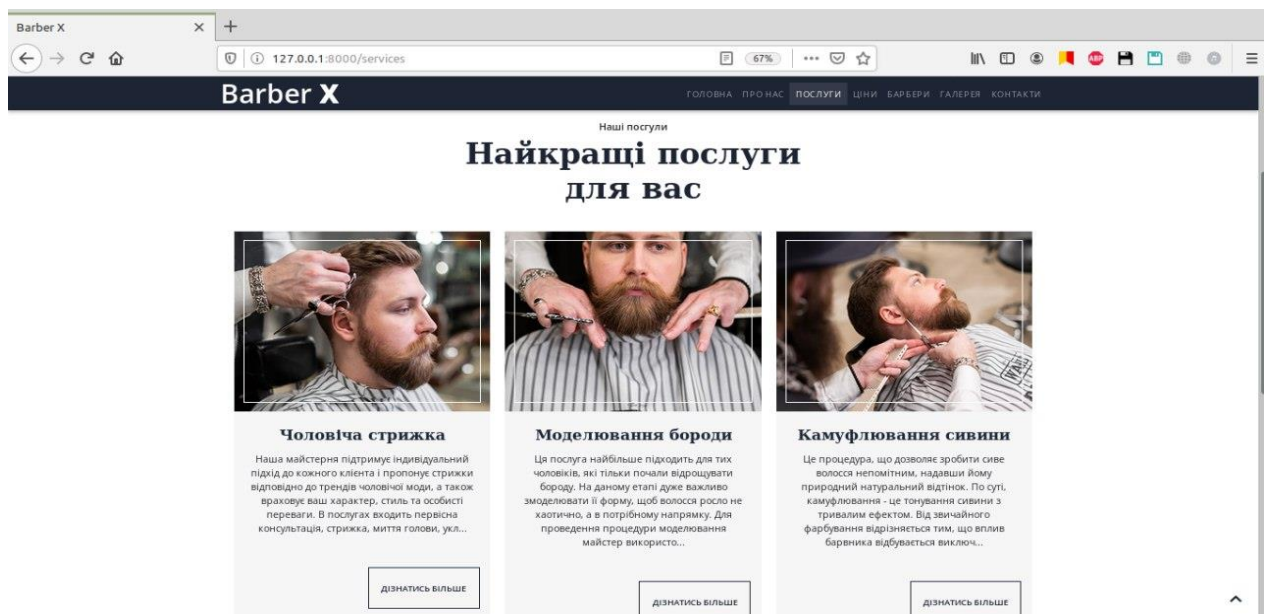
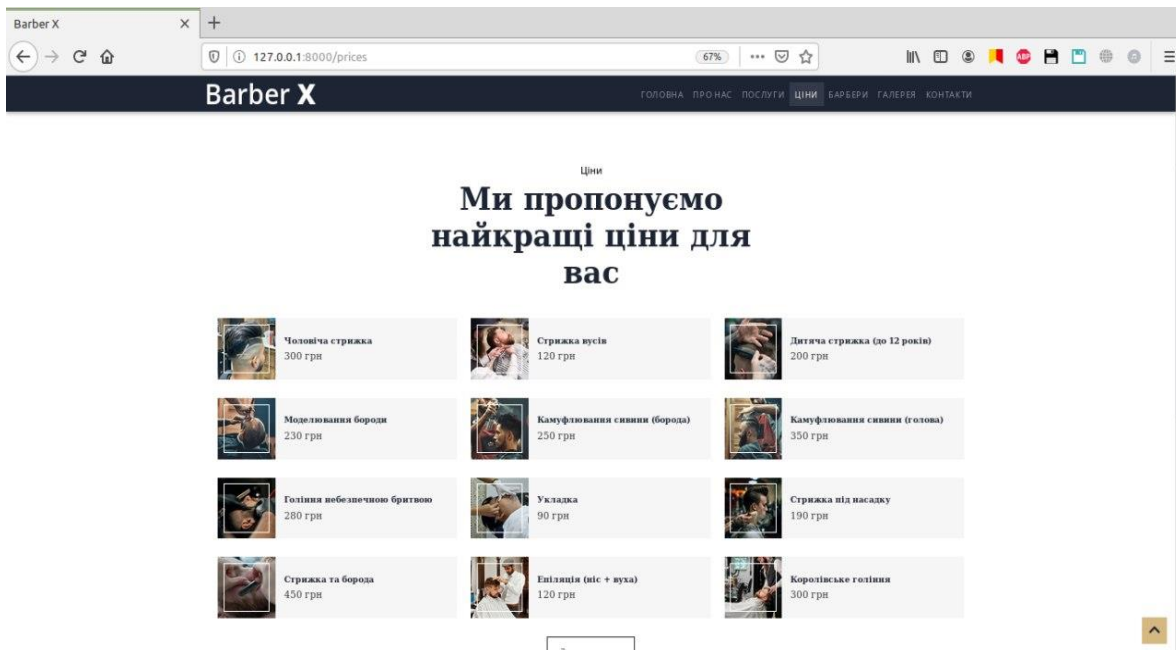


Рисунок 2.7 — Сторінка Послуги.

Перейшовши на сторінку ціни, користувач дізнається весь асортимент, який надає ця перукарня та ціни на нього рисунок 2.9.

									Арк.
									48
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					



Рисуно

к 2.9— Сторінка ціни.

Перейшовши на сторінку барбери, користувач може переглянути всіх працівників закладу, почитати про них детальну інформацію, і вирішити у майбутньому до якого барбера він хоче записатись на візит рисунок 2.10.

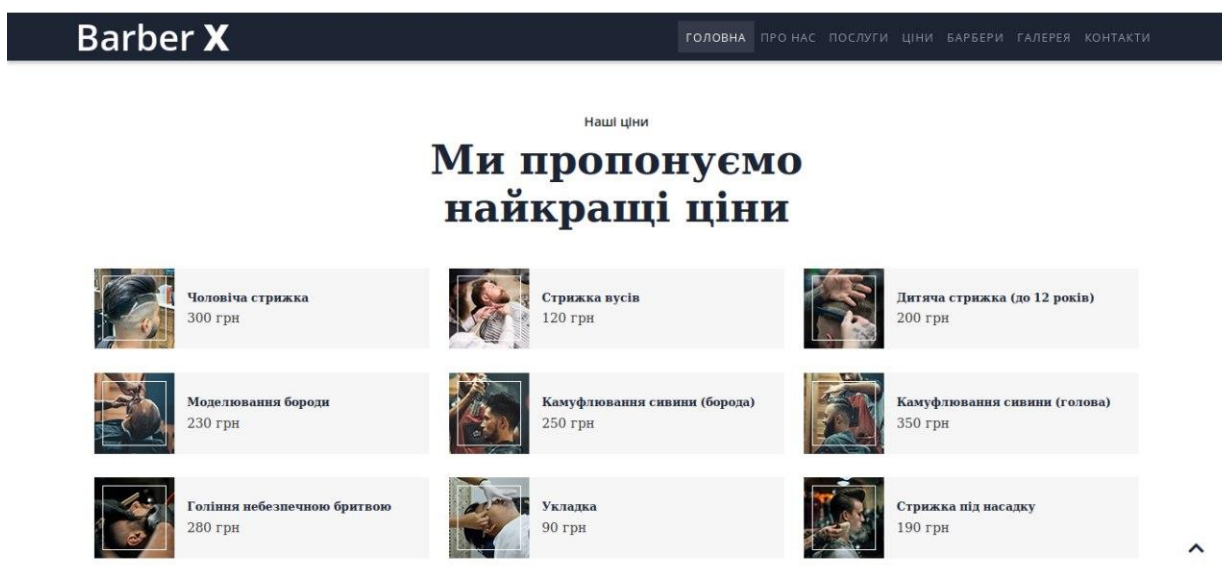


Рисунок 2.10— Сторінка ціни.

В кінцевому результаті, є сторінка з оформленням замовлення, відображені різноманітні поля, де користувач ввівши всі дані записується на візит у перукарню, на певну дату, на певний час, до певного барбера вибравши

Зм.Арк	№ докум.	Підпис	Дата

ДППЗ.190153.19.06.ПЗ

Арк.

49

певну послугу - рисунок 2.11.

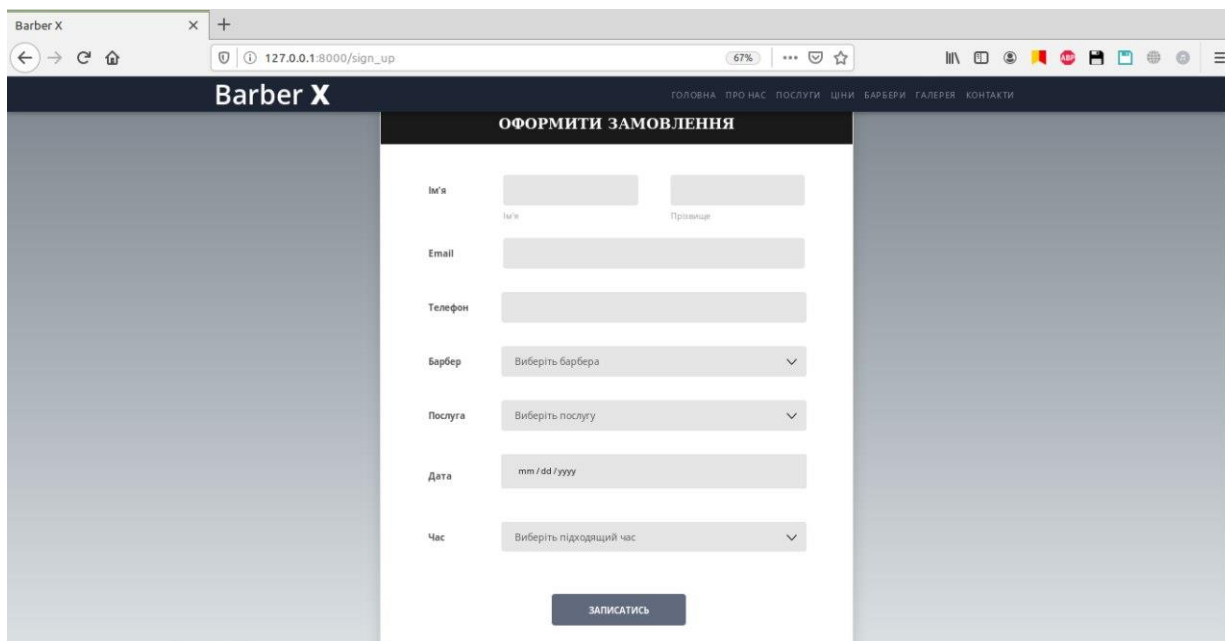


Рисунок 2.11.— Сторінка оформлення замовлення.

2.5 Аналіз та вибір технологій і методів реалізації веб-сайту

Розробка веб-сайтів – це процес створення веб-сайтів та додатків для інтернету. Від найпростіших, статичних веб-сторінок до платформ і додатків соціальних медіа, від веб-сайтів електронної комерції до систем управління контентом (CMS) і файлообмінників; всі інструменти, якими ми користуємось через інтернет щодня, були розроблені веб-розробниками. Веб-розробка може бути розбита на три шари (табл. 2.1): кодування на стороні клієнта (фронтенд), кодування на стороні сервера (бекенд) та технології баз даних.

Таблиця 2.1 – Частина веб розробки

Назва	Опис	Приклад технологій розробки
Клієнтська сторона	Сценарії на стороні клієнта або розробка інтерфейсу	HTML, CSS, Vue.js, Vuetify,

Продовження таблиці 2.1.

Кінець таблиці 2.1.

	щоб полегшити його роботу. База даних працює на сервері, і більшість веб-сайтів зазвичай використовують певну форму управління реляційною базою даних	
--	---	--

Отже, для створення сайту за темою дипломного проекту, було вибрано мову програмування PHP, а саме фреймворк Laravel, який на сьогоднішній день, є найбільш відомим, та найбільш розвиненим фреймворком у цьому напрямку. Завдяки йому злегкістю можна створювати веб-сайти різного рівня складності. Для створення фронтенд частини мною було вибрано чистий HTML і CSS. А для створення бази даних я обрав MySQL

127.0.0.1.

DB_PORT — номер мережного порту, через який відбудеться з'єднання із сервером бази даних. Якщо ви не вказували його явно при налаштуванні СУБД на сервері, то для MySQL за замовчуванням він буде 3306, як і в прикладі вище, для PostgreSQL - 5432, для інших підтримуваних LaravelDB потрібно дивитися, напам'ять не пам'ятаю.

DB_DATABASE — власне ім'я бази даних, яка буде використовуватися для зберігання інформації Laravel програми.

DB_USERNAME — ім'я користувача СУБД для доступу до даних, що зберігаються в базі даних. Користувачів, природно, у вас може бути кілька, причому кожен з них може мати свій набір прав. Події з маніпуляції користувачів з консолі MySQL описані у статті за посиланням на початку цієї публікації.

DB_PASSWORD — пароль користувача СУБД для підключення до database з програми Laravel.

3.2 Розробка програмних модулів

Структура проекту Laravel – це, в основному, структура папок, підпапок і файлів, включених у проект. Як тільки ми створимо проект у Laravel, ми отримаємо огляд структури програми, як показано на рисунку 3.1. Проект показаний тут, відноситься до кореневої папки Laravel, а саме до laravel-project. Він включає різні підпапки і файли. Аналіз папок та файлів, а також їх функціональні аспекти наведено нижче. Модульність - надає 20 вбудованих бібліотек і модулів, які допомагають покращити програму. Кожен модуль інтегровано з менеджером залежностей Composer, що полегшує оновлення. Він включає простори імен та інтерфейси, таким чином допомагає організувати ресурси та керувати ними. При розробці веб-додатка економиться значний час, оскільки Laravel повторно використовує компоненти з іншого фреймворку при

розробці веб-додатка.

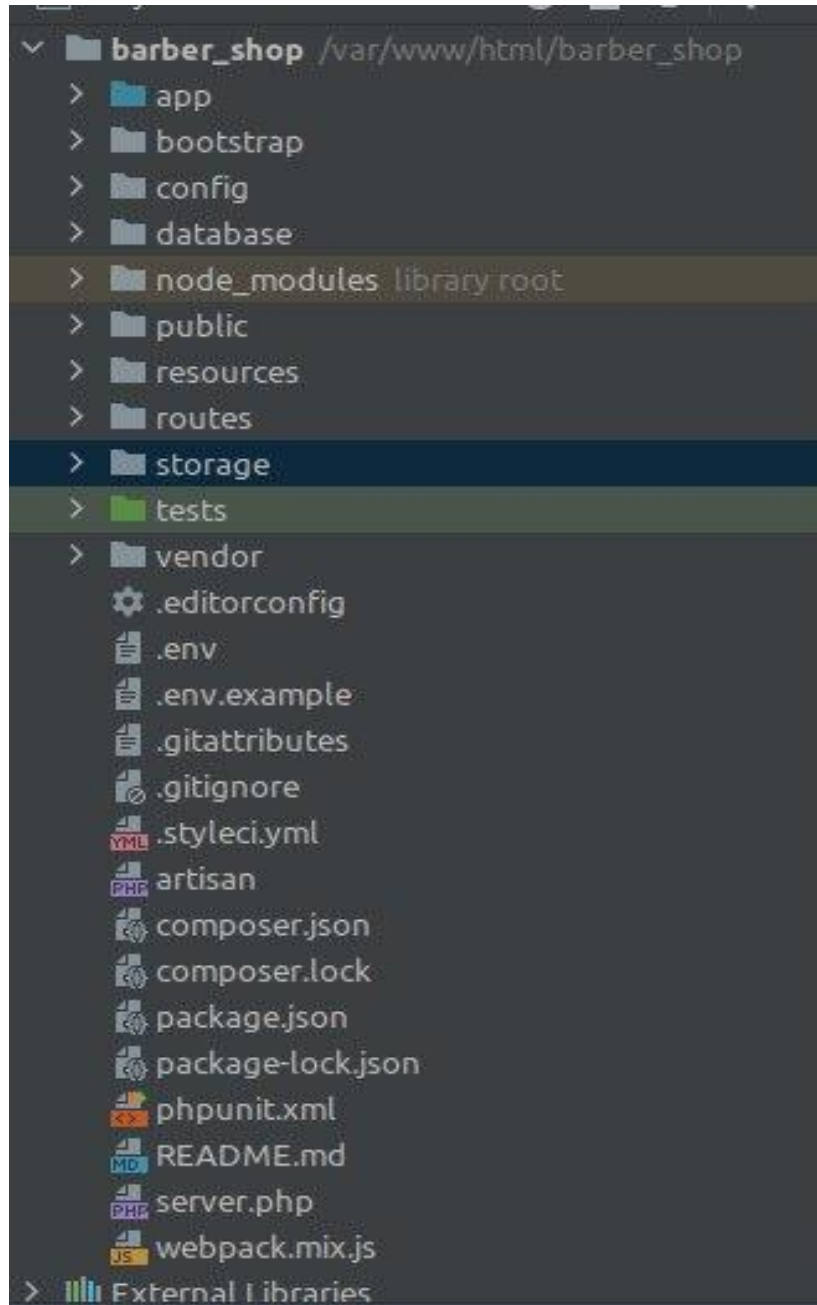


Рисунок 3.1 - Структура проекту

У папці Http рисунок 3.2.є підпапки для контролерів, міделварів, реквестів, проміжного програмного забезпечення та запитів програм. Оскільки Laravel слідує шаблону проектування MVC, ця папка включає модель, контролери та уявлення, визначені для певних каталогів. Підпапка проміжного програмного забезпечення включає механізм проміжного програмного

забезпечення, що включає механізм фільтрації і зв'язок між відповіддю і запитом.

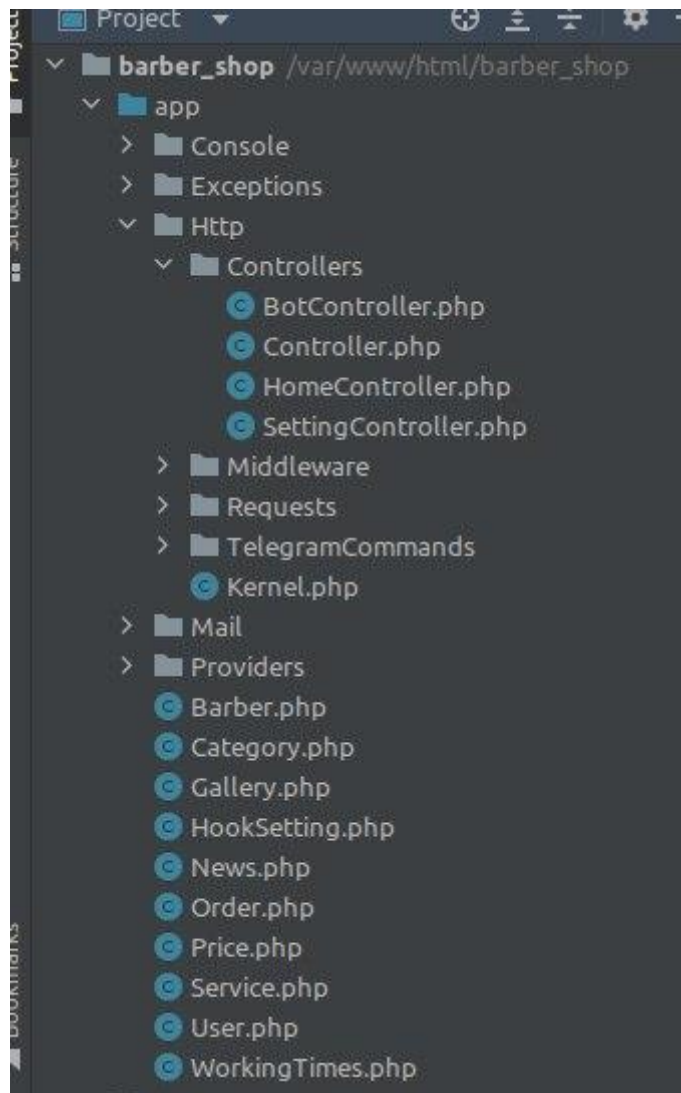


Рисунок 3.2- Структура папки Http

В папці контролер зберігаються різноманітні контролери такі як: BotController, HomeController, SettingController. Всі ці контролери відповідають за логіку проекту. При відкриванні певної сторінки спрацьовує в HomeController певна функція, яка обробляє певні дані та передає їх на інтерфейс.

Приклад HomeController:

```
class HomeController extends Controller{  
    public function index(){
```



```

        $data['galleries'] = Gallery::orderBy('position', 'ASC')->where('visible',1)-
>get();
    return view('galery', $data);
    }
    public function contacts({
        return view('contacts');
    }
    public function single(){
        return view('single');}
    public function sign_up(){
        $data['barbers'] = Barber::orderBy('position', 'ASC')->where('visible',1)-
>get();
        $data['prices'] = Price::orderBy('position', 'ASC')->where('visible',1)-
>get();
        $data['working_times'] = WorkingTimes::all();
    return view('signup',$data);
    }
    public function finish_sign_up(){
        return view('finish_signup');
    }
    public function ajax(Request $request){
        $date = $request->get('date');
        $barber_id = $request->get('barber_id');
        $working_time = Order::where('barber_id',$barber_id)-
>whereDate('date',$date)->select('working_time_id')->get()-
>pluck('working_time_id');
        $data['working_times'] =
WorkingTimes::whereNotIn('id',$working_time)->get()->pluck('time','id');
        return response()->json($data);

```

Зм.Арк	№ докум.	Підпис	Дата		

```

    }
    public function store(OrderRequest $request){
        $data = $request->validated();
        $data['status'] = 0;
        $order = Order::create($data);
        $unique_id = '#id' . $order->id .rand(0,10000);
        $order->update(['unique_id'=>$unique_id]);
        Mail::to($order->email)
            ->send(new DemoEmail($order));
        //відправляємо лист
        return view('finish_signup');
    }
}

```

Також в папці Http знаходяться всі моделі до різних таблиць у базі даних. Кожна модель містить ріноманітні поля закриті та відкриті, ріноманітні зв'язки з іншими таблицями.

Приклад однієї моделі:

```

class Order extends Model
{
    protected $fillable=['first_name', 'last_name', 'email', 'phone',
'date','temporary_date','notified','status','unique_id', 'price_id', 'barber_id',
'working_time_id','user_id', 'temporary_barber_id'];
    public function prices(){
        return $this->belongsTo(Price::class,'price_id');
    }
    public function barber(){
        return $this->belongsTo(Barber::class);
    }
    public function temporary_barber(){

```

```

return $this->belongsTo(Barber::class, 'temporary_barber_id');
    }
public function time(){
return $this->belongsTo(WorkingTimes::class, 'working_time_id');
    }
public function user({
return $this->belongsTo(User::class);
    }}

```

Папка config рисунок 3.3 включає різні конфігурації та пов'язані параметри, необхідні для безперебійної роботи програми Laravel. Різні файли, включені в папку конфігурації, як показано тут. Імена файлів працюють відповідно до функціональності, пов'язаної з ними.

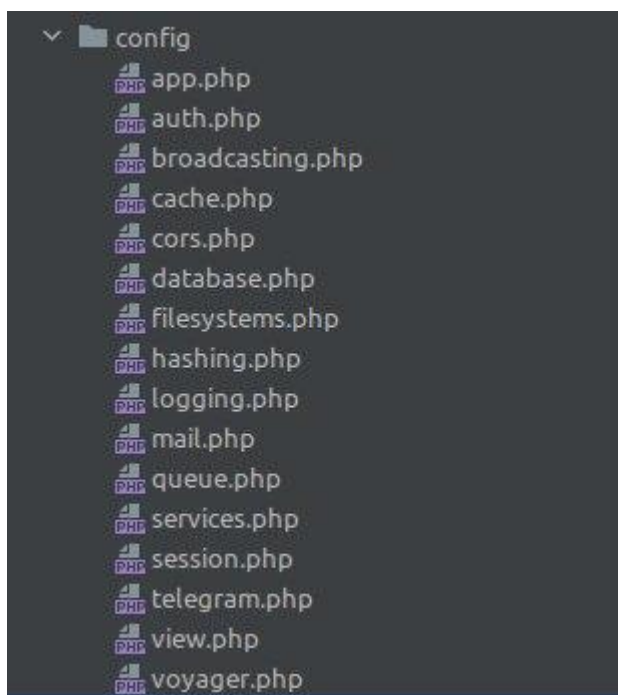


Рисунок 3.3- Структура папки config

Папка public - це коренева папка, яка допомагає в ініціалізації програми Laravel. Вона включає наступні файли та папки - .htaccess - цей файл визначає конфігурацію сервера. javascript та css - ці файли вважаються активами. index.php -

цей файл необхідний для ініціалізації веб-програми. .htaccess –

цей файл визначає конфігурацію сервера. javascript та css –

ці файли вважаються активами. index.php –

цей файл необхідний для ініціалізації веб-програми.

Каталог ресурсів рисунок 3.4 містить файли, які розширюють можливості вашого веб-програми. Підпапки, включені до цього каталогу, та їхнє призначення пояснюється нижче – assets – папка assets містить файли, такі як LESS та SCSS, які необхідні для стилізації веб-програми.

lang – ця папка містить конфігурацію для локалізації чи інтерналізації.

Уявлення – Уявлення – це HTML-файли або шаблони, які взаємодіють із кінцевими користувачами та відіграють основну роль в архітектурі MVC.

assets – папка assets містить файли, такі як LESS та SCSS, які необхідні для стилізації веб-програми.

lang – ця папка містить конфігурацію для локалізації чи інтерналізації.

Уявлення – Уявлення – це HTML-файли або шаблони, які взаємодіють із кінцевими користувачами та відіграють основну роль в архітектурі MVC.

									Арк.
									61
Зм. Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					

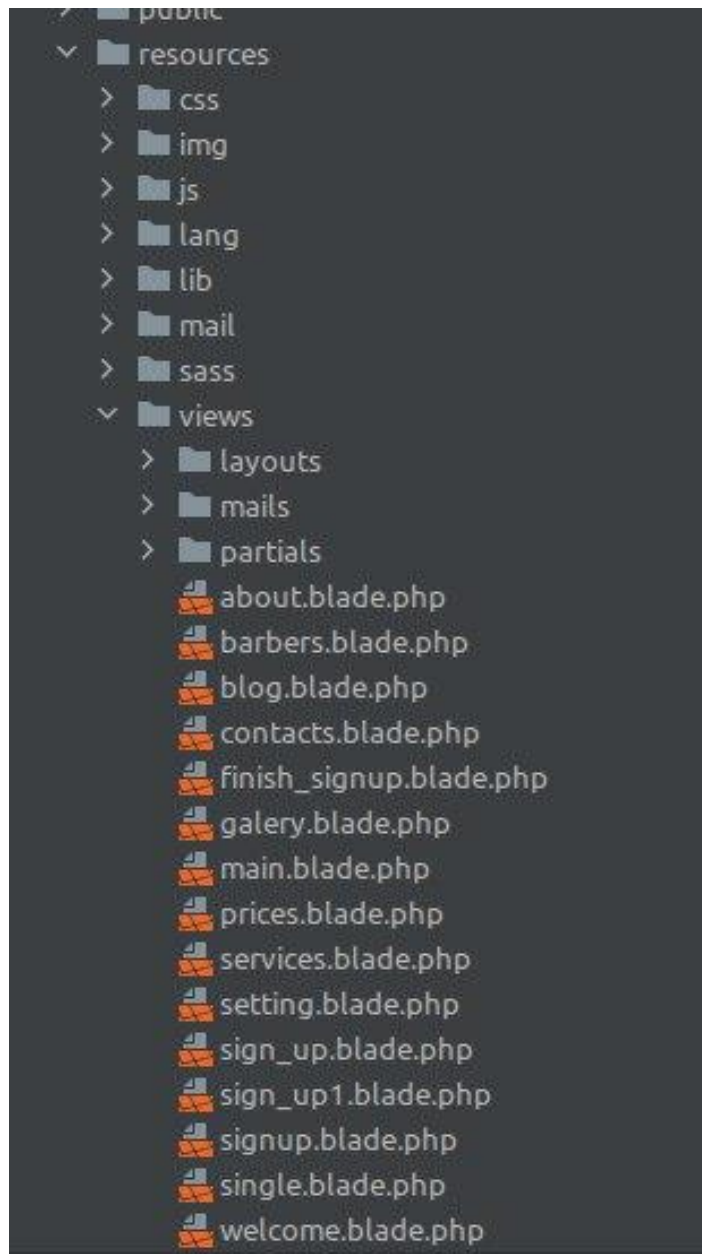


Рисунок 3.4- Структура папки resources

3.3 Керівництво користувача

Коли користувач заходить на розроблений сайт, першою сторінкою, яка йому відкриється, буде головна сторінка веб-сайту (рисунок 3.5). На цій сторінці буде відображений хедер з пунктами меню:

- головна;
- послуги;

Зм.Арк	№ докум.	Підпис	Дата	

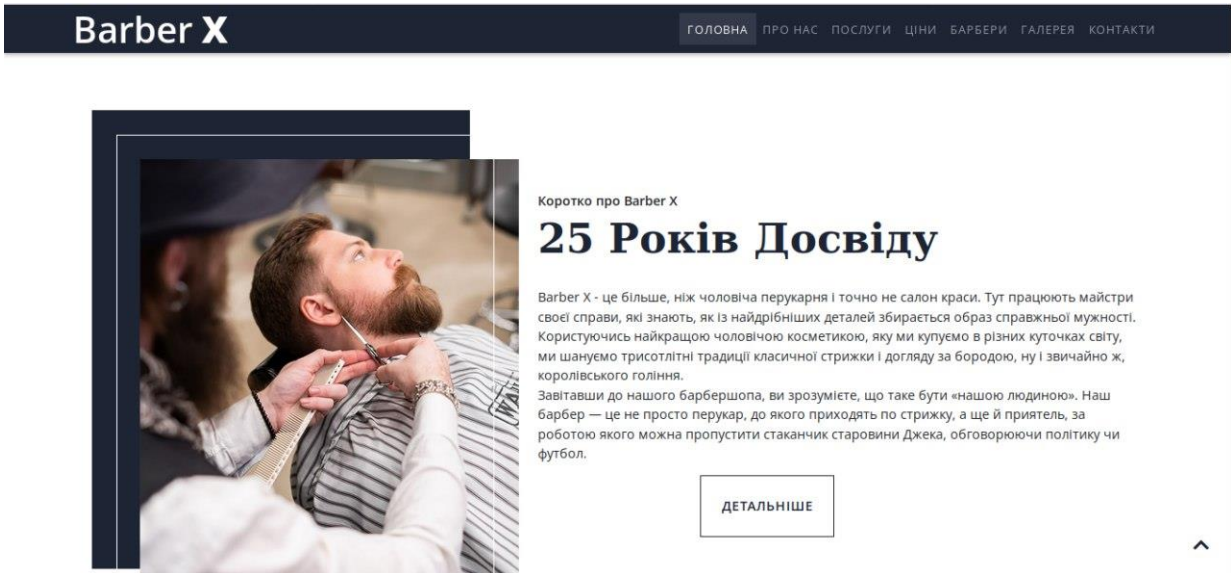


Рисунок 3.7– Детальна інформація про заклад

Прокрутивши в самий низ, буде відображено контактна інформація про заклад, а саме, номер телефону, часи роботи, вулиця розташування, email, та карта на якій показується розташування закладу, кнопка Записатись, та в низу буде відображатись меню, та силки на інстаграм та фейсбук. рисунок 3.8.

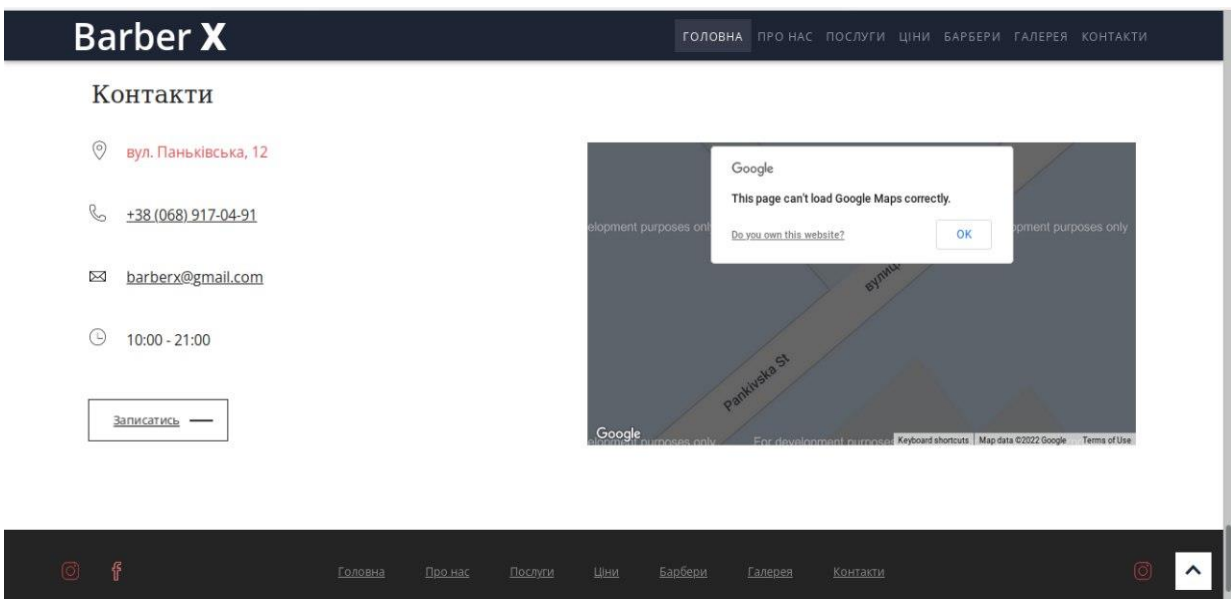


Рисунок 3.8– Контактна інформація

Нажавши на кнопку записатись, користувача буде переведено на нову сторінку запису на візит, у якій буде відображено різноманітні поля, такі як прізвище, імя, номер телефону, барбер, послуга, дата, час, які потрібно буде

Зм.Арк	№ докум.	Підпис	Дата

ВИСНОВКИ

У вступі доведено актуальність теми, визначено мету та завдання дипломного проектування.

У першому розділі проведено аналіз предметної області, розглянуто існуючі рішення та визначені функціональні і нефункціональні вимоги до розроблюваного програмного забезпечення.

У другому розділі проведено аналіз сучасних архітектур, розглянуто їх переваги і недоліки та визначено, що система буде відповідати монолітній архітектурі та моделі клієнт-сервер.

У третьому розділі підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, в результаті чого створено програмний продукт.

В четвертому розділі було виконано тестування системи та проведено його у відповідності до функціональних вимог, в результаті було підтверджено правильну роботу програми.

У ході розробки програмного продукту було реалізована веб-сайт для перукарні. У цьому сайті було створено інтуїтивний інтерфейс для користувача, таким чином щоб була можливість для гостя крок за кроком доходити до бажаного результату.

Під час розробки було проведено аналіз вимог предметної області, у якій були визначені усі вимоги для створення запису у перукарню, та з'єднання з телеграм каналом, що включають в себе розробку алгоритмів та написання коду для об'єктів.

Після аналізу різних варіантів середовищ розробки було обрано «PHPStorm», оскільки вона відповідає всім критеріям для розробки даного проекту. Для реалізації веб-сайту було обрано мову програмування PHP та фреймворк Laravel.

									Арк.
Зм.Арк	№ докум.	Підпис	Дата	ДППЗ.190153.19.06.ПЗ					73

ДОДАТОК А
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Робота виконується в рамках проекту розробки інтернет-платформи для надання перукарських послуг. Технічне завдання розроблено у відповідності до стандарту ГОСТ 19.201–78.

1 Підстава для розробки

Підставою для розробки є «Завдання на дипломний проект», затверджене завідувачем кафедри інженерії програмного забезпечення. Найменування розробки: Інтернет-платформа «Надання перукарських послуг».

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням додатку пошук та замовлення подорожей для звичайного користувача, а також облік турів та замовлень для адміністратора веб-додатку.

2.2 Експлуатаційне призначення

Програма повинна експлуатуватися на будь-яких пристроях, на яких є браузер. Кінцевим користувачем додатку може виступати будь-яка особа.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

Для звичайного користувача:

- перегляд цін;
- перегляд послуг;
- перегляд фото;
- фільтрація фото за категоріями;
- перегляд барберів;
- онлайн запис на візит;
- можливість переглянути та відредагувати замовлення у телеграм-боті;

Для адміністратора:

- авторизація в адмін панелі;
- перегляд, редагування, додавання фото;
- перегляд, редагування, додавання барберів;

- перегляд,редагування, додавання цін;
- перегляд,редагування, додавання послуг;
- перегляд,редагування, додавання робочих часів;
- перегляд,редагування, додавання інформації прозаклад;

3.2 Вимоги до надійності

Веб-додаток повинен забезпечувати такі вимоги до надійності:

- обробляти невірні дії користувача і попереджати його про можливі наслідки;
- можливість самостійно відновлюватись у разі збою;
- можливість резервного копіювання бази даних.

3.3 Умови експлуатації та вимоги до технічних засобів

Веб-сайт повинен працювати на всіх пристроях, які мають браузер та стабільний доступ до мережі «Інтернет»: смартфонах, планшетах та комп'ютерах. Браузер може бути будь-яким: Safari, GoogleChrome, Opera, MozillaFirefox, тощо.

Для роботи платформи без збоїв, потрібно, щоб пристрій, на якому вона запускається задовольняв наступні мінімальні вимоги:

- 512 Мб внутрішньої пам'яті;
- 1 Гб оперативної пам'яті;
- 2-ядерний процесор;
- доступ до мережі «Інтернет» зі швидкістю мінімум 25Мбіт/с.

3.4 Вимоги до інформаційної та програмної сумісності

При розробці веб-сайту буде використовуватись скриптова мова програмування php. Для розробки серверної частини використовуватиметься фреймворк laravel. Для розробки клієнтської частини була використано елементи інтерфейсу HTML, CSS. В якості бази даних буде використа MySQL.

3.5 Спеціальні вимоги

Програма повинна мати зручний, гарний та зрозумілий зовнішній інтерфейс користувача.

4 Вимоги до програмної документації

У момент здачі проекту замовнику надається наступний набір документів:

- текст програми;
- опис програми;
- технічне завдання;
- керівництво користувача.

5 Стадії та етапи розробки

Стадії та етапи розробки інтернет-платформи «Агентство подорожей» подані у таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки проекту

Стадія розробки	Етапи робіт	Зміст робіт
1	2	3
Технічне завдання 02.01.22 – 31.01.22	Обґрунтування необхідності розробки програми	Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання
Ескізний проект 01.02.22 – 26.02.22	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури

Кінець таблиці А.1

1	2	3
Технічний проект 29.02.22 – 19.03.22	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми
Робочий проект 20.03.22 – 15.04.22	Розробка програмного забезпечення	Реалізація програмного забезпечення; відлагодження; проведення попереднього тестування
Розробка програмної документації 16.04.22 – 22.04.22	Розробка документації до програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням
Тестування системи 23.04.22 – 30.04.22	Проведення тестування програмного забезпечення	Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення
Впровадження	Підготовка і передача програми	Підготовка і розгортання програмного забезпечення
Розробка програмної документації 16.04.22 – 22.04.22	Розробка документації до програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням

6 Порядок контролю та приймання

Контроль здійснюється кінцевими користувачами системи, підключеними на етапі тестування додатку.

ДОДАТОКВ
(обов'язковий)

КОД (ЛІСТИНГ) ПРОГРАМИ

Barbers.blade.php

```

@extends('layouts.master')

@section('content')
<!-- Page Header Start -->
<div class="page-header">
<div class="container">
<div class="row">
<div class="col-12">
<h2>Барбери</h2>
</div>
<div class="col-12">
<a href="">Головна</a>
<a href="">Барбери</a>
</div>
</div>
</div>
</div>
<!-- Page Header End -->
<!-- Team Start -->
<div class="team">
<div class="container">
<div class="section-header text-center">
<p>Наша команда</p>
<h2>Зустрічайте наших Барберів</h2>
</div>
<div class="row">

                @foreach($barbers as $barber)
<div class="col-lg-3 col-md-6">
<div class="team-item">
<div class="team-img">

</div>
<div class="team-text">
<h2>{{ $barber->name }} {{ $barber->last_name }}</h2>
<p>{{ $barber->work_position }}</p>
</div>
</div>
</div>
                @endforeach

</div>
</div>
</div>
<!-- Team End -->

```

```
@stop
```

Gallery.blade.php

```
@extends('layouts.master')
```

```
@section('content')
```

```
<!-- Page Header Start -->
<div class="page-header">
<div class="container">
<div class="row">
<div class="col-12">
<h2>Галерея</h2>
</div>
<div class="col-12">
<a href="">Головна</a>
<a href="">Галерея</a>
</div>
</div>
</div>
</div>
</div>
<!-- Page Header End -->
```

```
<!-- Portfolio Start -->
<div class="portfolio">
<div class="container">
<div class="section-header text-center">
<p>Галерея</p>
<h2>Фото нашого Barber X</h2>
</div>
<div class="row">
<div class="col-12">
<ul id="portfolio-filters">
<li data-filter="" class="filter-active">Bci</li>
    @foreach($categories as $category)
<li data-filter="{{ $category->id }}">{{ $category->title }}</li>
    @endforeach
</ul>
</div>
</div>
</div>
<div class="row portfolio-container">
```

```
    @foreach($galleries as $gallery)
```

```
    <div class="col-lg-4 col-md-6 col-sm-12 portfolio-item {{ $gallery->category_id }}">
```

```

<div class="portfolio-wrap">
<a href="{{Voyager::image($gallery->image)}}" data-lightbox="portfolio">

</a>
</div>
</div>
        @endforeach
</div>
</div>
</div>
<!-- Portfolio Start -->

```

```
@stop
```

HomeController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use App\Barber;
use App\Category;
use App\Gallery;
use App\Http\Requests\OrderRequest;
use App\Mail\DemoEmail;
use App\Order;
use App\Price;
use App\Service;
use App\WorkingTimes;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\Mail;

```

```
class HomeController extends Controller
```

```

{
    public function index()
    {
        $data['barbers'] = Barber::orderBy('position', 'ASC')->where('visible',1)->get();
        $data['services'] = Service::orderBy('position', 'ASC')->where('visible',1)->get();
        $data['prices'] = Price::orderBy('position', 'ASC')->where('visible',1)->get();
        return view('main',$data);
    }

    public function about()
    {
        $data['barbers'] = Barber::orderBy('position', 'ASC')->where('visible',1)->get();
        return view('about',$data);
    }
}

```

```
public function services()
{
    $data['services'] = Service::orderBy('position', 'ASC')->where('visible',1)->get();
    return view('services',$data);
}

public function prices()
{
    $data['prices'] = Price::orderBy('position', 'ASC')->where('visible',1)->get();
    return view('prices',$data);
}

public function barbers()
{
    $data['barbers'] = Barber::orderBy('position', 'ASC')->where('visible',1)->get();
    return view('barbers',$data);
}

public function portfolio()
{
    $data['categories'] = Category::orderBy('position', 'ASC')->where('visible',1)->get();
    $data['galleries'] = Gallery::orderBy('position', 'ASC')->where('visible',1)->get();
    return view('galery', $data);
}

public function contacts()
{
    return view('contacts');
}

public function single()
{
    return view('single');
}

public function sign_up()
{
    $data['barbers'] = Barber::orderBy('position', 'ASC')->where('visible',1)->get();
    $data['prices'] = Price::orderBy('position', 'ASC')->where('visible',1)->get();
    $data['working_times'] = WorkingTimes::all();
    return view('signup',$data);
}

public function finish_sign_up()
{
    return view('finish_signup');
}
```

```

public function ajax(Request $request)
{
    $date = $request->get('date');
    $barber_id = $request->get('barber_id');
    $working_time = Order::where('barber_id',$barber_id)->whereDate('date',$date)->select('working_time_id')-
>get()->pluck('working_time_id');

    $data['working_times'] = WorkingTimes::whereNotIn('id',$working_time)->get()->pluck('time','id');

    return response()->json($data);
}

public function store(OrderRequest $request){
    $data = $request->validated();
    $data['status'] = 0;
    $order = Order::create($data);
    $unique_id = '#id' . $order->id . rand(0,10000);
    $order->update(['unique_id'=>$unique_id]);

    Mail::to($order->email)
        ->send(new DemoEmail($order));
    //відправляємо лист
    return view('finish_signup');
}
}

```

migrationOrder.php

```
<?php
```

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateOrdersTable extends Migration
{
    /
    * Run the migrations.
    *
    * @return void
    */
    public function up()
    {
        Schema::create('orders', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->bigInteger('price_id')->unsigned()->nullable();
            $table->bigInteger('barber_id')->unsigned()->nullable();
            $table->bigInteger('temporary_barber_id')->unsigned()->nullable();

```

```

        $table->bigInteger('user_id')->unsigned()->nullable();
        $table->bigInteger('working_time_id')->unsigned()->nullable();
        $table->string('first_name');
        $table->string('last_name');
        $table->string('email');
        $table->string('phone');
        $table->date('date');
        $table->date('temporary_date')->nullable();
        $table->string('unique_id')->unique()->nullable();
        $table->integer('status');
        $table->boolean('notified')->default(false);
        $table->foreign('price_id')->references('id')->on('prices')->onDelete('cascade');
        $table->foreign('barber_id')->references('id')->on('barbers')->onDelete('cascade');
        $table->foreign('temporary_barber_id')->references('id')->on('barbers')->onDelete('cascade');
        $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
        $table->foreign('working_time_id')->references('id')->on('working_times')->onDelete('cascade');

        $table->timestamps();
    });
}

/
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('orders');
}
}

```

migrationNews.php

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateNewsTable extends Migration
{
    /
    * Run the migrations.
    *
    * @return void
    */
    public function up()

```

```

{
    Schema::create('news', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->text('description');
        $table->text('image');
        $table->text('time');
        $table->integer('position')->default(1);
        $table->boolean('visible')->default(true);
        $table->timestamps();
    });
}

/
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('news');
}
}

```

Price.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use TCG\Voyager\Traits\Translatable;

class Price extends Model
{
    use Translatable;

    protected $translatable = ['title'];
    protected $fillable=['title', 'price', 'image', 'position', 'visible'];
}

```

order.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Ramsey\Uuid\Type\Time;

class Order extends Model

```

```
{
    protected          $fillable=['first_name',          'last_name',          'email',          'phone',
'date','temporary_date','notified','status','unique_id',          'price_id',          'barber_id',          'working_time_id','user_id',
'temporary_barber_id'];

    public function prices()
    {
        return $this->belongsTo(Price::class,'price_id');
    }

    public function barber()
    {
        return $this->belongsTo(Barber::class);
    }

    public function temporary_barber()
    {
        return $this->belongsTo(Barber::class, 'temporary_barber_id');
    }

    public function time()
    {
        return $this->belongsTo(WorkingTimes::class, 'working_time_id');
    }

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}
```

ДОДАТОК Г
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

ІНТЕРНЕТ ПЛАТФОРМА “НАДАННЯ ПЕРУКАРСЬКИХ ПОСЛУГ”

Виконав:

студент групи ІПЗс-19-1

Лисюк Олег

Керівник:

Форкун Юрій Вікторович

2022р

Мета дослідження

- ▣ Метою даного проекту є створення інтернет платформи перукарських послуг, яка дозволить людям, відвідуючи сайт, передивитись інформацію про перукарню, дізнатись всі поточні ціни на сьогоднішній день, переглянути всі послуги, які пропонує ця перукарня, також переглянути різні фото, та контактну інформацію про неї. А саме головне буде те, що можна буде записатись на візит до перукаря, в онлайн режимі, з нагадуванням про візит у телеграм каналі.

ЦІЛЬ

Для досягнення мети проекту необхідно виконати такі задачі:

- провести дослідження предметної області;
- визначити задачі для функціоналу;
- вибрати технології, на яких буде створений сайт;
- продумати архітектуру бази даних;
- розробити архітектуру сайту;
- розробити сучасний вигляд сайту;
- розробити адмін частину;
- розробити та з'єднати з сайтом Telegram-bot;
- провести тестування адмін частини та всього сайту;

МОВИ ТА ТЕХНОЛОГІЇ



laravel



Для створення інтернет платформи перукарських послуг було використано такі технології як:

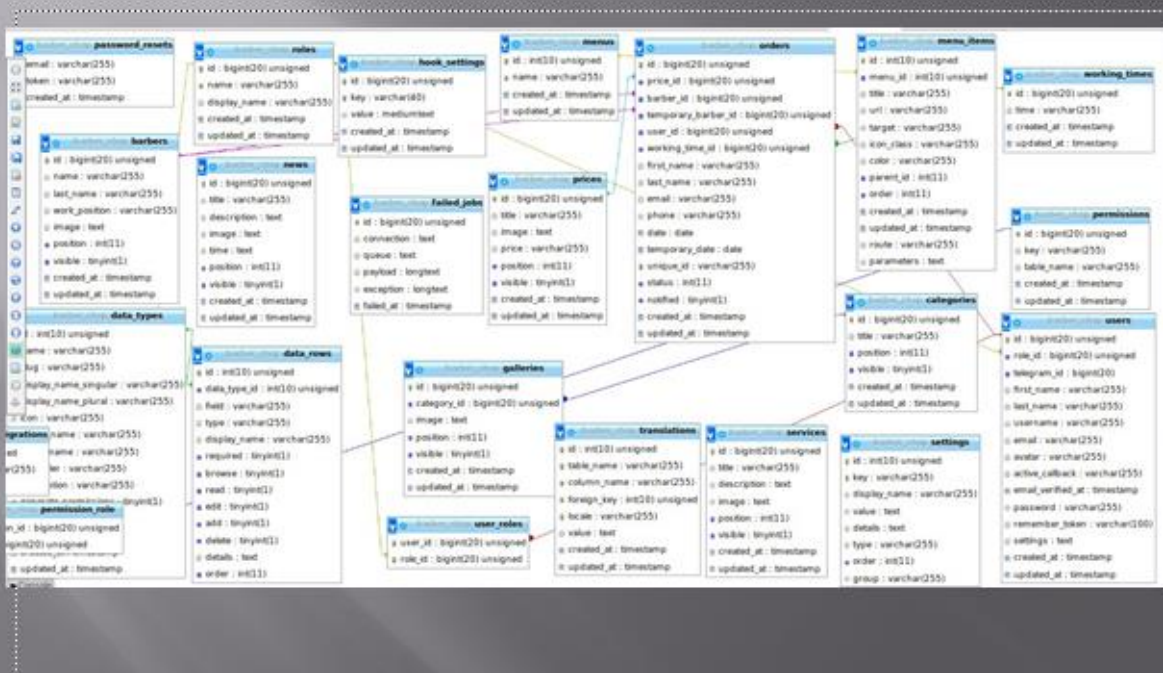
- PHP (Laravel)
- HTML
- CSS
- MySQL
- MVC
- Telegram API

BARBER X ЙОГО ПРИЗНАЧЕННЯ

Веб-сайт призначений, щоб:

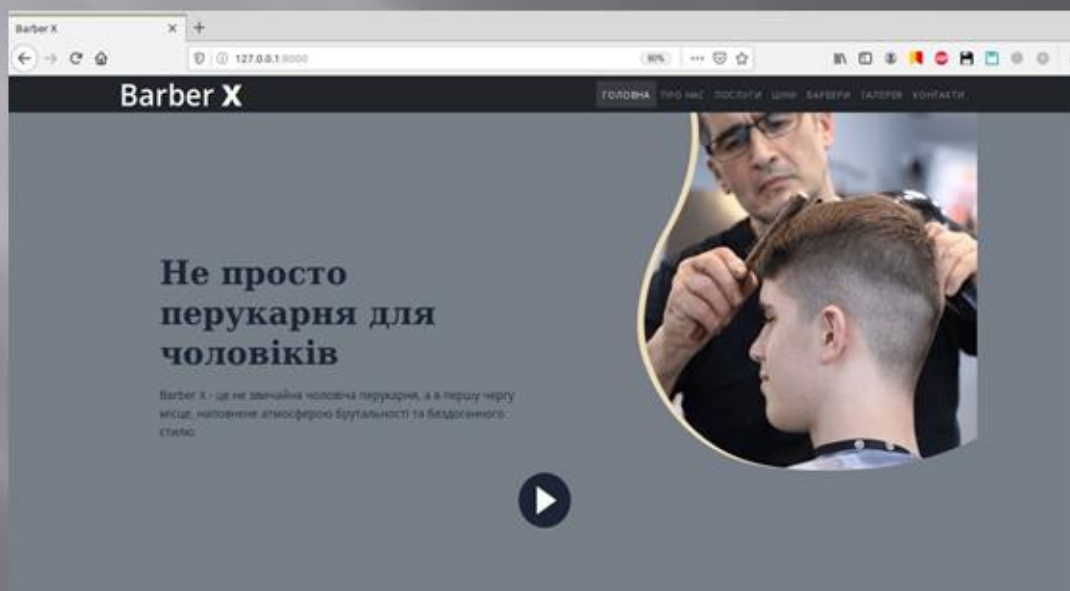
- Користувачі могли з легкістю дізнатись всю потрібну інформацію про цей заклад, без додаткових дзвінків у перукарню.
- Створити онлайн запис на візит до перукаря
- З'єднання з телеграм ботом веб-сайту

Схема бази даних



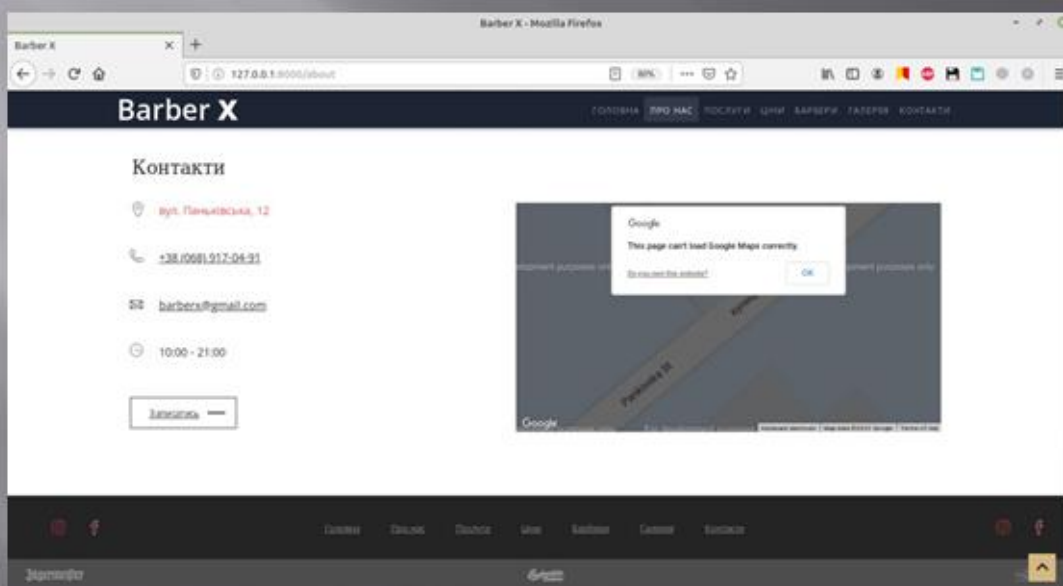
РЕЗУЛЬТАТИ РОБОТИ

Головна сторінка



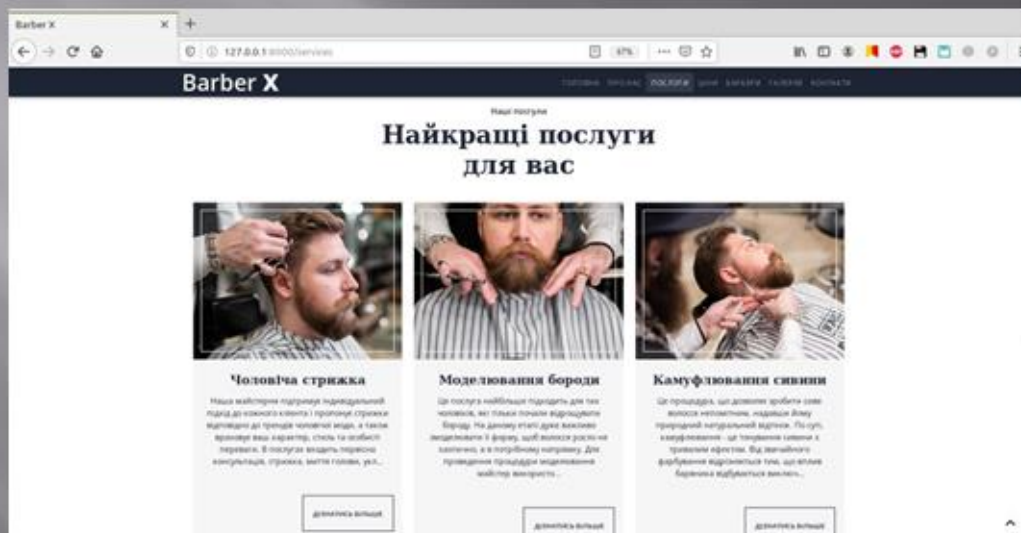
РЕЗУЛЬТАТ РОБОТИ

Контактна інформація



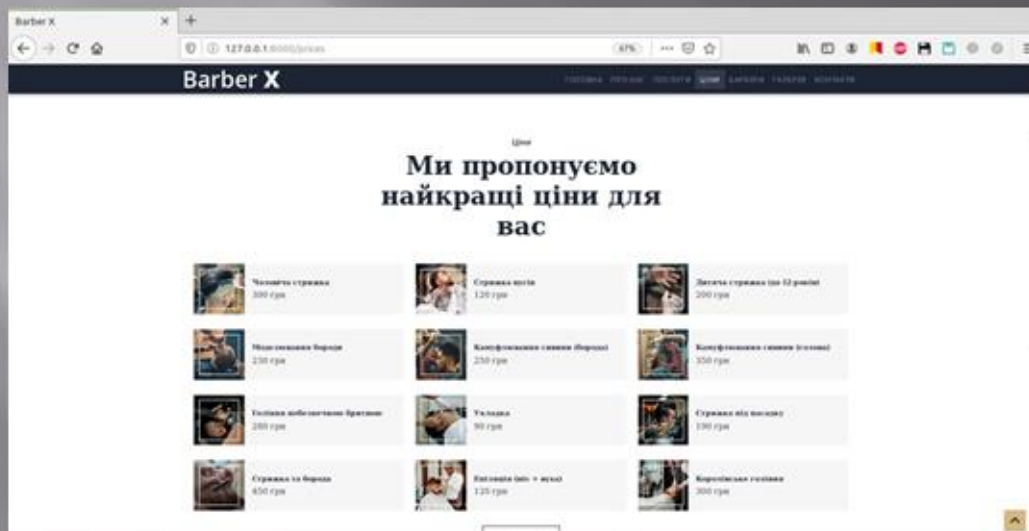
РЕЗУЛЬТАТ РОБОТИ

Сторінка послуги



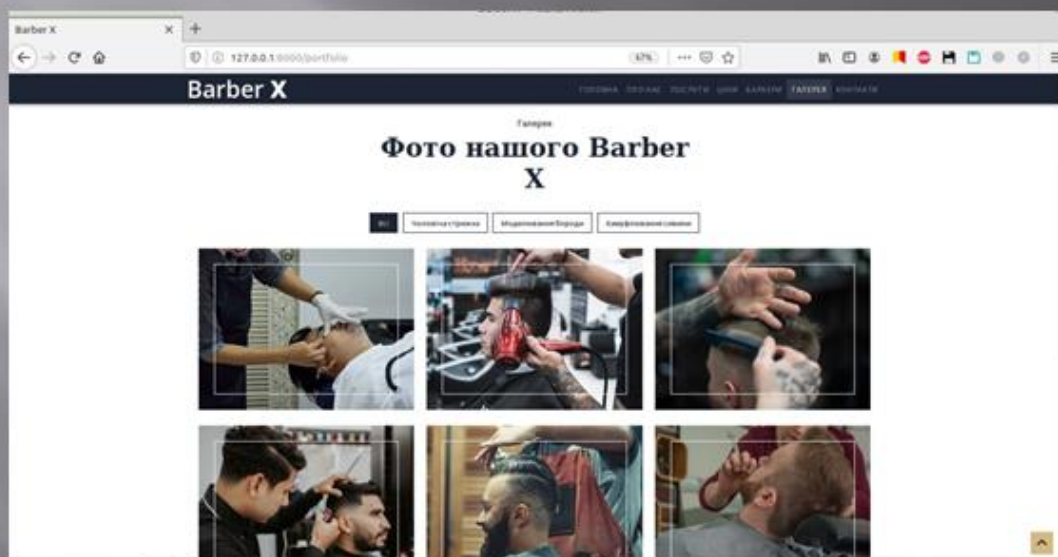
РЕЗУЛЬТАТ РОБОТИ

Сторінка ціни



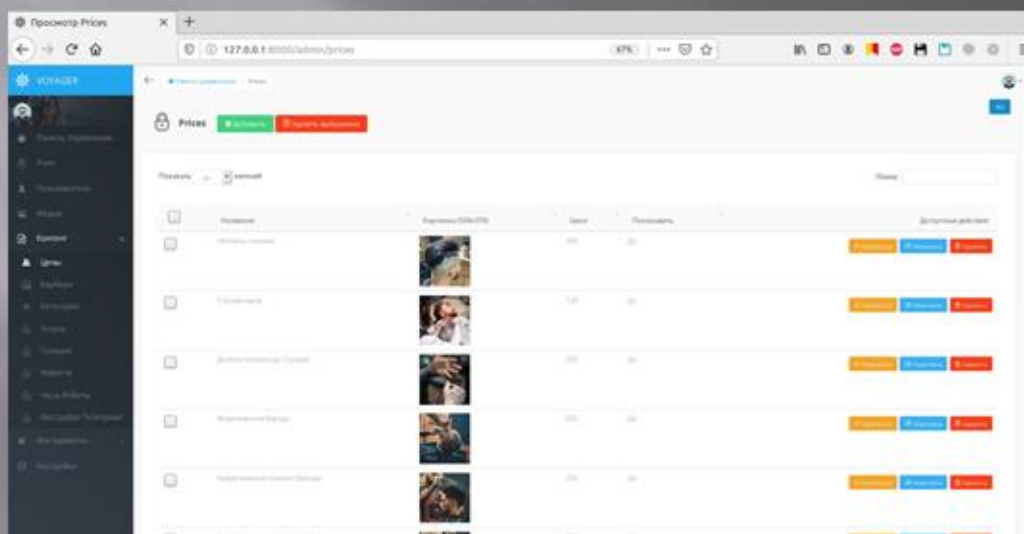
РЕЗУЛЬТАТ РОБОТИ

Сторінка галерея



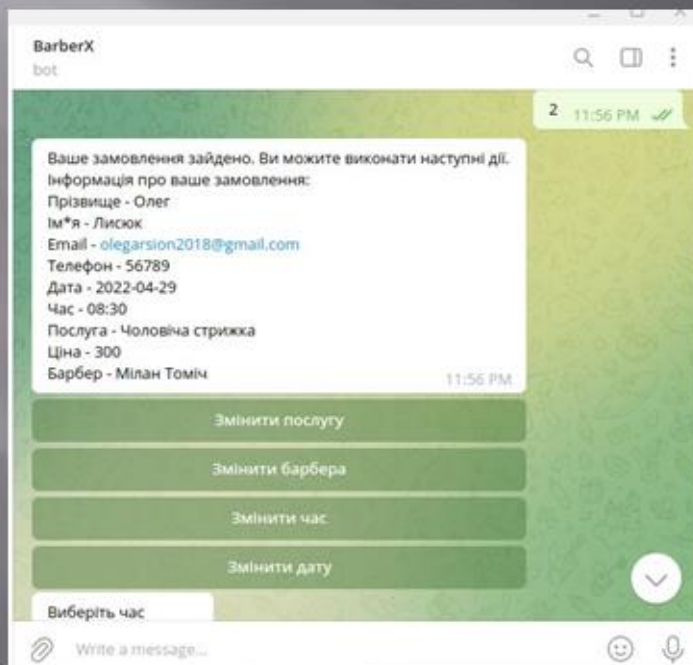
РЕЗУЛЬТАТ РОБОТИ

Адмін панель



РЕЗУЛЬТАТ РОБОТИ

Зв'язок з телеграм ботом



Дякую за увагу

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

здобувача вищої освіти

Лисюка Олега Валентиновича

Питання, запит

факультет ІТ, 3 курс, група ІПЗс-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення 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 плагіату в текстах робіт.

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

06.06.22

дата



Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 14.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 13%

ID: 104690 Назва: Інтернет-платформа для надання перукарських послуг Додано в БД: 2022-06-07 Автора: О.В. Лисюк Керівники: Ю.В. Форкун Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	92716	861	23619 (25%)	252 (29%)

Джерело плагиату

ID	Опис	Наявність плагиату в документі	
		Символи	Лексеми
103277	Назва: Звіт з переддипломної практики Додано в БД: 2022-05-04 Автора: Лисюк О. В Керівники: Форкун Ю.В. Консультанти: Опоненти:	13325 (14.0%)	130 (15.0%)



Ім'я користувача:
Кафедра ІПЗ

Дата перевірки:
07.06.2022 17:12:23 EEST

Дата звіту:
07.06.2022 17:12:49 EEST

ID перевірки:
1011493174

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005589

Назва документа: Лисюк ІПЗс-19-1_Диплом (2)

Кількість сторінок: 99 Кількість слів: 14944 Кількість символів: 114468 Розмір файлу: 10.81 MB ID файлу: 1011369387

19.5% Схожість

Найбільша схожість: 5.33% з джерелом з Бібліотеки (ID файлу: 1011233803)

5.0% Джерела з Інтернету 40

Сторінка 107

1.1% Джерела з Бібліотеки 1

Сторінка 108

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи 103

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗИЯ НА ДИПЛОМНИЙ ПРОЕКТ
освітнього ступеня «Бакалавр»

Дипломник Лисюк Олег Валентинович

Тема Інтернет-платформа для надання перукарських послуг

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг дипломного проекту:

Кількість листів креслень 0 ; кількість сторінок записки 96

1. Короткий зміст пояснювальної записки та прийнятих рішень. У дипломному проекті було досліджено і проаналізовано предметну область перукарських послуг та виявлено потреби потенційних користувачів програмного продукту. Був проведений детальний аналіз існуючих рішень, визначено їх переваги і недоліки, та доведено актуальність розробки нового програмного забезпечення. Було розробити технічне завдання, архітектуру веб-додатку та бази даних, вибрано технології для розробки, розроблено зручний та привабливий для користувачів інтерфейс, розроблено програмний продукт, а також протестовано готову інтернет-платформу.

2. Висновок про відповідність проекту поставленому завданню. Дипломний проект виконаний відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки та передових методів роботи. У вступі доведено актуальність теми, визначено мету та завдання дипломного проектування. У першому розділі проведено аналіз предметної області інтернет-платформа для надання перукарських послуг, розглянуто існуючі рішення та визначені функціональні і нефункціональні вимоги до розроблюваного програмного забезпечення. Було створено веб-сайт з зрозумілим інтерфейсом, який містить у собі весь детальний опис про послуги, ціни та барберів, створено систему онлайн запису з переходом у телеграм канал. У другому розділі проведено аналіз сучасних архітектур, розглянуто їх переваги і недоліки та визначено, що система буде відповідати монолітній архітектурі та моделі клієнт-сервер. Було створено базу даних відповідно до потреб проекту. У третьому розділі підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, в результаті чого створено програмний продукт. В четвертому розділі було виконано тестування системи та проведено його у відповідності до функціональних вимог, в результаті було підтверджено правильну роботу програми.

4. Позитивні сторони проекту. Тематика дипломного проекту є актуальною, оскільки користувачі потребують простого та зрозумілого сайту перукарських послуг, який дозволяє надавати всю доступну інформацію, про ціни, послуги, барберів, тощо. Цей проект також буде містити онлайн запис на візит, що завдяки цієї функції додає сайту більше можливостей та буде кращим над іншими веб-сайтами у цій галузі. Також було застосовано новітні технології для побудови програмного продукту та актуальні архітектурні рішення.

5. Негативні сторони проекту Проект реалізований лише з однією мовою, без можливості зміни. Було б доцільно додати ще одну-дві мови з можливістю вибору.

6. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення виконано відповідно до теми дипломного проекту та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про дипломний проект в цілому Дипломний проект заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики дипломного проекту. Графічний матеріал дає можливість наочно побачити деталі проектування системи.

8. Інші зауваження _____

9. Оцінка дипломного проекту Дипломний проект виконаний у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «задовільно».

РЕЦЕНЗЕНТ

Тетяна Говорученко, д.т.н., проф., зав. кафедрч КІКС ХНУ

" 30 " травня 2022 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: Інтернет-платформа для надання перукарських послуг

Автор: Лисюк Олег Валентинович

Спеціальність: 121 – Інженерія програмного забезпечення

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

Науковий керівник: Форкун Юрій Вікторович, 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, виправданій 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 та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій 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 помилки текстові спотворення, передбачувані спроби ухитити 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 дипломної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, бланк завдання), у структурі змісту, назвах розділів/підрозділів тощо та в 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) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, склав 8,27% і адресується до 364 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломного проекту.

Керівник



Ю.В. Форкун

Гарант ОП



Л. П. Бедріпок

Завідувач кафедри



Л. П. Бедріпок