

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

ДИПЛОМНИЙ ПРОЕКТ

Інтернет-платформа «The World of Watches» для організації продажу годинників
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.190161.01.12.ПЗ

Виконав студент III курсу група ПЗс-19-1


Підпис

Д.А. Стрілець
Ініціали, прізвище

Керівник канд. пед. наук, доцент.
Науковий ступінь, звання


Підпис

О.Г. Онишко
Ініціали, прізвище

Нормоконтролер канд. пед. наук, доцент


Підпис

Н. І. Праворська
Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення


Підпис

Л. П. Бедратюк
Ініціали, прізвище

6 червня 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 ІПЗ

Л. П. Бедратюк 

05 02 2022 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Стрільцю Дмитру Анатолійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інтернет-платформа «The World of Watches» для організації продажу годинників

Керівник проекту (роботи) Онишко Оксана Григорівна, канд. пед. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2022 р. № 18

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, 15 шт.)

6. Консультанти розділів дипломного проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання пр
Нормоконтроль	Праворська Н. І., доцент кафедри ІПЗ		
Антиплагіат	Гурман І. В., доцент кафедри ІПЗ		

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

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Прим
1 Ознайомлення з тематикою дипломного проектування (ДП), визначення та узгодження індивідуальних там 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, нормконтроль, 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

Д.А. Стрілець
Ініціали, прізвище

Керівник проекту (роботи)


Підпис

О. Г. Онишко
Ініціали, прізвище

АНОТАЦІЯ

Тема дипломного проекту: Інтернет-платформа «The World of Watches» для організації продажу годинників

Автор проекту: Стрілець Дмитро Анатолійович.

Керівник проекту: Онишко Оксана Григорівна.

Пояснювальна записка: 140с., 51 рис., 7 табл., 4 дод., 13 джерел.

Графічна частина: 15 слайдів.

ІНТЕРНЕТ-МАГАЗИН, PHP, Yii2, BOOTSTRAP, ВІДЖЕТ, AJAX, БД, MYSQL.

Метою проекту є розробка інтернет платформи яка допоможе організувати продаж різноманітних годинників онлайн, та допоможе користувачам обрати необхідний годинник, шляхом фільтрації, переглядом відгуків та детальний огляд фото.

У дипломному проекті було проаналізовано специфіку роботи інтернет-платформ, проведено аналіз схожих програмних рішень, визначено функціональні вимоги. Проведено аналіз існуючих методів та технологій для розробки веб-додатків та спроектовано архітектуру та адаптивний дизайн платформи.

Для розробки веб-додатку була використана мова програмування PHP, фреймворк Yii2 та база даних MySQL.

Після розробки платформи було проведено різноманітне тестування на відмовостійкість та відповідність вимогам.

Впровадження розробленого програмного продукту дозволить користувачам з легкістю обирати та купувати годинники, а магазинам підвищувати свій прибуток.

01.06.2022
Дата


Підпис

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	ДППЗ.190161.01.12.ПЗ	Пояснювальна записка	140		
2	A4		Завдання на дипломний проект	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4		Презентаційні матеріали	15		

ДППЗ.190161.01.12.ВД				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав		Стрілець Д.А.		02.06
Керівник		Онишко О.Г.		02.06
Н. Контр.		Праворська Н.		04.06
Зав. Каф.		Бедратюк Л.П.		6.06
Інтернет-платформа «The World of Watches» для організації продажу годинників				
Відомість документів				
		Лім.	Арк.	Аркушів
			1	1
ХНУ, ІПЗс-19-1				

ЗМІСТ

Вступ	6
1 Дослідження предметної області та постановка задачі	8
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей	8
1.2 Аналіз наявного програмно-технічного забезпечення предметної області.	12
1.3 Визначення вимог до інтернет-платформи	20
2 Проектування веб-додатка.....	24
2.1 Архітектура та функціональна структура інтернет-платформи	24
2.2 Аналіз та вибір типу бази даних , проектування бази даних	32
2.3 Проектування модулів інтернет-платформи.....	43
2.4 Проектування інтерфейсу користувача.....	46
2.6 Аналіз та вибір технологій і методів реалізації інтернет-платформи.....	52
3 Програмна реалізація	56
3.1 Розробка бази даних	56
3.2 Розробка програмних модулів.....	65
3.3 Керівництво користувача.....	76
3.4 Технічні характеристики веб-додатка	77
3.5 Завантаження веб-додатка на хостинг.....	79
4 Тестування веб-додатка	83
4.1 Аналіз методів тестування веб-додатка	83
4.2 Тестування веб-додатка	86
4.3 Аналіз результатів тестування веб-додатка	90
Висновки.....	94
Перелік джерел посилання	96
Додаток А Технічне завдання	98

ДППЗ.190161.01.12.ПЗ								
Змн.	Арк.	№ докум.	Підпис	Дата	Інтернет-платформа «The World of Watches» для організації продажу годинників Пояснювальна записка	Літ.	Арк.	Аркуші
		Виконав	Стрілець Д.А.	02.06		4	140	
		Керівник	Онишко О.Г.	02.06		ХНУ, ІПЗс-19-1		
		Н. Контр.	Праворська Н.І.	04.06				
		Зав. Каф.	Бедратюк Л.П.	6.05				

Додаток Б Діаграма "Сутність-зв'язок»	103
Додаток В Вигляд користувацького інтерфейсу	105
Додаток Г Код(лістинг) програми	111
Додаток Д Презентаційні матеріали	132

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		5

ВСТУП

Час – найцінніший ресурс будь якої людини, у всіх він обмежений і тому так важливо бути пунктуальним та правильно розпоряджатись своїм часом.

Ваш смартфон не зробить вас пунктуальним, а ось годинник-так. Наручний годинник-це найзручніша річ для того, аби контролювати час. « -Мені не потрібен годинник: я маю телефон», - це можна почути від більшості молодих людей. Регулярно діставати смартфон, щоб подивитися котра година- не найкращий варіант. Погодьтеся, беручи до рук телефон, ви не просто подивитесь на час. «Треба ж перевірити соціальні мережі, новий пост, переглянути блог..», -такі плани побудує ваш мозок. Ви витратите купу дорогоцінного часу у гаджеті, замість того, аби вмить глянути на годинник і продовжити роботу. Годинник-витончений та елегантний спосіб керувати часом, регулювати плани.

Годинники як інструмент визначення часу давно і надійно прижилися в людській реальності. Романтики та прагматики, художники та бізнесмени, чоловіки та жінки, дорослі та діти - всі хочуть знати котра година в потрібний момент. Скільки хвилин пройшло, скільки залишилося, а ще - які саме моделі і якого бренду виблискують на руці у друга, коханої або конкурента.

З давніх часів, коли людина здогадалася всунути в землю паличку та іншою паличкою взялась відміряти рівні проміжки часу на піску, не вщухають суперечки - яким повинен бути годинник? Він повинен бути точним, і у кожної людини повинна бути можливість придбати собі такий годинник, аби з точністю розпоряджатись своїм дорогоцінним часом і всюди встигати.

Актуальність теми дипломного проектування полягає в тому, що усі люди користуються годинниками, а наявні інтернет-платформи для купівлі годинників не надають усіх необхідних можливостей для вибору необхідного годинника або підбору його за параметрами які вам необхідні. Окрім того, усі наявні рішення надають адміністраторам можливість фільтрувати увесь контент

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		6

на сайті та видаляти відгуки які залишили користувачі, тим самим забираючи у потенційних власників годинника можливість прочитати реальні відгуки від користувача конкретного годинника.

Мета проекту – розробка інтернет-платформи яка дозволить користувачам підбирати та обирати необхідний їм годинник з можливістю переглядати фото у збільшеному масштабі та можливістю перегляду відгуків та рейтингу.

Завдання, які необхідно вирішити для досягнення мети:

- провести аналіз предметної області та визначити її особливості;
- провести аналіз існуючих програмних рішень;
- сформулювати технічне завдання;
- спроектувати архітектуру інтернет-додатку;
- ознайомитися з можливими методами та способами розробки додатку;
- виконати програмну реалізацію проекту використовуючи обраний фреймворк;
- провести тестові випробування додатку;

Результатом вирішення вище представлених завдань буде повноцінна інтернет-платформа яка дозволить користувачам легко та зручно підбирати підходящі годинники.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		7

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Годинник - це пристрій та аксесуар, що широко використовуються на зап'ясті, що допомагає нам контролювати час. Наручні годинники розвиваються з сімнадцятого століття. Ви можете знайти годинники всіх видів, від дуже дешевих, виготовлених на основі пластику, до дорогих, які виготовляються з дорогоцінним камінням.

Будь-який предмет, натхненний військовими, має функціональне застосування. Наручний годинник включно. Як вже сказано, наручний годинник вперше використали в 19 столітті військовими для синхронізації маневрів під час війни. З тих пір спеціалізовані годинники були в глибинах океану з дайверами і високо в небі з авіаторами.

Однією з великих переваг – особливо порівнюючи з смартфоном – є те, як довго він може працювати. Багато годинників виготовляються або з автономним живленням, або з невеликою кількістю енергії від акумулятора. Подумайте тільки, – роки без технічного обслуговування ... а скільки прослужить смартфон? Кращі годинники не використовують новітні технології. Вони оснащені потужною механічною технологією годинникового механізму, яка перемагає електриці. Тому, коли на вашому мобільному телефоні сідає акумулятор, ви можете довіряти технології, яка продовжує працювати на вашому зап'ясті.

Коли ви носите годинник – менш ймовірно, що ви відкриєте свій телефон як відволікання. Часто перевірка години по телефону призводить до “зависальної” діяльності, включаючи перевірку кожної програми-месенджера, електронної пошти і Facebook.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		8

Коли вам потрібно відстежувати щось настільки ж важливе, як час – годинник, як унікальний пристрій для вимірювання і управління, виправдана річ.

З часів сонячного годинника (і, ймовірно, раніше) люди були одержимі пристроями, які вимірюють час, адже наш час обмежений.

Носіння годинника позитивно впливає на ставлення людей до часу. Ми стаємо більш уважними до того, як ми проводимо свої 24 години на добу.

Саме тому, годинники є надзвичайно популярним приладом для вимірювання часу, який хоче собі кожен без винятку, і тоді постає питання де ж можна придбати годинник?

Раніше усю нішу продажу годинників займали так звані офлайн магазини, де люди могли використати головну перевагу таких магазинів – фізичний контакт з товаром який може бути купленим, але наразі порівнюючи переваги та недоліки офлайн та онлайн магазинів можна з легкістю визначити який тип всетаки більш популярний та підходить усім без винятку.

Ще нещодавно головним аргументом прихильників «реальних» торгових точок був рівень довіри покупців, які побоювалися здійснювати покупки «наосліп». Необхідність передоплати і велике число інтернет-шахраїв також гальмували розвиток онлайн-продажів. Поступово тенденція змінилася. Переваги інтернет-магазинів як для компаній, так і для покупців ставали все більш очевидними. А в умовах пандемії COVID-19, онлайн-магазини перестали бути альтернативою, перетворившись в деяких країнах єдиним доступним варіантом для покупців. Давайте розберемо основні переваги інтернет-продажів для бізнесу і споживачів не тільки в умовах карантину.

Головні переваги онлайн-торгівлі для бізнесу можна розділити на три блоки, такі, як: економія бюджету, економія часу та ефективність.

Витрати на оренду торгових площ. Інтернет-магазину не потрібно торговий зал, офіс продажів і навіть склад, якщо бізнес використовує метод прямих поставок (дропшипінг). Необхідність таких вкладень особливо вдаряє по підприємцю на самому початку розвитку.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		9

Регулярні витрати на обслуговування магазину. Комунальні платежі, прибирання приміщень та інші необхідні витрати на забезпечення роботи офлайн точки часто вельми обтяжливі для власників. Сайт вимагає мінімальних постійних витрат на оплату хостингу і домену.

Поріг входження. Запуск сайту може обійтися в скромну суму – до \$ 1500 доларів. У більшості випадків цього достатньо для створення, наповнення і початку роботи онлайн-магазину.

Рекламний бюджет. Прихильники оффлайн-магазинів часто стверджують, що початковий потік відвідувачів не вимагає вкладень в рекламу: проходячи мимо, перехожі автоматично будуть заходити всередину. Ось тільки оформлення вітрини і облаштування торгового залу також відносяться до витрат на рекламу. Малоймовірно, що покупців можна буде залучити без них. SEO-просування не вимагає великих грошей, порівняно з рекламою оффлайн магазину. Противники скажуть, що перших результатів доведеться чекати кілька місяців і на старті в будь-якому випадку необхідно купувати рекламу. Тільки витрати на неї значно нижчі, ніж залучення клієнтів в оффлайн-магазин.

Ризики: вкладення в нерентабельний товар, прострочення та інші небезпеки. Власник роздрібного магазину в разі непередбачених обставин або помилок планування ризикує. Прострочений товар, продукція, яка не користується популярністю, карантин і інші неприємності вимагають додаткових витрат. Втрати від невдалої діяльності інтернет-проекту набагато скромніше.

Старт продажів. Перші замовлення з сайту можуть з'явитися вже через 2-3 тижні після того, як ви почали створювати проект. Навіть пошуки торгових площ під магазин зазвичай займають набагато більше часу.

Швидкість виходу на самоокупність. Основна причина – низькі вкладення в старт проекту при створенні сайту і швидкий початок продажів. У цих умовах вже в перші тижні можна покрити всі витрати і отримувати прибуток.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		10

Автоматизація. Функціонування онлайн-майданчики можна повністю або частково автоматизувати, що дозволяє економити час, сили і грошові витрати. У невеликих магазинах практично всі обов'язки може виконувати одна людина.

Охоплення аудиторії. Один з найважливіших ознак, чим відрізняється інтернет-магазин від звичайного, – відсутність жорсткої прив'язки до місця розташування. Продажами можна управляти фактично з будь-якої точки світу.

Аналітика. Ця перевага інтернет-магазину підвищує ефективність бізнесу без особливих складнощів, часу і трудовитрат. Налаштування сервісів аналітики дають можливість зібрати докладну інформацію про відвідувачів сайту, особливості їхньої поведінки, перетворення в покупців і т. д. Все це є основою для більш ефективної роботи з цільовою аудиторією і, як результат, зростання прибутковості.

Тимчасові обмеження. Цілодобовий режим роботи офлайн точки вимагає серйозних фінансових витрат, а встановлений графік може не збігатися з можливостями клієнтів. Замовлення в інтернет-магазині приймаються в будь-який час, незалежно від зайнятості власника, операторів або менеджерів.

Асортимент. В рамках онлайн-проекту практично немає обмежень по розширенню асортименту продукції. Всі витрати, пов'язані зі змінами, стосуватимуться лише актуалізації інформації на сайті. Ще один плюс – можливість швидкої зміни сфери діяльності і заміни товарної лінійки в залежності від потреб покупців і тенденцій бізнесу.

Але незважаючи на переваги інтернет-магазинів користувачі не поспішають купувати та довіряти особисту інформацію а також власні кошти невідомим інтернет-платформам, проте завдяки системам кібербезпеки покупки в інтернеті перестають лякати користувачів. Дані надійно захищені, і будь-які схеми шахрайства швидко розкриваються. Відбувається зростання довіри до онлайн-магазинів, а з ним і усвідомлення переваг онлайн-покупок перед необхідністю походу до «реальних» гіпермаркетів або бутіків.

Серед найважливіших плюсів покупці виділяють :

– необмеженість вибору;

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		11

- покупки в режимі 24/7;
- більш привабливі пропозиції за рахунок економії підприємців;
- не потрібно витратити час на пошуки потрібного товару в звичайних магазинах;
- безпеку даних і конфіденційність;
- зручний сервіс: порівняння товарів, фільтрація, оповіщення про появу продукції, автозамовлення, примірка будинку і т.д .;
- наявність відгуків.

Оцінюючи плюси і мінуси відкриття інтернет-магазину, все частіше підприємці визнають, що цей тип торгівлі дає масу переваг при менших ризиках і вкладеннях, саме тому доцільно розробляти саме інтернет магазин.

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

На сьогоднішній день інтернет простір налічує безліч інтернет-магазинів де можна купити годинники, проте більшість з них спеціалізуються не лише на годинниках а тому їм приділено не багато уваги на цих платформах. Лише деякі з схожих програмних рішень можуть похизуватись великою клієнтською базою та лише позитивними відгуками, саме тому необхідно провести їх аналіз та дослідити усі помилки аби не допустити їх при розробці проекту.

Першим розглянутим програмним рішенням є інтернет-магазин best-time.biz. Інтерфейс веб-сайту є досить простим, зручним та адаптивним, немає зайвої нагромадженості інтерфейсу яка б заважала нормально користуватись усіма можливостями ресурсу. Магазин відповідає майже усім вимогам, проте деякі з них або взагалі не доступні або мають помилки у реалізації. Однією з таких помилок є можливість залишити відгук не зареєстрованим користувачам. Відгуки які залишили користувачі які не купували товар та не користувались

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		12

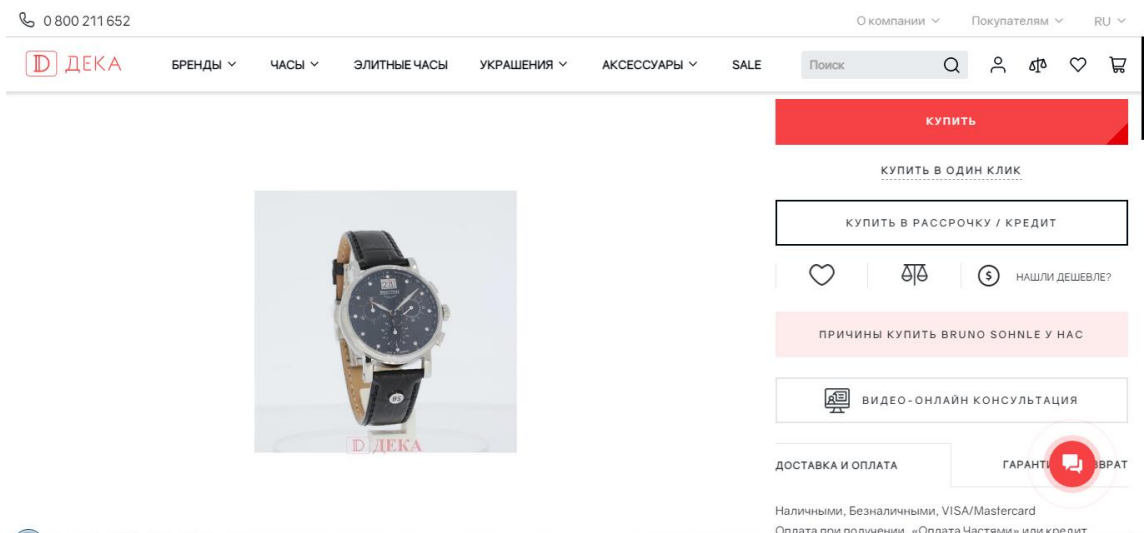


Рисунок 1.3 – Відображення товару у форматі 360 в інтернет-магазині deka.ua

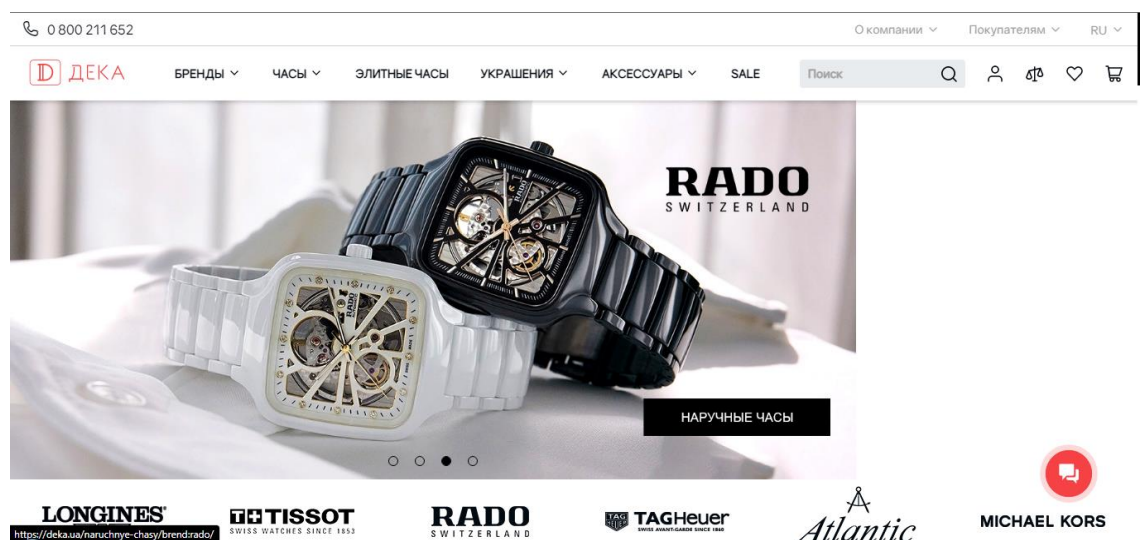


Рисунок 1.4 – Головна сторінка інтернет-магазину deka.ua

art-de-luxe.com.ua – це інтернет-магазин який спеціалізується не лише на годинниках, тому їм приділено не багато уваги. Насамперед можна побачити що на сайті міститься досить слабка система фільтрації, всього лише за трьома полями, що не забезпечує належним чином можливість підбору необхідного годинника. Перевагою є те, що відгуки можуть залишати лише авторизовані користувачі, але в той же час можливість авторизуватись на платформі не працює, це можна побачити на рисунку 1.5. Також недоліком є те, що відгуки

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		15

можна залишати лише для магазину а не для конкретного товару. Відсутня також можливість додати товар до списку бажань та можливість порівняти годинники між собою.

Інтерфейс інтернет-магазину є дуже простим, майже без анімації але з адаптивним дизайном. Інтерфейс веб-сайту art-de-luxe.com.ua можна побачити на рисунку 1.6.

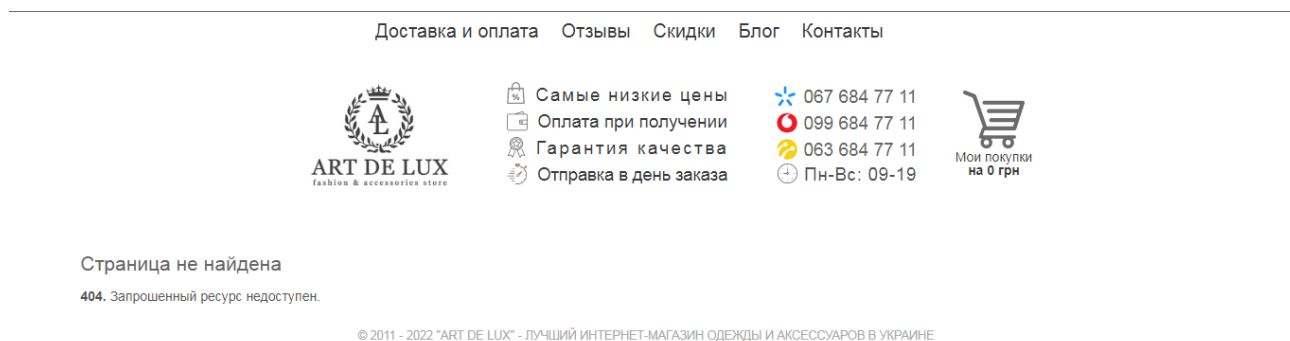


Рисунок 1.5 – Сторінка авторизації на сайті art-de-luxe.com.ua

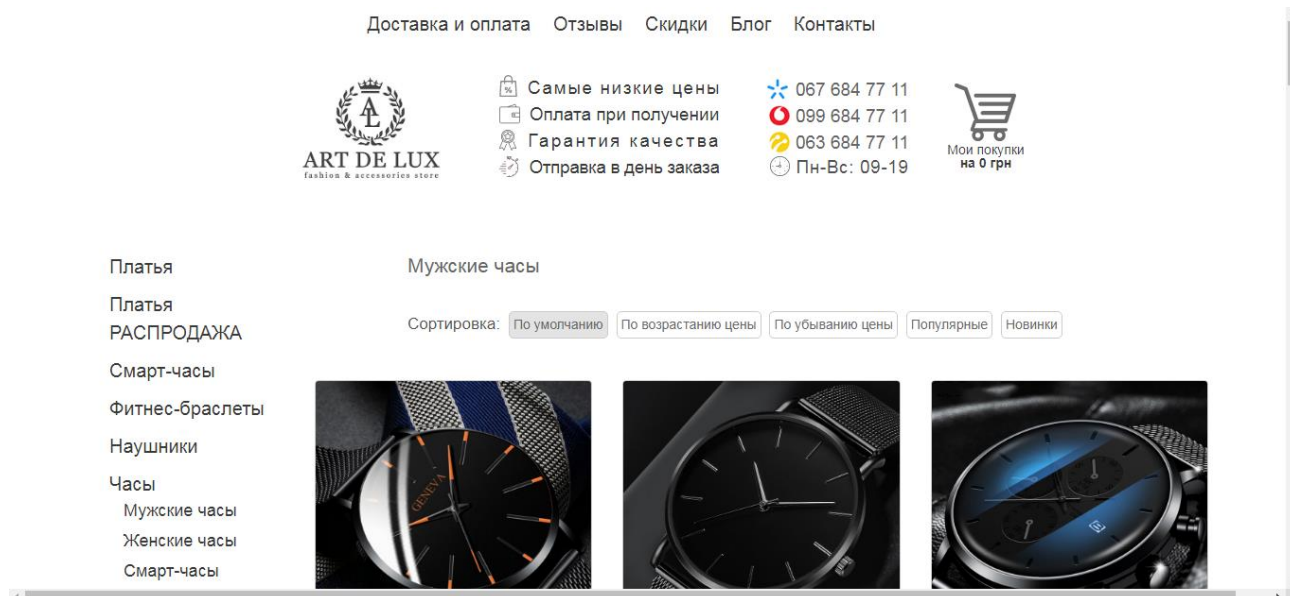


Рисунок 1.6 – Сторінка з чоловічими годинниками на сайті art-de-luxe.com.ua

bezel.com.ua – зручний інтернет-магазин який розгорнуто на платформі для продажів і покупок prom.ua, містить простий та зручний інтерфейс, якщо не

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		16

фільтрації по усіх полях, та можливість сортування. Є також можливість додати товар до списку бажань та можливість порівняти товари між собою. Однією з найбільших переваг є можливість задати питання яке цікавить користувача про конкретну модель годинника. Інтерфейс веб-сайту можна побачити на рисунку 1.8.

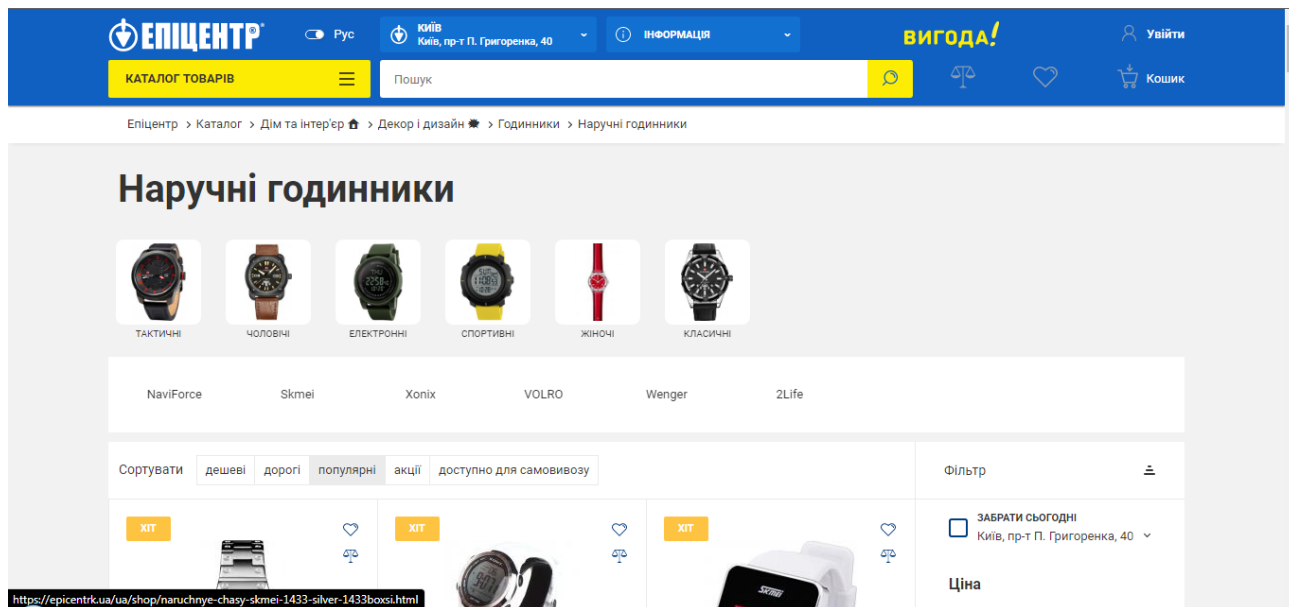


Рисунок 1.8 – Головна сторінка веб-сайту epicentrk.ua

Для більш наочного представлення та порівняння найпопулярніших інтернет-магазинів де можна придбати годинники, між собою, вони були проаналізовані за деякими ключовими критеріями. На основі отриманих даних було складено порівняльну таблицю наявного програмного забезпечення (таблиця 1.1).

Таблиця 1.1 – Порівняльна характеристика наявного ПЗ

Назва	Зручний адаптивний дизайн	Зручна навігація	Фільтрація	Можливість залишити відгук
1	2	3	4	5
best-time.biz	Так	Так	По будь якому полю	Така можливість наявна навіть у не зареєстрованого користувача, тож відгуки можуть бути недостовірними.

Кінець таблиці 1.1.

1	2	3	4	5
deka.ua	Ні. Безліч спливаючих вікон які заважають нормально користуватись веб-сайтом.	Так	По будь якому полю	Така можливість наявна навіть у не зареєстрованого користувача, тож відгуки можуть бути недостовірними.
art-de-luxe.com.ua	Ні. Дуже простий інтерфейс, з нескінченним завантаженням товарів яке ніяк не можна зупинити, через що не можливо переглянути нижню частину веб-сайту	Ні	Фільтрація доступна лише за декількома параметрами	Відгук можна залишити лише про сам магазин а не про конкретний товар, перевагою є те, що відгуки можуть залишати лише авторизовані користувачі, але кнопка логіну не працює
bezel.com.ua	Так	Так	По будь якому полю	Відгуки можна залишати лише після покупки конкретного годинника
epicentrk.ua	Так	Так	По будь якому полю	Відгуки можна залишати не авторизованому користувачу, є поле де користувач вказує чи купував він годинник, і скільки користується

Завдяки проведеному аналізу наявного програмно-технічного забезпечення було визначено усі недоліки та переваги розглянутих рішень, що дозволило визначити, що розроблювана система повина:

- мати зручний, інтуїтивно зрозумілий та адаптивний дизайн;
- мати систему фільтрації за усіма полями.
- мати можливість залишити відгук авторизованим користувачам.

- мати зручну навігацію;
- мати можливість додати товар у список бажань;
- мати можливість додати товар для порівняння.

1.3 Визначення вимог до інтернет-платформи

Після розгляду схожих програмних рішень можемо сформулювати ряд вимог яким повинна відповідати інтернет-платформа, найзручніше їх можна зобразити у вигляді UML діаграми варіантів використання (UseCase Diagram).

Універсальна мова моделювання (Unified Modelling Language або UML) — це мова позначень або побудови діаграм, призначена для визначення, візуалізації і документування моделей орієнтованих на об'єкти систем програмного забезпечення.

Діаграми варіантів використання описують взаємозв'язки і залежності між групою варіантів використання і акторами, що беруть участь у процесі.

Важливо зауважити, що діаграми варіантів використання не призначено для показу компонування, вони не можуть описати внутрішню структуру системи. Їх призначено для полегшення обміну інформацією між майбутніми користувачами системи і замовником, вони особливо корисні для визначення переліку можливостей, які повинна мати система. У діаграмі варіантів використання можна сказати, що система має робити, але не те, як вона досягає потрібних результатів, для останнього ці діаграми просто не придатні.

У таблиці 1.2 подано опис акторів тобто користувачів в таблиці 1.3 – варіанти використання для кожного актора.

Таблиця 1.2 – Опис акторів

Актор	Короткий опис
1	2
Гість (неавторизований користувач)	Може переглядати та фільтрувати каталог годинників, може переглядати конкретні годинники, переглядати список новин, додавати товари у кошик та оформлювати замовлення.

- можливість залишити відгук;
- можливість переглянути наявні відгуки;
- можливість авторизації та реєстрації;
- можливість перегляду останніх новин;
- можливість додати товар у обране;
- можливість порівняти товари;
- Можливість перегляду конкретного товару;
- Можливість оформити замовлення.

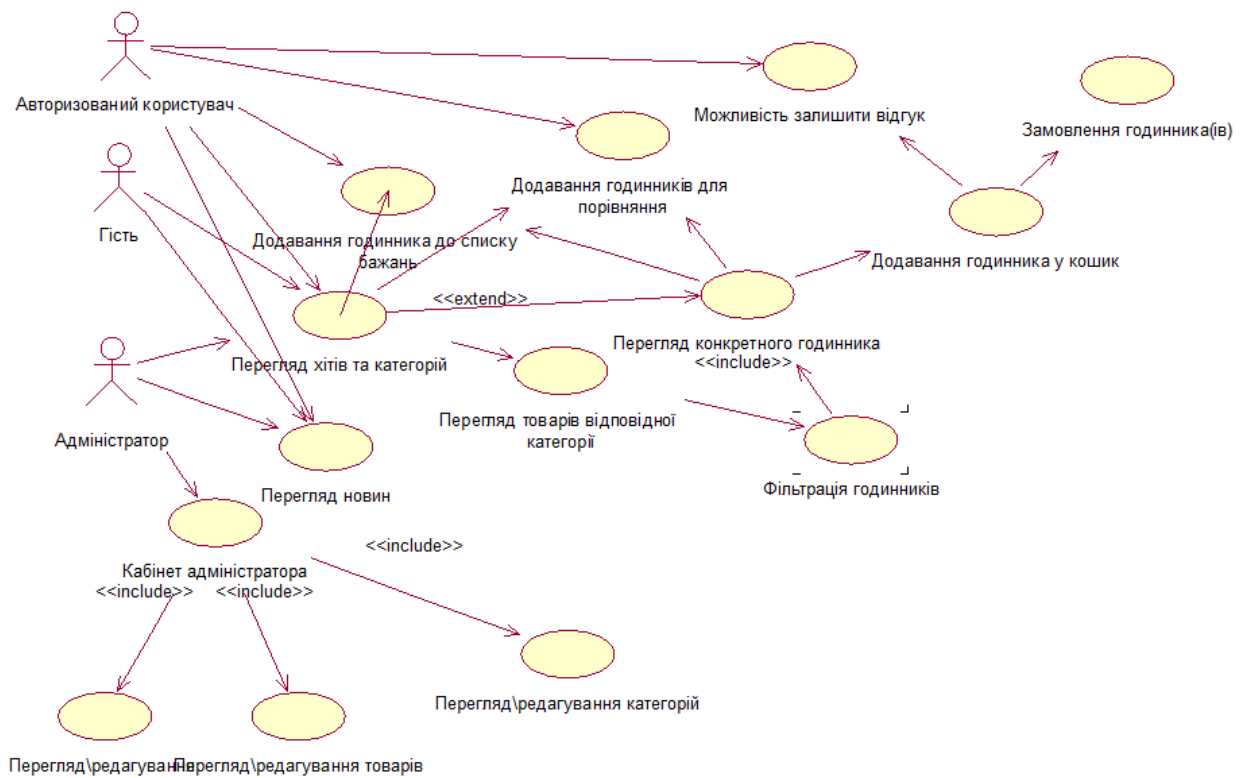


Рисунок 1.9 – Діаграма UseCase(варіантів використання)

2 ПРОЕКТУВАННЯ ВЕБ-ДОДАТКА

2.1 Архітектура та функціональна структура інтернет-платформи

Архітектурний шаблон - це узагальнене рішення поширеної проблеми в архітектурі програмного забезпечення в заданому контексті.

Шаблон - це вирішення завдання в певному контексті.

Для вибору найкращої архітектури та архітектурного шаблону розглянемо декілька основних видів архітектурних шаблонів:

- Багаторівнева архітектура;
- Канали і фільтри;
- Клієнт – сервер;
- Модель - представлення – контролер;
- Керована подіями архітектура;
- Архітектура на основі мікросервісів.

Найпоширеніший архітектурний шаблон - багаторівнева архітектура (або «n-рівнева»). Вона добре відома більшості архітекторів, проєктувальників і розробників. Обмежень за кількістю і типом рівнів ніяких немає, проте в більшості випадків така архітектура складається з чотирьох рівнів: представлення даних, бізнес-логіка, зберігання даних і база даних. Схема багаторівневої архітектури зображена на рисунку 2.1.

У будь-яких складних системах необхідно забезпечувати незалежну розробку і розвиток окремих частин. Тому розробникам системи треба зрозумілий і добре задокументований поділ завдань, щоб модулі системи можна було розробляти і підтримувати незалежним чином.

Програмне забезпечення необхідно розділити так, щоб модулі можна було розробляти і розвивати окремо - з мінімальною взаємодією між частинами, забезпечуючи переносимість, модифікацію і повторне використання.

Щоб домогтися такого поділу, при використанні багаторівневого шаблону, програмне забезпечення поділяється на сутності, звані рівнями. Кожен

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		24

рівень - група модулів, що надають взаємопов'язаний набір сервісів. Їх застосування має бути односпрямованим. Рівні повністю поділяють ПО, причому кожна частина доступна через публічний інтерфейс.

Перша ідея полягає в тому, що у кожного з рівнів є певні роль і відповідальність. Наприклад, рівень представлення даних відповідає за обробку всього призначеного для користувача інтерфейсу. Такий поділ завдань в багаторівневій архітектурі спрощує формування ефективних ролей і відповідальності.

Друга ідея полягає в тому, що шаблон багаторівневої архітектури є технічне поділ - на відміну від поділу по доменах: тобто, це групи компонентів, а не домени.

І, нарешті, третя ідея: кожен рівень позначається як закритий або відкритий. Запит, переміщаючись з рівня на рівень і минаючи закритий рівень, повинен обов'язково пройти через нього - пропустити закритий рівень не можна.



Рисунок 2.1 – Схема багаторівневої архітектури

Наявність рівнів знижує продуктивність. Цей шаблон не підходить для високопродуктивних застосунків: проходити кілька рівнів архітектури для виконання бізнес-запиту - це неефективно.

Крім того, додавання рівнів збільшує повну вартість і ускладнює систему.

Використовувати цей підхід слід для невеликих, простих застосунків і веб-сайтів. Також цей шаблон добре підходить, якщо бюджет і час обмежені.

Канали і фільтри

Один з шаблонів які часто використовуються в архітектурі ПО - шаблон каналів і фільтрів. Схема шаблону канали та фільтри зображена на рисунку 2.2.

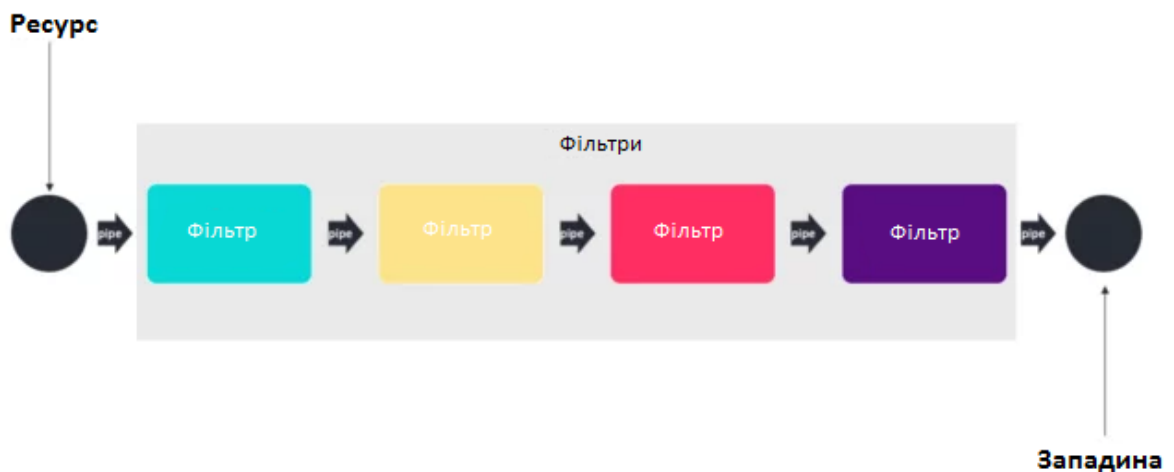


Рисунок 2.2 – Схема шаблону канали та фільтри

Часто в системах необхідно перетворювати потоки дискретних елементів даних від введення до виводу. На практиці багато типів перетворень повторюються неодноразово, тому бажано зробити з них незалежні компоненти які можна повторно використовувати.

Такі системи необхідно розділяти на повторно використовувані слабо пов'язані компоненти з простими узагальненими механізмами взаємодії - таким чином їх можна буде зручно поєднувати один з одним. Слабо пов'язані компоненти з узагальненою взаємодією легко використовувати повторно. А оскільки вони незалежні, їх виконання можна запускати паралельно.

У цій архітектурі фільтри пов'язані між собою комунікаційними каналами. Перша ідея полягає в тому, що кожен з каналів з міркувань продуктивності має тип «точка - точка» і є односпрямованим: приймає вхідні дані від одного джерела і направляє вихідні дані в приймач.

У такому підході виділяють фільтри чотирьох видів:

- генератор (source) - відправна точка процесу;
- перетворювач (map) - виконує перетворення деяких або всіх даних;
- випробувач (reduce) - перевіряє один або кілька критеріїв;
- споживач (sink) - кінцева точка.

Чи не найкращий вибір для інтерактивних систем, оскільки така архітектура орієнтована на перетворення даних.

Надмірне використання синтаксичного аналізу і синтезу знижує продуктивність і ускладнює написання самих фільтрів.

Архітектура каналів і фільтрів застосовується в найрізноманітніших застосунках, особливо при вирішенні завдань, що забезпечують просту односторонню обробку - наприклад, інструменти EDI (електронний обмін даними), ETL (витяг, перетворення і завантаження).

Клієнт - сервер

Є загальні ресурси та сервіси, до яких потрібно забезпечити доступ великої кількості розподілених клієнтів, і при цьому необхідно контролювати доступ або якість обслуговування. Схема архітектури клієнт-сервер зображена на рисунку 2.3.

Керуючи набором загальних ресурсів і сервісів, можна забезпечити модифікацію і повторне використання, для чого загальні сервіси виносяться окремо, щоб їх можна було змінювати в одному місці або в невеликій кількості

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		27

місце. Потрібно поліпшити масштабованість і доступність до використання за рахунок централізованого управління цими ресурсами і сервісами при одночасному розподілі самих ресурсів між декількома фізичними серверами.

У підході «клієнт - сервер» компоненти і сполучні елементи мають певну поведінку:

- Компоненти, звані «клієнтами», відправляють запити компоненту, званому «сервер», і чекають відповіді;
- Компонент «сервер» отримує запит від клієнта і відправляє йому відповідь.

Сервер може бути вузьким місцем щодо продуктивності, а також єдиною точкою відмови. Змінювати прийняте рішення про розміщення функціональних можливостей (на клієнті або на сервері) після створення системи - це зазвичай складно і дорого.

Підхід «клієнт - сервер» можна застосовувати в моделюванні частини системи, що має багато компонентів, які відправляють запити (це «клієнти») іншого компонента (це «сервер»), який забезпечує роботу сервісів, - наприклад, онлайн-застосунки (електронна пошта, обмін документами і банківська справа).

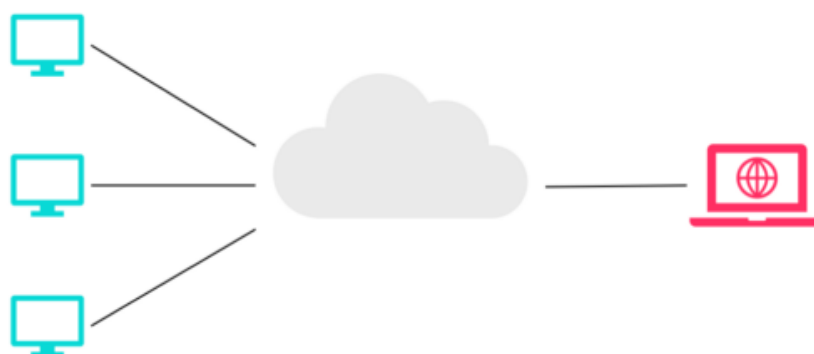


Рисунок 2.3 – Схема шаблону клієнт-сервер

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		28

Модель - представлення – контролер

Зазвичай в інтерактивному застосунку частіше за інших змінюється призначений для користувача інтерфейс. Користувачам потрібно бачити дані в різному поданні - наприклад, як лінійчату або кругову діаграму, - і обидва подання повинні відображати поточний стан даних. Схема шаблону MVC зображена на рисунку 2.4. Шаблон «модель - представлення - контролер» (MVC) розділяє функціональність програми на компоненти трьох видів:

- Модель - містить дані програми;
- Представлення - відображає деяку частину базових даних і взаємодіє з користувачем;
- Контролер - діє як посередник між моделлю і представленням і управляє повідомленнями про зміну стану.

Для простих користувацьких інтерфейсів така складність може бути надмірною. Абстракції «модель», «уявлення» і «контролер» можуть не дуже добре підходити в разі деяких наборів інструментів для розробки користувацького інтерфейсу. Архітектурний шаблон MVC зазвичай використовується в мобільних і веб-застосунках при розробці користувацьких інтерфейсів.

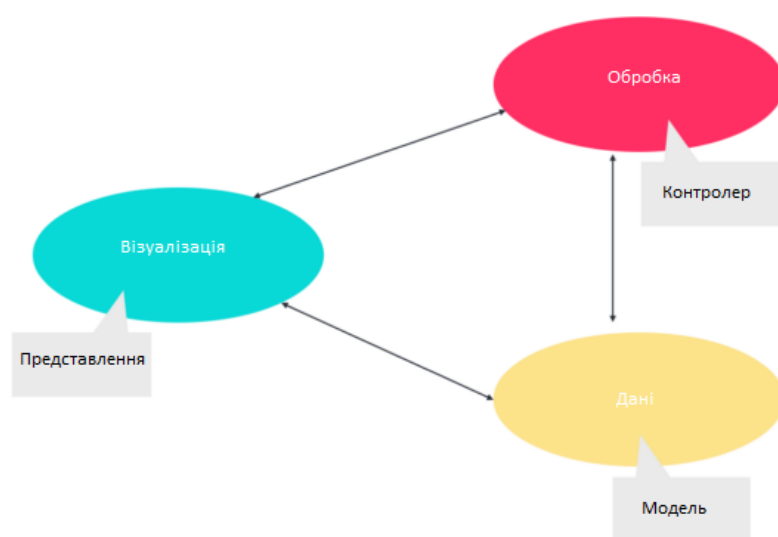


Рисунок 2.4 – Схема шаблону MVC

Керована подіями архітектура

Шаблон архітектури програмного забезпечення, який призначений для створення подій, їх виявлення, споживання і реагування на них.

Подія може бути визначена як значна зміна стану. Наприклад, коли споживач купує автомобіль, стан автомобіля змінюється з «на продаж» до «продано». Архітектура системи дилера автомобілів може трактувати цю зміну стану як подію, поява якої може стати відомою іншим застосункам даної архітектури.

З формальної точки зору, те, що виробляється, публікується, поширюється, виявляється і споживається (як правило, асинхронно) є повідомленням, яке називають сповіщенням про подію (або нотифікацією), а не самою подією, яка є зміною стану, що викликає появу повідомлення.

З недоліків можливі проблемні області - продуктивність і відновлення після помилок. Схема керованої подіями архітектури зображена на рисунку 2.5.

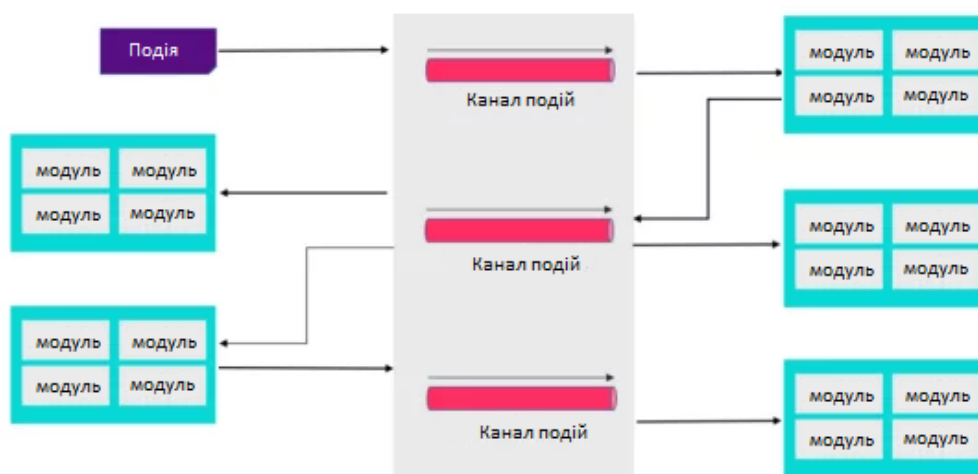


Рисунок 2.5 – Схема керованої подіями архітектури

Архітектура на основі мікросервісів

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		30

Така архітектура підходить для створення застосунків у вигляді наборів сервісів: кожен сервіс розгортається і масштабується незалежно і має власний «кордон», який обслуговується за допомогою API. Різні сервіси можуть писатися на різних мовах програмування, управляти власними базами даних і розроблятися різними командами.

Системи повинні бути спроектовані так, щоб витримувати збої в роботі сервісів, а це вимагає ретельнішого моніторингу систем. Додаткові витрати ресурсів на організацію роботи сервісів і взаємодії подій. Крім того, буде потрібно більше пам'яті.

Архітектура мікросервісів застосовна в багатьох випадках, особливо коли використовується великий конвеєр даних. Наприклад, мікросервісна архітектура – відмінний вибір для системи звітності про продажі в роздрібних магазинах компанії. Кожен крок у процесі підготовки даних буде оброблятися мікросервісом: збір, очищення, нормалізація, збагачення, агрегація даних, звітність і т. д.

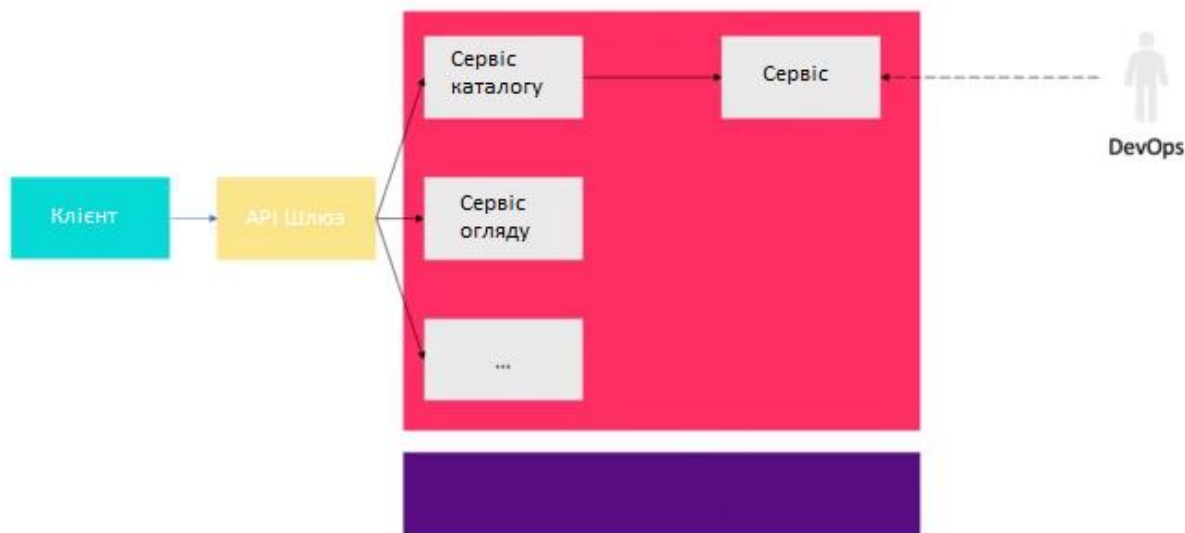


Рисунок 2.6 – Схема архітектури на основі мікросервісів

Розглянувши можливі архітектурні рішення, було прийнято використовувати клієнт-серверну архітектуру з використанням шаблону MVC. Такий вибір дозволить задовольнити усі функціональні та не функціональні вимоги до програмної системи.

2.2 Аналіз та вибір типу бази даних , проектування бази даних

Почнемо з трьох типів БД, які все ще можуть зустрічатися в спеціалізованих середовищах, але в основному замінені надійними і продуктивними альтернативами.

Перший і найпростіший спосіб зберігання даних - текстові файли. Приклад такого файлу зображений на рисунку 2.7. Метод застосовується і сьогодні для роботи з невеликими обсягами інформації. Для поділу полів використовується спеціальний символ: кома або крапка з комою в csv-файлах датасета, двокрапка або пробіл в *nix-подібних системах:

/ Etc / passwd в * nix системі

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
syslog:x:102:106:./home/syslog:/usr/sbin/nologin
bob:x:1000:1000:Bob Smith,,:/home/bob:/bin/bash
```

Рисунок 2.7 – Приклад текстового файлу у якому зберігаються дані

Особливості:

– обмежений тип і рівень складності інформації, що зберігається;

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		32

- важко встановити зв'язки між компонентами даних;
- відсутність функцій паралелізму;
- практичні тільки для систем з невеликими вимогами до читання і запису;
- використовуються для зберігання конфігураційних даних;
- немає необхідності в сторонньому програмному забезпеченні.

Ієрархічні бази даних

На відміну від текстових таблиць, в наступному типі БД з'являються зв'язки між об'єктами. В ієрархічних базах даних кожен запис має одного «батька». Це створює деревоподібну структуру, в якій записи класифікуються за їхніми стосунками з ланцюжком батьківських записів. Приклад ієрархічної бази даних зображено на рисунку 2.8.

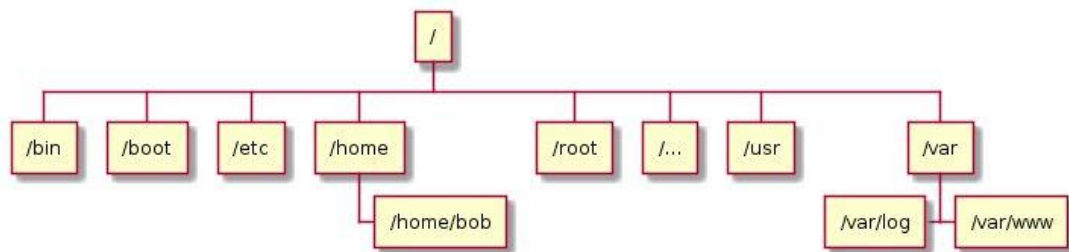


Рисунок 2.8 – Приклад побудови ієрархічних зв'язків

Особливості:

- інформація організована у вигляді дерева з відносинами «предок-нащадок»;
- кожен запис може мати не більше одного з батьків;
- зв'язки між записами виконані у вигляді фізичних покажчиків;
- неможливо реалізувати відносини «багатьох до багатьох».

Мережеві бази даних

Мережеві бази даних розширюють функціональність ієрархічних: записи можуть мати більше одного батька. А значить, можна моделювати складні відносини. Приклад мережевої бази даних зображено на рисунку 2.9.

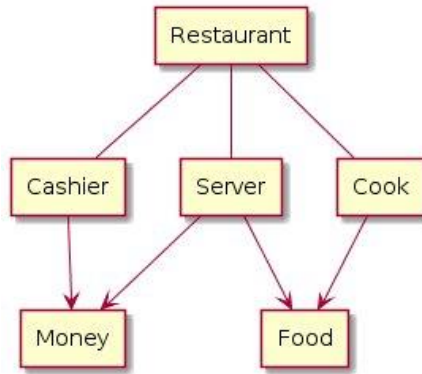


Рисунок 2.9 – Приклад мережевої бази даних

Особливості:

- мережеві бази даних подаються не деревом, а загальним графом
- обмежені тими ж шаблонами доступу, що ієрархічні БД

SQL бази даних

Реляційні бази даних - найстаріший тип досі широко використовуваних БД загального призначення. Дані та зв'язки між даними організовані за допомогою таблиць. Кожен стовпець у таблиці має ім'я і тип. Кожен рядок представляє окремий запис або елемент даних в таблиці, який містить значення для кожного з стовпців. Приклад такої бази даних зображено на рисунку 2.10.

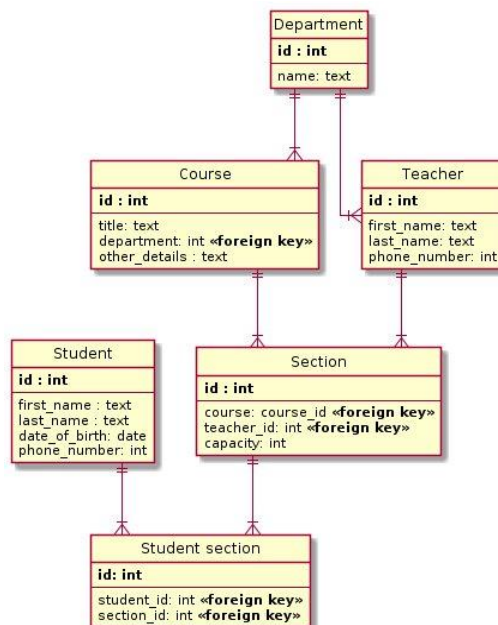


Рисунок 2.10 – Приклад реляційної бази даних

Особливості:

- поле в таблиці, зване зовнішнім ключем, може містити посилання на стовпці в інших таблицях, що дозволяє їх з'єднувати;
- високоорганізована структура і гнучкість робить реляційні БД потужними і такими, що адаптуються до різних типів даних;
- для доступу до даних використовується мова структурованих запитів (SQL);
- надійний вибір для багатьох додатків.

Бази даних «ключ-значення»

У базах даних «ключ-значення» для зберігання інформації ви надаєте ключ і об'єкт даних, який потрібно зберегти. Наприклад, JSON-об'єкт, зображення або текст. Щоб зробити запит, відправляєте ключ і отримуєте blob-об'єкт . Приклад такої бази даних зображено на рисунку 2.11

key:	value
user_id:	f5badc33-5bd7-4b65-a737-b5304675f476
color:	blue
repetitions:	3
text:	hello world
data:	{ ... }

Рисунок 2.11 – Приклад таблиці бази даних «ключ-значення»

Особливості:

- сховища забезпечують швидкий доступ з незначними витратами;
- часто зберігають дані конфігурацій і інформацію про стан даних, представлених словниками або хешем;
- немає жорсткої схеми відносини між даними, тому в таких БД часто зберігають одночасно різні типи даних;
- розробник відповідає за визначення схеми іменування ключів і за те, щоб значення мало відповідний тип / формат.

Документні бази даних

Документні бази даних (також документоорієнтовані БД або сховища документів), спільно використовують базову семантику доступу і пошуку сховищ ключів і значень. Такі БД також використовують ключ для унікальної ідентифікації даних. Різниця між сховищами «ключ-значення» і документними БД полягає в тому, що замість зберігання blob-об'єктів, документоорієнтовані бази зберігають дані в структурованих форматах - JSON, BSON або XML. Приклад такої бази даних зображено на рисунку 2.12.

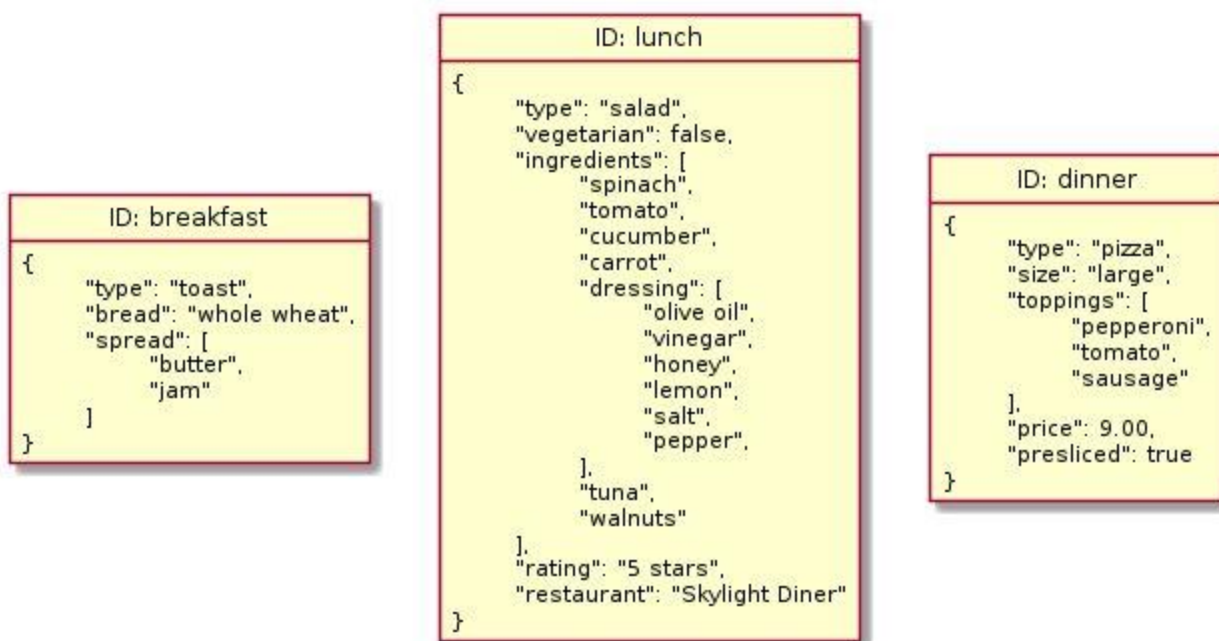


Рисунок 2.12 – Приклад документної бази даних

Особливості:

- база даних не виділяє окремий формат або схему;
- кожен документ може мати свою внутрішню структуру;
- документні БД є хорошим вибором для швидкої розробки;
- в будь-який момент можна змінювати властивості даних, не змінюючи структуру або самі дані.

Графові бази даних.

Замість зіставлення зв'язків з таблицями і зовнішніми ключами, графові бази даних встановлюють зв'язки, використовуючи вузли, ребра і властивості. Приклад такої бази даних зображено на рисунку 2.13.

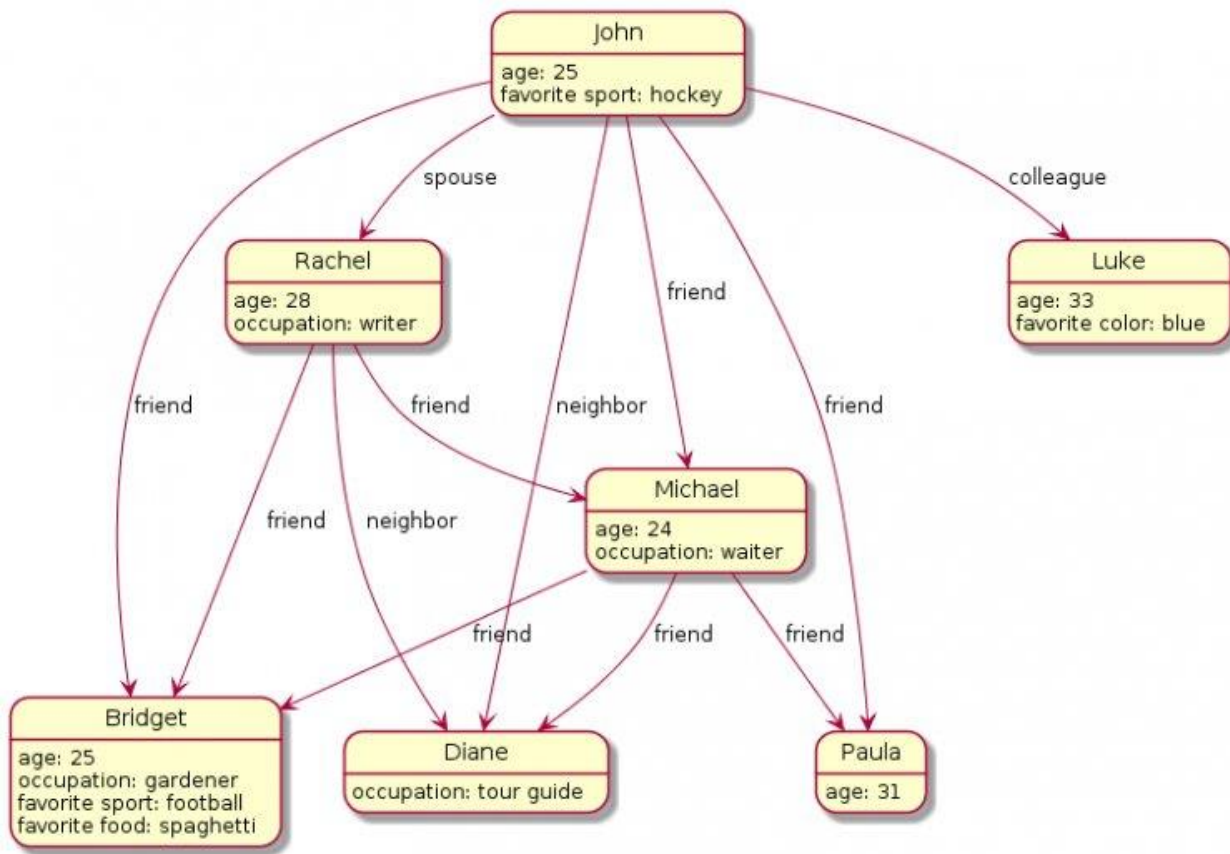


Рисунок 2.13 – Приклад графової бази даних

Графові бази представляють дані у вигляді окремих вузлів, які можуть мати будь-яку кількість пов'язаних з ними властивостей.

Особливості:

- виглядають аналогічно мережевим;
- фокусуються на зв'язках між елементами;
- явно відображають зв'язки між типами даних;
- не вимагають покрокового обходу для переміщення між елементами;
- немає обмежень в типах зв'язків.

Стовбчикові бази даних

Стовбчикові бази даних (також нереляційні колоночні сховища або бази даних з широкими стовпцями) належать до сімейства NoSQL БД, але зовні схожі на реляційні БД. Як і реляційні, стовпчикові БД зберігають дані, використовуючи рядки і стовпці, але з іншим зв'язком між елементами.

У реляційних БД всі рядки повинні відповідати фіксованій схемою. Схема визначає, які стовпчики будуть в таблиці, типи даних та інші критерії. У стовпчикових базах замість таблиць є структури - «стовпчик сімейства». Сімейства містять рядки, кожна з яких визначає власний формат. Рядок складається з унікального ідентифікатора, що використовується для пошуку, за яким слідують набори імен та значень стовпців. Приклад такої бази даних зображено на рисунку 2.14.

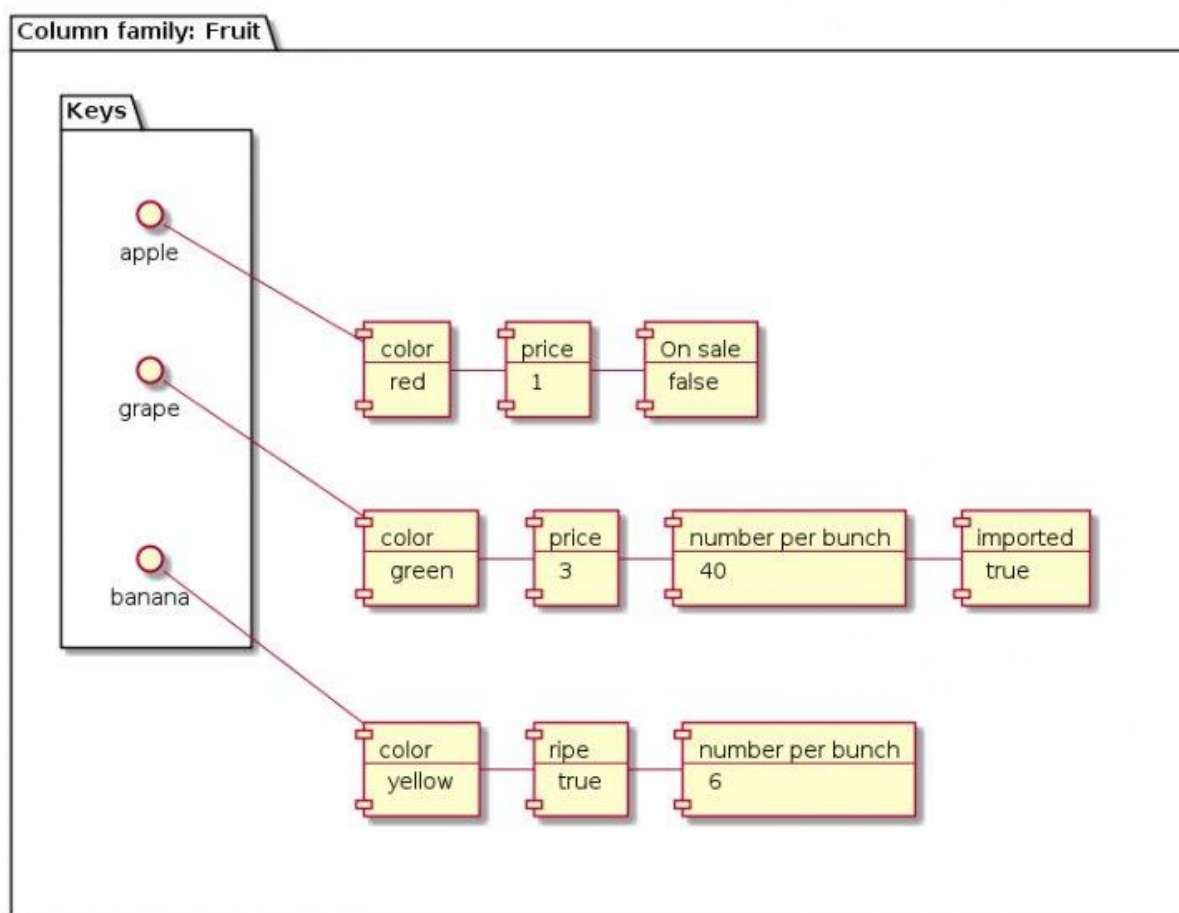


Рисунок 2.14 – Приклад стовпчикової бази даних

Особливості:

- БД зручні при роботі з додатками, що вимагають високої продуктивності;
- дані та метадані доступні по одному ідентифікатору;
- гарантоване розміщення всіх даних з рядка в одному кластері, що спрощує сегментацію і масштабування даних.

Бази даних часових рядів

Бази даних часових рядів створені для збору і управління елементами, що змінюються з плином часу. Більшість таких БД організовані в структури, які записують значення для одного елемента. Наприклад, можна створити таблицю для відстеження температури процесора. У середині кожне значення буде складатися з тимчасової мітки і показника температури. У таблиці може бути декілька метрик. Приклад такої бази даних зображено на рисунку 2.15.

Time	CPU Temp	System Load	Memory Usage %
2019-10-31T03:48:05+00:00	37	0.85	92
2019-10-31T03:48:10+00:00	42	0.87	90
2019-10-31T03:48:15+00:00	33	0.74	87
2019-10-31T03:48:20+00:00	34	0.72	77
2019-10-31T03:48:25+00:00	40	0.88	81
2019-10-31T03:48:30+00:00	42	0.89	82
2019-10-31T03:48:35+00:00	41	0.88	82

Рисунок 2.15 – Приклад бази даних часових рядів

NewSQL бази даних

NewSQL бази даних успадковують реляційну структуру і семантику, але побудовані з використанням більш сучасних, масштабованих конструкцій. Мета - забезпечити більшу масштабованість, ніж реляційні БД, і більш високі гарантії узгодженості, ніж в NoSQL. Компроміс між узгодженістю і доступністю є фундаментальною проблемою розподілених баз даних, описаних теоремою CAP.

Особливості:

- можливість горизонтального масштабування;

- висока доступність;
- велика продуктивність і реплікація;
- невеликий функціонал і гнучкість;
- чимале споживання ресурсів і необхідність спеціалізованих знань для роботи з базою даних.

Багатомодельні бази даних

Багатомодельні бази даних - бази, які поєднують функціональні можливості кількох видів БД. Переваги такого підходу очевидні - одна і та ж система може використовувати різні уявлення для різних типів даних.

Спільне розміщення даних з декількох типів БД в одній системі дозволяє виконувати нові операції, які в іншому випадку були б ускладнені або неможливі. Наприклад, багатомодельні бази можуть дозволити користувачам отримати доступ до даних, що зберігаються в різних типах БД, і управляти ними в рамках одного запиту, а також підтримують узгодженість даних при виконанні операцій, що змінюють інформацію відразу в декількох системах.

Наслідки:

- допомагають зменшити навантаження на СУБД;
- дозволяють розширюватися до нових моделей у міру зміни потреб без внесення змін до базової інфраструктури;
- забезпечують безперервний доступ і простий розподіл даних;
- мають лінійну масштабованість і прості для розробки.

Проаналізувавши розглянуті типи баз даних для розробки дипломного проекту було обрано реляційний тип баз даних MySQL. Дані та зв'язки у таких база даних реалізовані за допомогою таблиць. Усі стовпчики у таблицях мають назву та тип даних, а рядки представляють окремий запис. Невід'ємною частиною процесу розробки таких БД є нормалізація, тобто усунення структурних недоліків, які призводять до надмірності даних. Нормалізація бази даних відбувається поетапно, завдяки послідовному приведенні БД в нормальну форму рівня n. Нормальна форма — властивість відношення в реляційній моделі даних, що характеризує його з точки зору надмірності, яка потенційно може

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		40

привести до логічно помилкових результатів вибірки або зміни даних. Нормальна форма визначається як сукупність вимог, яким має задовольняти відношення. Усього існує 6 нормальних форм та дві додаткових(Бойса-Кодда, домен/ключ), але для нормального функціонування буде достатньо привести БД усього до перших трьох.

Перша нормальна форма утворює ґрунт для структурованої схеми бази даних:

– Кожна таблиця повинна мати основний ключ: мінімальний набір колонок, які ідентифікують запис.

– Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи неключові атрибути.

– Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.

Друга нормальна форма вимагає, аби дані, що зберігаються в таблицях із композитним ключем, не залежали лише від частини ключа:

– Схема бази даних повинна відповідати вимогам першої нормальної форми.

– Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.

Третя нормальна форма вимагає, аби дані в таблиці залежали винятково від основного ключа:

– Схема бази даних повинна відповідати всім вимогам другої нормальної форми.

– Будь-яке поле, що залежить від основного ключа та від будь-якого іншого поля, має виноситись в окрему таблицю.

Для початку необхідно виділити сутності предметної області з яких буде формуватись наше відношення. З функціональних вимог до розроблюваного додатку виокремлюємо наведені в ТЗ дані які необхідно зберігати і отримуємо наступний набір атрибутів:

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		41

Product – представляє сутність яка буде зберігати інформацію про годинники.

Атрибути сутності Product:

- Name – назва товару;
- Content – опис товару;
- Price – ціна товару;
- Img – посилання на фото товару;
- Hit – атрибут який буде показувати чи являється товар хітом;
- New – атрибут який буде показувати чи є товар новинкою;
- Sale – атрибут який буде показувати чи є розпродаж на товар;
- Brand – бренд товару;
- Sex – стать для якої підходить товар;
- Material – матеріал з якого виготовлено годинник;
- Bracelet – атрибут який показує тип кріплення годинника(ремінець або браслет).

User - представляє зареєстрованого користувача в системі.

Атрибути сутності User:

- Username – ім'я користувача;
- Password – пароль користувача.

Category – представляє категорію товарів.

Атрибути сутності Category:

- Name – ім'я категорії

Order – представляє замовлення користувача.

Атрибути сутності Order:

- Created_at – дата оформлення замовлення;
- Updated_at – дата змінення замовлення;
- Qty – кількість продуктів в замовленні
- Sum – сума замовлення;
- Status – статус замовлення;
- Name – ім'я замовника;

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		42

- Email – пошта замовника;
- Phone – телефон замовника;
- Address – адреса замовника.

Order Items – представляє товари які містяться у замовленні.

Атрибути сутності Order Items:

- Name – назва товару;
- Price – ціна товару;
- Qty Item – кількість товару у замовленні;
- Sum Item – сума по конкретному товару у замовленні.

Review – представляє відгуки до товару.

Атрибути сутності Review:

- Rating – рейтинг товару;
- Coment – коментар до товару;
- Date – дата публікації відгуку.

Для кожної сутності необхідно задати ключові поля та привести до третьої нормальної форми. На основі цих сутностей було розроблено діаграму Сутність-зв'язок.

Діаграма Сутність-зв'язок подана у додатку Б.

Саме така архітектура БД дозволить у повному обсязі виконати функціональні вимоги розроблюваного проекту.

2.3 Проектування модулів інтернет-платформи

Для розробки сучасного інтернет-магазину необхідно використовувати модулі та віджети які значно розширять функціональні та не функціональні можливості проекту.

Одним з перших модулів який слід використовувати на кожному веб-сайті є модуль переходу в початок сторінки.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		43

Модулі навігації сприяють зручному використанню сайту будь-якої тематики, дозволяючи швидко переходити на початок сторінки при натисканні на відповідну кнопку. Модуль буде розроблено у вигляді кнопки з знаком стрілки яка дивиться вгору, розміщеної в правому нижньому куту веб-сайту, та буде з'являтися лише коли користувач спуститься до певного моменту на сторінці, тоб-то на відстань одного екрану вниз.

Для кращої взаємодії з користувачами також буде розроблено модуль зворотнього зв'язку, який буде містити форму для введення даних з такими елементами:

- Name – поле у яке користувачу потрібно буде ввести своє ім'я щоб адміністрація могла до нього звертатись під час подальшої розмови;
- Email – поле у яке користувачу потрібно буде ввести email на який надійде відповідь від адміністрації магазину на питання або пропозицію яка буде надіслана;
- Subject – поле у яке потрібно буде ввести тему звернення користувача, наприклад якась помилка або неспівпадіння цін і т.д.;
- Body – текстова зона у яку можна буде у розгорнутому вигляді описати проблему або пропозицію з якою користувач хоче звернутись до адміністрації;
- Verification Code – поле для введення символів з картинки яка буде зображена лівіше від поля вводу, так звана перевірка на людину.

Розробка такого модулю дозволить адміністрації підтримувати зв'язок із користувачами інтернет-платформи, та отримувати повідомлення про можливі помилки або пропозиції які можуть запропонувати користувачі.

Фотогалерея – необхідний елемент будь-якого проекту. Вона дозволяє демонструвати різноманітний візуальний контент: як товари на комерційних сайтах, так і страви з меню ресторанів, готельні номери, дизайн інтер'єру, проведення заходів, роботу команди та багато іншого.

Завдяки якісним фотографіям користувачі проводять на сайті більше часу, покращуються поведінкові фактори та збільшується трафік за рахунок аудиторії, яка здійснює пошук за картинками.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		44

Саме завдяки модулю фотогалереї користувачі зможуть переглянути усі фото конкретного годинника, що представлені на сторінці перегляду товару. Сам модуль буде відкриватись при натисканні на фото на сторінці перегляду товару. Після натискання кнопки фото будуть відкриватись у розмірі у якому їх додали на сторінку без зменшення, а також зправа та зліва екрану будуть з'являтись кнопки для перемикання між іншими фото товару.

Ще одним модулем який необхідно буде розробити є модуль для виведення категорій. Даний модуль буде виводити категорії у вигляді випадаючого списку, заголовками списку будуть головні категорії, при натисканні на які буде розгортатись список дочірніх категорій. При натисканні на дочірні категорії модуль буде перенаправляти користувача на сторінку перегляду категорій де будуть виводитись усі товари певної категорії.

На сторінці перегляду товарів однієї категорії або ж усіх товарів необхідно розробити модуль фільтрації. Він буде розроблений у вигляді форми з такими елементами:

- Category – випадаючий список, де користувач зможе обрати необхідну йому категорію або ж усі категорії;
- Name – текстове поле, де користувачу необхідно ввести повну або часткову назву продукту який його цікавить;
- Brand - текстове поле, де користувачу необхідно ввести повну або часткову назву бренду який його цікавить;
- Sex – випадаючий список, де користувач зможе обрати стать для якої підходять товари;
- Material – текстове поле, де користувачу необхідно ввести повну або часткову назву матеріалу який його цікавить;
- Bracelet – випадаючий список, де користувач зможе обрати тип фіксації годинника(браслет/ремінець);
- Min Price – текстове поле, де користувач зможе ввести мінімальну ціну годинників;

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		45

– Max Price – текстове поле, де користувач зможе ввести максимально допустиму ціну яка йому підходить.

– Search – кнопка з текстом «Знайти» натискаючи яку користувач запускає модуль у роботу.

Коли користувач ввів параметри які його цікавлять і натиснув кнопку «Знайти», модуль повинен вивести товари які підходять під обрані критерії без перезавантаження сторінки.

Також потрібно розробити модуль для публікації відгуків, який буде представлений у вигляді форми, з такими елементами:

– Name – текстове поле, яке вже буде заповнене іменем користувача що авторизувався, так як модуль буде доступний лише для авторизованих користувачів;

– Rating: - поле яке складається з 5 елементів у формі зірки, де користувач зможе обрати скільки зірок він поставить годиннику, від 1 до 5;

– Comment – текстове поле великого розміру де користувач зможе написати свій відгук про конкретну модель годинника;

– Publish – кнопка, натиснувши на яку користувач приведе модуль у дію і відправить свій відгук у базу даних для публікації, після запису у БД відгук одразу ж опублікується на сторінці з товаром.

Також одним з найголовніших модулів є модуль кабінету адміністратора, де повинні бути реалізовані моделі CRUD(Create – Read - Update – Delete) по кожній таблиці БД, що дозволить адміністраторам редагувати увесь контент на сайті.

Отже у цій частині розділу було спроектовано модулі які будуть використовуватись у роботі інтернет-платформи.

2.4 Проектування інтерфейсу користувача

Проектування інтерфейсу складається з багатьох кроків:

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		46

- Проведення дослідження;
- створення користувацьких сценаріїв;
- створення структури інтерфейсу;
- процес прототипування;
- визначення стилістики інтерфейсу;
- створення концепції дизайну;
- кінцеве оформлення всіх сторінок продукту;
- створення анімації;
- підготовка необхідних матеріалів для розробників.

Також проводиться тестування дизайну інтерфейсу, щоб зрозуміти наскільки зручним він буде для користувача.

Веб додатку потрібен не стільки привабливий інтерфейс, скільки вдячні клієнти, які будуть із задоволенням користуватись застосунком. Для цього вже і потрібен грамотно спроектований інтерфейс, для створення якого використовуються перевірені веб-інструменти.

Завдяки цим інструментам є можливість не тільки швидко створювати варіанти інтерфейсу, але і легко вносити в нього правки просто в онлайн-платформі. Axure, InVision, Figma та інші системи дозволяють економити час на розробку і поліпшити логіку інтерфейсу будь-якого веб-ресурсу.

Для оформлення дизайну додатку було обрано три основних кольори: блакитний, білий і чорний. Для початку потрібно спроектувати шаблон, який складається з двох частин, так званих хедера(верхня частина) сайту та футера(нижня частина), адже два цих елемента будуть зображені на кожній сторінці без виключення.

У хедері буде міститись навігація, посилання на соціальні мережі магазину, контакти розробника а також логотип та доступні дії для користувача, такі, як: Список бажань(лише для авторизованих користувачів), Порівняти(лише

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		47

для авторизованих користувачів), Кошик, Авторизація, Вихід(тільки для авторизованих користувачів). І ще одним з найголовніших елементів хедера буде поле для пошуку, скориставшись яким можна буде знайти необхідний годинник по повній або частковій назві. Вигляд хедеру зображено на рисунку



Рисунок 2.16 – Зовнішній вигляд хедера інтернет-платформи

Футер ж міститиме відео необхідні користувачам для полегшення підбору необхідних годинників, які можна буде відкрити і переглянути безпосередньо на сайті, адресу магазину, та безліч полів з інформацією які допоможуть користувачам знайти відповіді на їх запитання. Зовнішній вигляд футера зображено на рисунку 2.17.

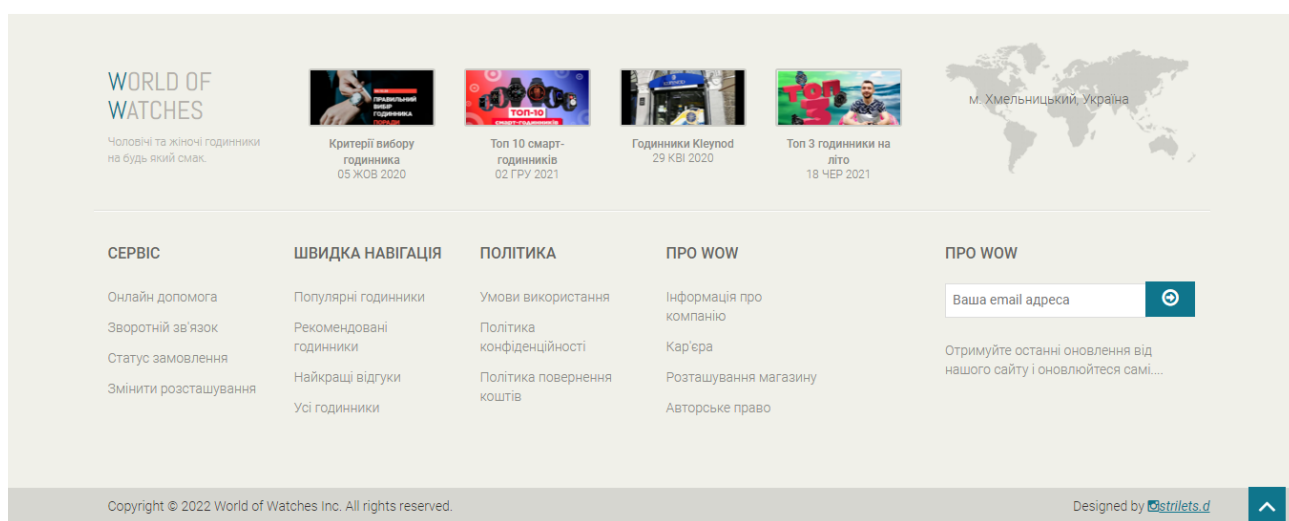


Рисунок 2.17 – Зовнішній вигляд футера інтернет-платформи

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		48

Однією з найголовніших сторінок веб-сайту є головна сторінка, адже переглянувши її можна одразу зробити висновок про якість і функціональність веб-додатку. Насамперед користувачів повинен зустрічати так званий слайдер або іншими словами карусель з фото, які будуть автоматично змінюватись з анімацією та привертати увагу кінцевого користувача. Зовнішній вигляд слайдеру зображено на рисунку 2.18.

Далі користувачів будуть зустрічати хіти, тоб-то годинники які найкраще продаються, і випадаючий список категорій, натиснувши на який користувач перейде до сторінки перегляду товарів конкретної категорії. Віджет категорій буде відображатись у лівій частині сторінки, забираючи не велику частину простору, залишаючи місце для відображення годинників. Годинники будуть відображатись у вигляді так званої сітки, по три елемента у рядку, усього в два рядки аби не перевантажувати головну сторінку зайвим контентом. Зовнішній вигляд виведення годинників зображено на рисунку 2.19.



Рисунок 2.18 – Зовнішній вигляд слайдера на головній сторінці

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		49

вкладки, на одній з яких буде виводитись уся детальна інформація про годинник а на іншій усі відгуки, які залишили користувачі. Також у другій вкладці буде виведена форма для зареєстрованих користувачів, заповнивши цю форму вони зможуть залишити відгук про годинник. Зовнішній вигляд вкладки з відгуками зображено на рисунку 2.22.

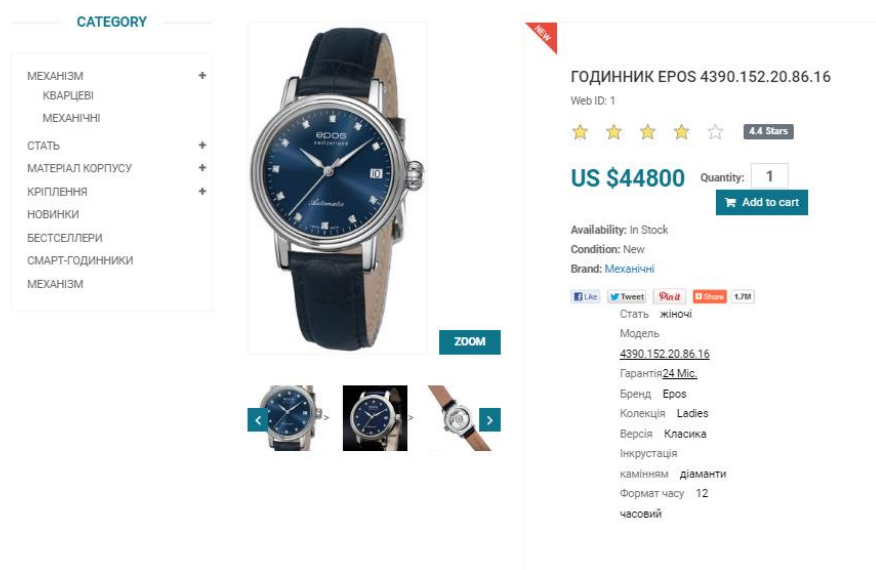


Рисунок 2.20 – Вигляд сторінки перегляду годинника

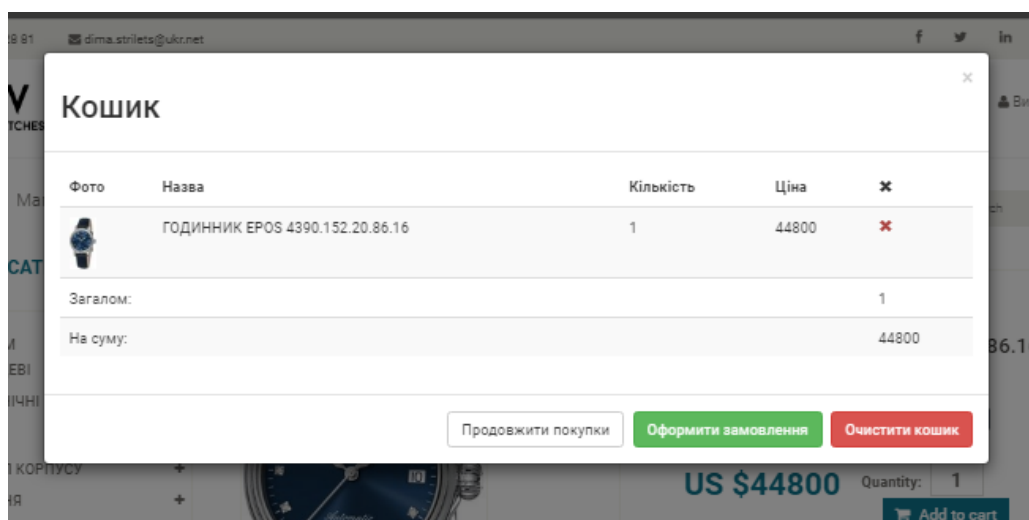


Рисунок 2.21 – Вигляд модального вікна кошику

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		51

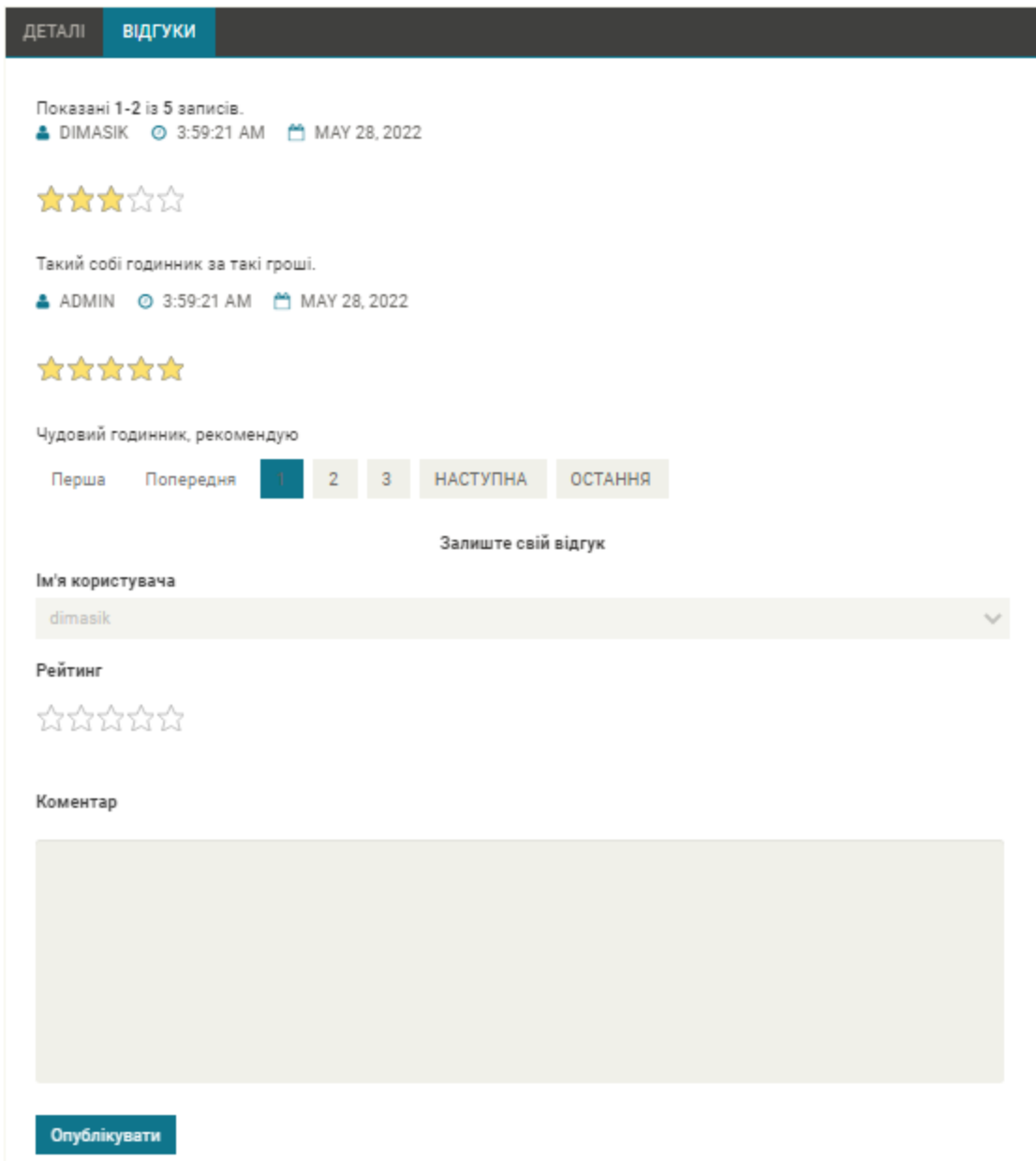


Рисунок 2.22 – Вигляд вкладки з відгуками

Отже у цій частині було спроектовано інтерфейс користувача, який дозволить максимально просто та зручно здійснювати покупки годинників. Спроектований користувацький інтерфейс можна переглянути у додатку В.

2.6 Аналіз та вибір технологій і методів реалізації інтернет-платформи

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		52

Для початку розробки сайту необхідно обрати серверну мову програмування.

На даних момент існують такі серверні мови програмування:

- Ruby;
- PHP;
- Python.

Кожна мова програмування має фреймворк(каркас) який полегшує розробку за допомогою тієї чи іншої мови, та збільшує її можливості в декілька разів.

Програмний фреймворк (англ. software framework) — це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. Відповідно — програмний фреймворк може містити в собі також допоміжні програми, деякі бібліотеки коду, скрипти та загалом все, що полегшує створення та поєднання різних компонентів великого програмного забезпечення чи швидке створення готового і не обов'язково об'ємного програмного продукту. Побудова кінцевого продукту відбувається, зазвичай, на базі єдиного API.

Фреймворки на мові PHP:

– Laravel - безкоштовний, з відкритим кодом PHP-фреймворк, створений Taylor Otwell і призначений для розробки веб-додатків відповідно до шаблону model-view-controller (MVC). Деякі з особливостей Laravel є модульна система упакування з виділенням менеджером залежностей Composer, різні способи для доступу до реляційних баз даних, утиліти, які допомагають в розгортанні додатків і технічного обслуговування, а також його орієнтація на синтаксичний цукор;

– Symfony – відкритий PHP фреймворк, що реалізує концепцію модель вид контролер (MVC) та автоматизує найзагальніші веб-задачі, являє собою широконалаштовну систему пов'язаних класів і призначений для розробки та керування веб-застосунками. Випускається

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		53

під MIT ліцензією. Symfony є вільним програмним забезпеченням. Веб-сайт першої версії symfony-project.com був запущений 18 жовтня 2005 року;

- Yii — це високопродуктивний веб-фреймворк, написаний на PHP, реалізує парадигму модель-вид-контролер. Yii — скорочення від «Yes It Is!».

Фреймворки на мові Ruby:

- Ruby on Rails – це об'єктно-орієнтований програмний каркас (фреймворк) для створення веб-додатків, написаний на мові програмування Ruby. Ruby on Rails надає каркас модель-вид-контролер (Model-View-Controller) для веб-додатків, а також забезпечує їх інтеграцію з веб-сервером і сервером бази даних;

- Camping – це мікрофреймворк для створення веб-додатків написаний на Ruby. Розмір вихідного коду Camping-у менше 4 Кб.

Фреймворки на мові Python:

- Django – це високорівневий відкритий Python фреймворк (програмний каркас) для розробки веб-систем. Названо його було на честь джазмена Джанго Рейнхардта (відповідно до музичних смаків одного зі засновників проекту);

- Pyramid (англ. Pyramid - піраміда) - програмний каркас (фреймворк) для розробки веб-додатків з відкритим вихідним кодом, написаний на мові Python в рамках проекту Pylons;

Проаналізувавши наявні серверні мови та фреймворки, було зроблено вибір у користь мови програмування PHP та фреймворку Yii2.

Переваги використання Yii2 для розробки:

- використовуються стандартні способи рішення задач, що зменшує заплутаність програмного коду;

- скорочується час, що використовується на рутинні задачі, такі як: перевірка та валідація форм, обробка запитів, безпека даних;

- полегшена робота з терміналом, яка реалізується за рахунок дотримання міжнародних стандартів;

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		54

- поліпшення підтримки коду з використанням загальної архітектури та методів;
- наявність надзвичайно активної спільноти розробників, що підтримують фреймворк, розробляють та вводять до нього нові підходи, можливості та технології.

Фреймворк Yii2 має дві версії розгортання: базову версію (Simple Version) та прогресивну (Advanced Version). Версії розгортання Yii2 розрізняються підтримкою різних типів додатків. Базова версія створена для швидкого розгортання простого сайту, а прогресивна версія дозволяє керувати Front-End, Back-End та консольними частинами додатку, повністю розмежовувати їх та чітко розділяти архітектуру сайту.

Отже, в процесі написання розділу було розглянуто та проаналізовано види архітектур додатків та архітектурні шаблони, та визначено, що найбільше для виконання усіх функціональних і не функціональних вимог підійде клієнт-серверна архітектура та архітектурний шаблон MVC.

Окрім того, було проведено аналіз різних типів БД та обрано тип який найбільше підходить до конкретного програмного рішення, а саме реляційна база даних MySQL.

Також було спроектовано модулі які будуть використовуватись в роботі інтернет-платформи та дозволять розширити її функціонал, зручність та пришвидшити її роботу.

Окрім вищезгаданого, було також виконано проектування основних елементів користувацького інтерфейсу які будуть максимально прості та зручні для кінцевого користувача. А також проаналізовано доступні методи та технології які можуть підійти для розробки проекту, та обрано фреймворк який зможе задовольнити усі функціональні вимоги до інтернет-платформи, це Yii2 Framework на базі мови програмування PHP.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		55

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Розробка бази даних

Для розробки бази даних було обрано СКБД MySQL, адже вона ідеально підходить для керування обраним раніше типом бази даних, а саме реляційною БД. Також для зручності буде використано PhpMyAdmin.

PhpMyAdmin — вебдодаток з відкритим кодом, написаний мовою PHP із графічним веб-інтерфейсом для адміністрування бази даних MySQL або MariaDB. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних. Ця програма користується великою популярністю у веб-розробників, оскільки дозволяє керувати базою даних MySQL без вводу SQL запитів через дружній інтерфейс і з будь-якого комп'ютера під'єданого до інтернету без необхідності встановлення додаткового програмного забезпечення. Сторінку входу до phpMyAdmin зображено на рисунку 3.1.

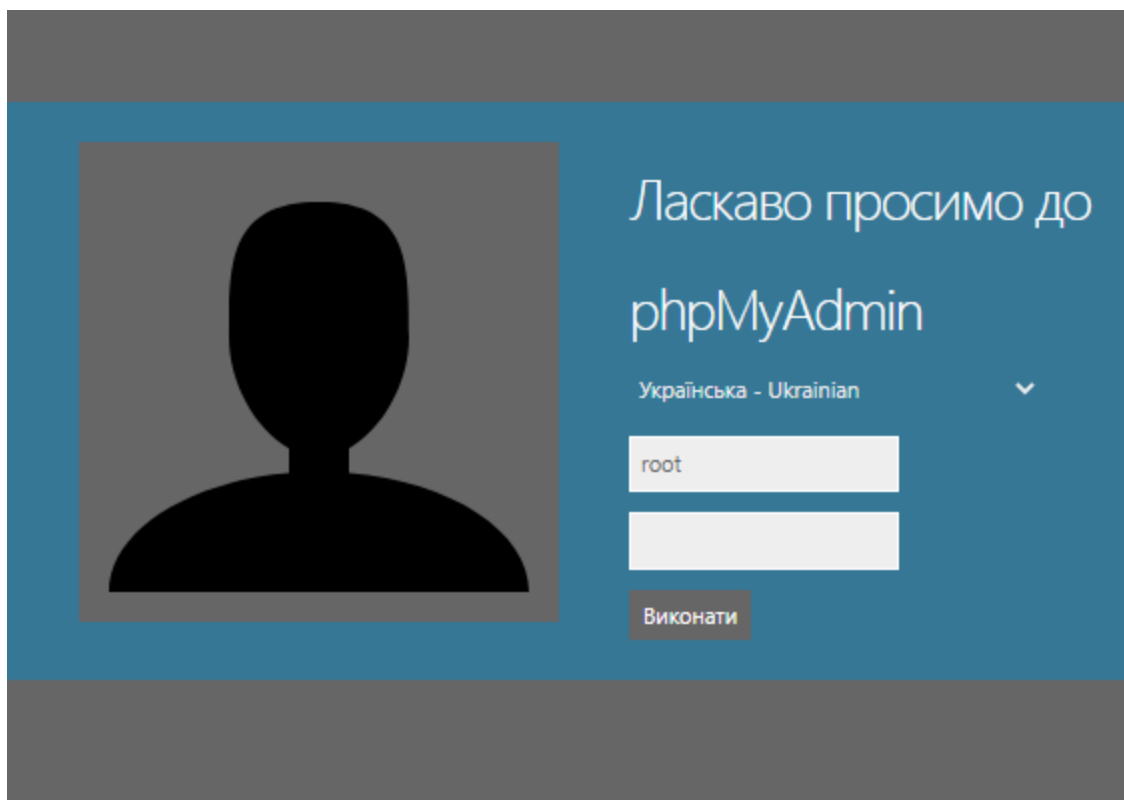


Рисунок 3.1 – Сторінка входу до phpMyAdmin

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		56

Увійти до системи можна за допомогою логіну root та пароллю root. Одразу після входу потрапляємо на сторінку з базами даних, де можемо створити власну, для цього вводимо її назву і обираємо кодування(рисунок 3.2), далі натискаємо кнопку створити і отримуємо створену БД.

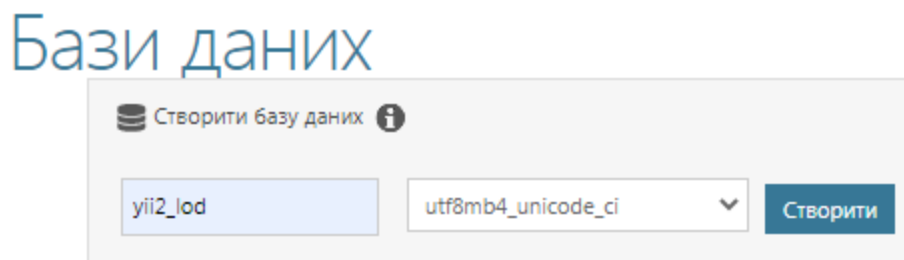


Рисунок 3.2 – Форма для створення нової бази даних

Для створення таблиць вводимо у відповідні поля назву таблиці і число стовбців(рисунок 3.3).

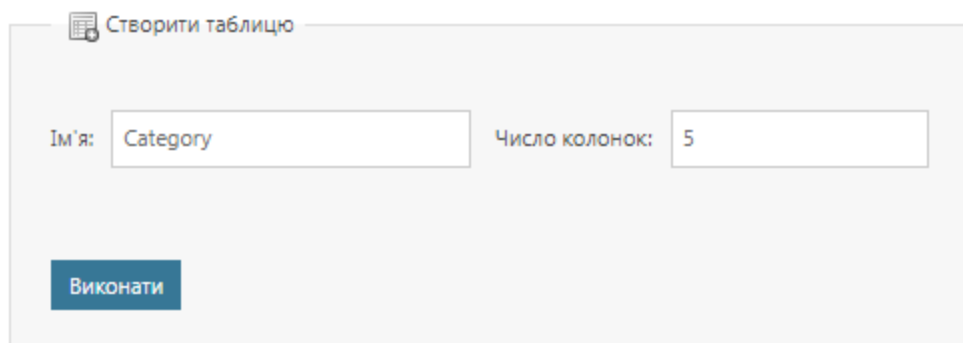


Рисунок 3.3 – Форма для створення нової таблиці

Далі створюємо таблиці згідно спроектованих сутностей у розділі 2.2 Аналіз та вибір типу бази даних , проектування бази даних.

Структура таблиці категорії(рисунок 3.4).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id	int(10)	UNSIGNED	Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	parent_id	int(10)	UNSIGNED	Hi	0		
<input type="checkbox"/>	3	name	varchar(255) <i>utf8_general_ci</i>		Hi	Немає		
<input type="checkbox"/>	4	keywords	varchar(255) <i>utf8_general_ci</i>		Так	NULL		
<input type="checkbox"/>	5	description	varchar(255) <i>utf8_general_ci</i>		Так	NULL		

Рисунок 3.4 – Структура таблиці Category

Структура таблиці користувач(рисунок 3.5).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id	int(10)	UNSIGNED	Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	username	varchar(255) <i>utf8_general_ci</i>		Hi	Немає		
<input type="checkbox"/>	3	password	varchar(255) <i>utf8_general_ci</i>		Hi	Немає		
<input type="checkbox"/>	4	auth_key	varchar(255) <i>utf8_general_ci</i>		Так	NULL		

Рисунок 3.5 – Структура таблиці User

Структура таблиці відгук(рисунок 3.6).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id	int(10)	UNSIGNED	Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	product_id	int(11)	UNSIGNED	Hi	Немає		
<input type="checkbox"/>	3	user_id	int(11)	UNSIGNED	Hi	Немає		
<input type="checkbox"/>	4	rating	int(11)		Hi	Немає		
<input type="checkbox"/>	5	coment	varchar(255) <i>utf8mb4_unicode_ci</i>		Hi	Немає		
<input type="checkbox"/>	6	date	datetime		Hi	CURRENT_TIMESTAMP		

Рисунок 3.6 – Структура таблиці Review

Необхідно встановити зв'язки за ключовими полями product_id та user_id. Поле product_id необхідно з'єднати з полем id таблиці Product, тим самим встановлюючи між таблицями зв'язок багато до одного(багато відгуків – один продукт). А поле user_id необхідно з'єднати з полем id таблиці user встановлюючи зв'язок багато до одного(багато відгуків – один користувач).

Структура таблиці елементи замовлення(рисунок 3.7).

У даній таблиці необхідно встановити зв'язки за ключовими полями order_id та product_id. Поле product_id необхідно з'єднати з полем id таблиці Product, тим самим встановлюючи між таблицями зв'язок багато до одного(багато елементів замовлень – один продукт). А поле order_id необхідно з'єднати з полем id таблиці order встановлюючи зв'язок багато до одного(багато елементів замовлення – одне замовлення).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1 id	int(10)		UNSIGNED	Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2 order_id	int(10)		UNSIGNED	Hi	Немає		
<input type="checkbox"/>	3 product_id	int(10)		UNSIGNED	Hi	Немає		
<input type="checkbox"/>	4 name	varchar(255)	utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	5 price	float			Hi	Немає		
<input type="checkbox"/>	6 qty_item	int(11)			Hi	Немає		
<input type="checkbox"/>	7 sum_item	float			Hi	Немає		

Рисунок 3.7 – Структура таблиці Order Items

Структура таблиці замовлення(рисунок 3.8).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id	int(10)	UNSIGNED	Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	created_at	datetime		Hi	Немає		
<input type="checkbox"/>	3	updated_at	datetime		Hi	Немає		
<input type="checkbox"/>	4	qty	int(10)		Hi	Немає		
<input type="checkbox"/>	5	sum	float		Hi	Немає		
<input type="checkbox"/>	6	status	enum('0', '1')	utf8_general_ci	Hi	0		
<input type="checkbox"/>	7	name	varchar(255)	utf8_general_ci	Hi	Немає		
<input type="checkbox"/>	8	email	varchar(255)	utf8_general_ci	Hi	Немає		
<input type="checkbox"/>	9	phone	varchar(255)	utf8_general_ci	Hi	Немає		
<input type="checkbox"/>	10	address	varchar(255)	utf8_general_ci	Hi	Немає		

Рисунок 3.8 – Структура таблиці Order

Структура таблиці продукт(рисунок 3.9).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id	int(10)	UNSIGNED	Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	category_id	int(10)	UNSIGNED	Hi	Немає		
<input type="checkbox"/>	3	name	varchar(255)	utf8_general_ci	Hi	Немає		
<input type="checkbox"/>	4	content	text	utf8_general_ci	Так	NULL		
<input type="checkbox"/>	5	price	float		Hi	0		
<input type="checkbox"/>	6	keywords	varchar(255)	utf8_general_ci	Так	NULL		
<input type="checkbox"/>	7	description	varchar(255)	utf8_general_ci	Так	NULL		
<input type="checkbox"/>	8	img	varchar(255)	utf8_general_ci	Так	no-image.png		
<input type="checkbox"/>	9	hit	enum('0', '1')	utf8_general_ci	Hi	0		
<input type="checkbox"/>	10	new	enum('0', '1')	utf8_general_ci	Hi	0		
<input type="checkbox"/>	11	sale	enum('0', '1')	utf8_general_ci	Hi	0		
<input type="checkbox"/>	12	brand	varchar(255)	utf8_general_ci	Так	NULL		
<input type="checkbox"/>	13	sex	varchar(255)	utf8_general_ci	Так	NULL		
<input type="checkbox"/>	14	material	varchar(255)	utf8_general_ci	Так	NULL		
<input type="checkbox"/>	15	bracelet	varchar(255)	utf8_general_ci	Так	NULL		

Рисунок 3.9 – Структура таблиці Product

У даній таблиці необхідно встановити зв'язки за ключовим полем category_id. Поле category_id необхідно з'єднати з полем id таблиці Category, тим самим встановлюючи між таблицями зв'язок багато до одного (багато елементів товарів – одна категорія).

Далі для з'єднання бази даних з проектом необхідно створити моделі та додати назву бази даних, пароль та логін у файл конфігурації фреймворку.

Файл налаштувань бази даних знаходиться у папці з проектом \config\db.php:

```
<?php
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=yii2_lod',
    'username' => 'root',
    'password' => 'root',
    'charset' => 'utf8',
];
```

Для створення моделей необхідно перейти у ГП (рисунок 3.10).

При створенні будь-якої програми нам так чи інакше доводиться писати той самий код, який цілком можна охарактеризувати терміном типовий код.

Щоб полегшити завдання, сучасні фреймворки пропонують нам інструменти для генерування такого коду. У фреймворку Yii2 таким інструментом є Gii. Це ніщо інше, як модуль, який ми можемо налаштувати у конфігурації фреймворку.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		61

Welcome to Gii a magical tool that can write code for you

Start the fun with the following code generators:

Model Generator

This generator generates an ActiveRecord class for the specified database table.

Start »

CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

Start »

Controller Generator

This generator helps you to quickly generate a new controller class with one or several controller actions and their corresponding views.

Start »

Form Generator

This generator generates a view script file that displays a form to collect input for the specified model class.

Start »

Module Generator

This generator helps you to generate the skeleton code needed by a Yii module.

Start »

Extension Generator

This generator helps you to generate the files needed by a Yii extension.

Start »

Get More Generators

Рисунок 3.10 – Вигляд модуля GII

У вікні модуля GII обираємо Model Generator та переходимо у меню створення моделей. У вікні, що з'явилося в першому полі вводимо назву таблиці(повинна співпадати з назвою таблиці у БД), у другому назву моделі яка буде реалізовувати цю таблицю, також вказуємо простір де розташовувати нашу модель і клас який буде наслідувати модель(рисунок 3.11).

Active Record забезпечує об'єктно-орієнтований інтерфейс для доступу та маніпулювання даними, що зберігаються у базах даних. Клас Active Record відповідає таблиці в базі даних, об'єкт Active Record відповідає рядку цієї таблиці, а атрибут об'єкта Active Record є значенням окремого стовпця рядка. Замість безпосереднього написання SQL-виразів ви зможете отримувати доступ до атрибутів Active Record та викликати методи Active Record для доступу та маніпулювання даними, що зберігаються у таблицях бази даних.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		62

Model Generator

This generator generates an ActiveRecord class for the specified database table.

Table Name

product

Model Class Name

Product

Namespace

app\models

Base Class

yii\db\ActiveRecord

Database Connection ID

db

Рисунок 3.11 – Сторінка створення моделей

Ось так виглядає створена модель Product:

```
<?php
namespace app\models;
use yii\db\ActiveRecord;
/**
 * This is the model class for table "product".
 *
 * @property string $id
 * @property string $category_id
 * @property string $name
 * @property string $content
 * @property double $price
 * @property string $keywords
 * @property string $description
 * @property string $img
 * @property string $hit
 * @property string $new
 * @property string $sale
 * @property string $brand
 * @property string $sex
 * @property string $material
 * @property string $bracelet
 */
```

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		63

```

class Product extends ActiveRecord{

    public function behaviors()
    {
        return [
            'image' => [
                'class' => 'rico\yii2images\behaviors\ImageBehave',
            ]
        ];
    }
    public function rules()
    {
        return [
            [['category_id', 'name'], 'required'],
            [['category_id'], 'integer'],
            [['content', 'hit', 'new', 'sale'], 'string'],
            [['price'], 'number'],
            [['name', 'keywords', 'description', 'img', 'brand', 'sex',
'material', 'bracelet'], 'string', 'max' => 255],
            [['image'], 'file', 'extensions' => 'png, jpg'],
            [['gallery'], 'file', 'extensions' => 'png, jpg', 'maxFiles'
=> 4],

        ];
    }
    /**
     * @inheritdoc
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID товару',
            'category_id' => 'Категорія',
            'name' => 'Назва',
            'content' => 'Контент',
            'price' => 'Ціна',
            'keywords' => 'Ключові слова',
            'description' => 'Мета-опис',
            'image' => 'Фото',
            'gallery' => 'Галерея',
            'hit' => 'Хіт',
            'new' => 'Новинка',
            'sale' => 'Розпродаж',
        ];
    }
}

```

						ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата			64

```

        'brand' => 'Бренд',
        'sex' => 'Стать',
        'material' => 'Матеріал',
        'bracelet' => 'Браслет/Ремінець',
        'min_price' => 'Мінімальна ціна',
        'max_price' => 'Максимальна ціна'
    ];
}
public static function tableName(){
    return 'product';
}
public function getCategory(){
    return $this->hasOne(Category::className(), ['id' =>
'category_id']);
}
}

```

Метод `getCategory()` реалізує зв'язок між таблицею `Product` та `Category`;
 Решта моделей створено таким ж способом, окрім моделей які не наслідують клас `ActiveRecord`, тобто не є табличками у БД.

3.2 Розробка програмних модулів

Для початку необхідно реалізувати модуль який буде використовуватись на кожній сторінці, а саме модуль кошика.

Спочатку створюємо контролер, у якому прописуємо функції які буде виконувати модуль. Назва цього контролера буде `CartController`.

```

<?php
namespace app\controllers;
use app\models\Product;
use app\models\Cart;
use app\models\Order;
use app\models\OrderItems;
use Yii;
class CartController extends ApplicationController
{

```

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		65


```

        Yii::$app->session->setFlash('success', 'Ваше замовлення
прийнято. Менеджер незабаром зв\'яжеться з вами.');
```

```

        Yii::$app->mailer->compose('order', ['session' =>
$session])
        ->setFrom(['dima.strilets5079@gmail.com' => 'World Of
Watches'])
        ->setTo($order->email)
        ->setSubject('Замовлення')
        ->send();
        $session->remove('cart');
        $session->remove('cart.qty');
        $session->remove('cart.sum');
        return $this->refresh();
    }else{
        Yii::$app->session->setFlash('error', 'Ошибка оформления
заказа');
    }
}
return $this->render('view', compact('session', 'order'));
}

```

У цьому методі ми отримуємо масив Сесій за допомогою команди `Yii::$app->session`, адже саме там зберігається інформація про товари які користувач додав до кошика. Далі йде перевірка на отримання запиту типу `$_POST`, якщо такий запит надходить, це означає що користувач ввів дані і натиснув кнопку оформити замовлення, ми зберігаємо це замовлення у базу даних і відправляємо лист користувачу на вказану ним пошту за допомогою об'єкта `Yii::$app->mailer`. Якщо ж запит не надійшов то просто виконуємо рендер сторінки кошика.

Також одним із найважливіших елементів для роботи даного модуля є його відображення, прописуємо його код у паці `Views\Cart\Cart-modal.php`. Дане відображення буде відкриватись кожен раз коли користувач буде натискати на кнопку кошика або додавати новий товар.

Код файлу `cart-modal.php`:

```
<?php if(!empty($session['cart'])): ?>
```

					ДППЗ.190161.01.12.ПЗ	Арк.
						67
Зм.	Арк	№ докум.	Підпис	Дата		

```

<div class="table-responsive">
  <table class="table table-hover table-striped">
    <thead>
      <tr>
        <th>Фото</th>
        <th>Назва</th>
        <th>Кількість</th>
        <th>Ціна</th>
        <th><span class="glyphicon glyphicon-remove" aria-
hidden="true"></span></th>
      </tr>
    </thead>
    <tbody>
      <?php foreach($session['cart'] as $id => $item):?>
        <tr>
          <td><?= \yii\helpers\Html::img($item['img'], ['alt' =>
$item['name'], 'height' => 50]) ?></td>
          <td><?= $item['name']?></td>
          <td><?= $item['qty']?></td>
          <td><?= $item['price']?></td>
          <td><span data-id="<?= $id?>" class="glyphicon
glyphicon-remove text-danger del-item" aria-hidden="true"></span></td>
        </tr>
      <?php endforeach?>
      <tr>
        <td colspan="4">Загалом: </td>
        <td><?= $session['cart.qty']?></td>
      </tr>
      <tr>
        <td colspan="4">На суму: </td>
        <td><?= $session['cart.sum']?></td>
      </tr>
    </tbody>
  </table>
</div>
<?php else: ?>
  <h3>Кошик порожній</h3>
<?php endif;?>

```

Також у наш шаблон необхідно додати код, який буде викликати це модальне вікно.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		68


```

public function init(){
    parent::init();
    if( $this->tpl === null ){
        $this->tpl = 'menu';
    }
    $this->tpl .= '.php';
}

public function run(){
    // get cache
    if($this->tpl == 'menu.php'){
        $menu = Yii::$app->cache->get('menu');
        if($menu) return $menu;
    }

    $this->data = Category::find()->indexBy('id')->asArray()->all();
    $this->tree = $this->getTree();
    $this->menuHtml = $this->getMenuHtml($this->tree);
    // set cache
    if($this->tpl == 'menu.php'){
        Yii::$app->cache->set('menu', $this->menuHtml, 60);
    }
    return $this->menuHtml;
}

protected function getTree(){
    $tree = [];
    foreach ($this->data as $id=>.&$node) {
        if (!$node['parent_id'])
            $tree[$id] = &$node;
        else
            $this->data[$node['parent_id']]['childs'][$node['id']] =
&$node;
    }
    return $tree;
}

protected function getMenuHtml($tree, $tab = ''){
    $str = '';
    foreach ($tree as $category) {
        $str .= $this->catToTemplate($category, $tab);
    }
}

```

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		70

```

    }
    return $str;
}

protected function catToTemplate($category, $tab){
    ob_start();
    include __DIR__ . '/menu_tpl/' . $this->tpl;
    return ob_get_clean();
}
}

```

Код шаблону menu.php:

```

<li>
    <a href="<?=\yii\helpers\Url::to(['category/view', 'id' =>
$category['id']]) ?>"
        <?=$category['name']?>
        <?php if( isset($category['childs']) ): ?>
            <span class="badge pull-right"><i class="fa fa-
plus"></i></span>
            <?php endif;?>
        </a>
        <?php if( isset($category['childs']) ): ?>
            <ul>
                <?=$this->getMenuHtml($category['childs'])?>
            </ul>
        <?php endif;?>
</li>

```

У цьому файлі ми отримуємо об'єкт який містить інформацію про 1 рядок таблиці Category, та виводимо його так як нам необхідно.

Викликати даний модуль можна прописавши всього 1 рядок коду на будь якій сторінці:

```
<?=\app\components\MenuWidget::widget(['tpl' => 'menu'])?>
```

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		71

Наступним модулем який необхідно розробити є модуль відгуків. Для початку необхідно створити форму яка буде генеруватись на сторінці для перегляду товару. Для цього заходимо у папку Product і створюємо файл `_form.php`. Описуємо необхідні нам поля та віджети(віджет рейтингу):

```
<?php
use app\models\User;
use kartik\rating\StarRating;
use yii\helpers\ArrayHelper;
use yii\helpers\Html;
use yii\widgets\ActiveForm;
/* @var $this yii\web\View */
/* @var $model app\models\Review */
/* @var $form yii\widgets\ActiveForm */
?>

<?php $form = ActiveForm::begin([
    'action' => ['product/create', 'id' => $model->product_id],
    'method' => 'post',
]); ?>

<?php
$users = User::find()->where(['id'=> Yii::$app->user->identity['id']])-
>all();
$params = [
    'prompt' => 'Користувач',
    'class' => 'username',
    'disabled' => 'disabled'
];
echo $form->field($model, 'user_id')-
>dropDownList(ArrayHelper::map($users, 'id', 'username'), $params) ?>

<?= $form->field($model, 'rating')->widget(StarRating::classname(), [
    'pluginOptions' => [
        'step' => 1,
        'min' => 1,
        'max' => 5,
        'showClear' => false,
        'showCaption' => false,
    ]
]); ?>

<?= $form->field($model, 'product_id')->hiddenInput()->label(false) ?>
```

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		72

```

<?= $form->field($model, 'coment')->textarea(['rows' => '6']) ?>
<div class="form-group">
    <?= Html::submitButton('Опублікувати', ['class' => 'btn btn-success'])
?>
</div>
<?php ActiveForm::end(); ?>
Також необхідно створити шаблон для виводу відгуків:
<?php

use app\models\User;
use kartik\rating\StarRating;
use yii\helpers\Html;

$user = User::find()->where(['id' => $model->user_id])->one();
$formatter = \Yii::$app->formatter;
echo '<div class="col-sm-12">
    <ul>
        <li><a href=""><i class="fa fa-user"></i>' . $user-
>username . '</a></li>
        <li><a href=""><i class="fa fa-clock-o"></i>' .
$formatter->asTime($model->date) . '</a></li>
        <li><a href=""><i class="fa fa-calendar-o"></i>' .
$formatter->asDate($model->date) . '</a></li>
    </ul>
    ' . StarRating::widget([
        'name' => 'rating_2',
        'value' => $model->rating,
        'disabled' => true,
        'pluginOptions' => [
            'readonly' => true,
            'showClear' => false,
            'showCaption' => false,
        ],
    ]) . '<p>' . $model->coment . '</p></div>
';

```

Далі нам необхідно прописати функції які будуть відповідати за функціонал даного модуля. Так, як модуль буде використовуватись на сторінці перегляду товару, його функції потрібно описати у контролері ProductController.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		73

```

<?php
namespace app\controllers;
use app\models\Category;
use app\models\Product;
use app\models\ProductSearch;
use app\models\Review;
use app\models\ReviewSearch;
use DateTime;
use Yii;
use yii\data\ActiveDataProvider;
use yii\data\Pagination;

class ProductController extends AppController
{
    public function actionView($id)
    {
        $model = new Review();
        $reviews = Review::find()->where(['product_id' => $id])->all();
        $suma = 0;
        $rev_count = 0;
        foreach ($reviews as $review) {
            $suma += $review->rating;
            $rev_count += 1;
        }
        if ($rev_count === 0) {
            $rating = 0;
        } else {
            $rating = $suma / $rev_count;
        }
        $product = Product::findOne($id);
        if (empty($product))
            throw new \yii\web\HttpException(404, 'Такого товару немає');
        $categories = Category::find()->where(['<>', 'parent_id', 0])-
>all();

        $query = Review::find()->where(['product_id' => $id]);
        $dataProvider = new ActiveDataProvider([
            'query' => $query,
            'pagination' => [
                'pageSize' => '2'
            ]
        ]);
    }
}

```

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		74

```

        $hits = Product::find()->andFilterWhere(['<>', 'id', $id])-
>andFilterWhere(['hit' => '1'])->limit(6)->all();
        $this->setMeta('wow | ' . $product->name, $product->keywords,
$product->description);
        $model->user_id = Yii::$app->user->identity['id'];
        $model->product_id = $id;
        $pages = new Pagination(['totalCount' => $query->count(),
'pageSize' => 2, 'forcePageParam' => false, 'pageSizeParam' => false]);
        $review = $query->offset($pages->offset)->limit($pages->limit)-
>all();

        return $this->render('view', [
            'dataProvider' => $dataProvider,
            'product' => $product,
            'hits' => $hits,
            'model' => $model,
            'rating' => $rating,
            'rev_count' => $rev_count
        ]);
    }
    public function actionCreate($id)
    {
        $model = new Review();
        $model->load(Yii::$app->request->post());
        $model->product_id = $id;
        $model->user_id = Yii::$app->user->identity['id'];
        $model->save();
        return $this->redirect(['view', 'id' => $id]);
    }
}

```

Функція `actionCreate` буде викликатись при переході на сторінку `product/create`, вона буде викликатись коли користувач натисне кнопку «Опублікувати» на формі з модулем.

Функція `actionView` викликається коли користувач натискає на конкретний товар і переходить на сторінку для його перегляду `product/view/id`.

У цій функції ми отримуємо дані які нам потрібні, тобто інформацію про товар, схожі товари та усі відгуки до даного товару.

Викликати модуль можна за допомогою маленького коду:

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		75

```

<?php Pjax::begin(); ?>

<?= ListView::widget([
    'dataProvider' => $dataProvider,
    'itemView' => '_list',
    'pager' => [
        'firstPageLabel' => 'Перша',
        'lastPageLabel' => 'Остання',
        'nextPageLabel' => 'Наступна',
        'prevPageLabel' => 'Попередня',
        'maxButtonCount' => 5,
    ],
]) ?>
<?php if(!Yii::$app->user->isGuest): ?>
<p class="text-center"><b>Залиште свій
відгук</b></p>

<?php echo $this->render('_form', ['model'
=> $model]); ?>

<?php endif; ?>
<?php Pjax::end(); ?>

```

Код(лістинг) платформи представлено у додатку В.

3.3 Керівництво користувача

Після розробки проекту необхідно здійснюється опис покрокового користування веб-додатком для різних категорій користувачів: адміністратора, зареєстрованого користувача та гостя.

Гість потрапляючи на головну сторінку інтернет платформи може переглядати список категорій, та список хітів(товари які найкраще продаються). У гостя є можливість обрати пену категорію та перейти на сторінку перегляду товарів обраної категорії. На цій сторінці також доступна система фільтрації користуючись якою гість може підібрати для себе годинник який йому найбільше підходить за вказаними параметрами. Обравши потрібний годинник гість може додати його у кошик або перейти до сторінки перегляду

					ДППЗ.190161.01.12.ПЗ	Арк.
						76
Зм.	Арк	№ докум.	Підпис	Дата		

товару. На сторінці перегляду детальної інформації гість може користуватись галереєю для перегляду фото, переглядати відгуки про товар але не може їх залишати. Також гостю доступна можливість переглянути свій кошик та стрічку новин, а також можливість авторизації або реєстрації. Для реєстрації необхідно ввести логін та пароль, зареєструвавшись гість може авторизуватись та отримати повний спектр можливостей інтернет-платформи. Переглянувши свій кошик гість може перейти до оформлення замовлення. На сторінці підтвердження замовлення йому необхідно ввести своє Ім'я, електронну пошту та адресу, після заповнення форми гість може відправити підтвердження замовлення і одразу ж отримає листа з квитанцією замовлення.

Авторизувавшись функціональні можливості користувача одразу збільшуються. Для авторизованого користувача відкривається можливість додати товар у список порівнянь та список бажань, а також можливість залишити відгук на сторінці перегляду детальної інформації про товар, та можливість виставити товару рейтинг від 1 до 5.

Ще одним користувачем якому будуть доступні усі функції є адміністратор. Окрім усіх можливостей звичайного користувача у адміністратора є можливість перейти до кабінету адміністратора прописавши у стрічці браузера /admin. На сторінці адміністратора одразу виводиться список замовлень які оформляли користувачі, адміністратор може переглядати, редагувати та видаляти замовлення, а також міняти їх статус. Далі у адміністратора є можливість створювати нові категорії а також переглядати та редагувати усі наявні. Також у адміністратора доступна можливість додавати нові товари, з наявним на сторінці редактором опису, який можна повністю кастомізувати за бажанням, а також є можливість додавання фото до опису, таким чином опис можна оформити як повноцінну статтю.

3.4 Технічні характеристики веб-додатка

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		77

Умови експлуатації повинні відповідати технічним нормам використання ПК. Також для експлуатації додатку необхідно мати на пристрої будь який браузер та доступ до мережі інтернет, тобто вимогами для експлуатації є вимоги для використання браузера.

Вимоги до складу і параметрів технічних засобів.

Система повинна задовольняти таким вимогам для різних операційних систем:

Windows:

– Windows 7, Windows 8, Windows 8.1, Windows 10 або більш новішої версії.

– Процесор Intel Pentium 4 або більш новішої версії з підтримкою SSE3; наявність мережевої плати або wifi адаптера.

macOS:

– OS X El Capitan (10.11) або більш новішої версії.

Linux:

– Ubuntu 18.04 (64-розрядна версія) або більш новіша версія, Debian 10 або більш новіша версія, openSUSE 15.2 або більш новіша версія, Fedora Linux 32 або більш новіша версія.

– Процесор Intel Pentium 4 або більш новіша версія з підтримкою SSE3.

Android:

– Android 6.0 Marshmallow або більш новіша версія.

Також для зображення оптимальних характеристик серверної та клієнтської частини було розроблено діаграму Deployment(рисунок 3.1»2).

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		78

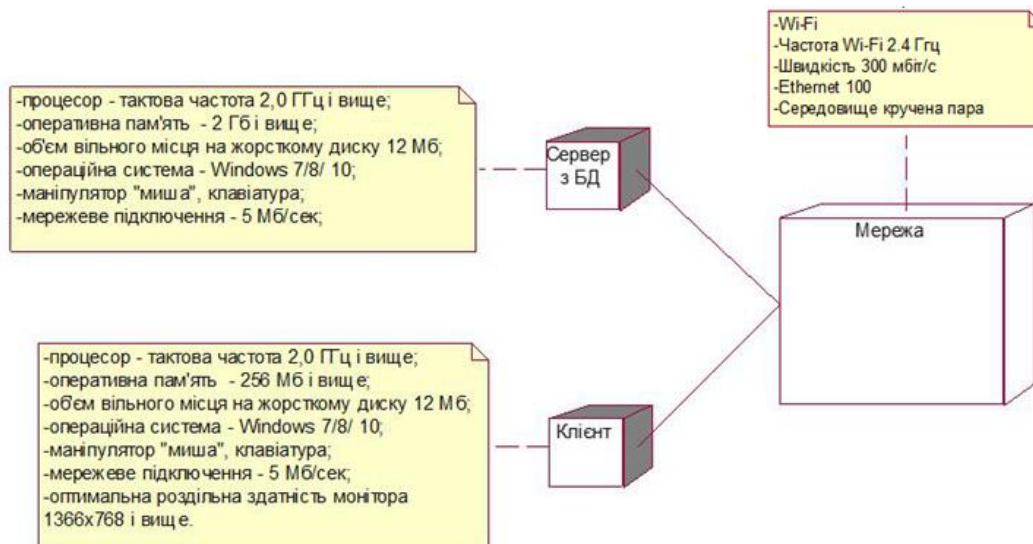


Рисунок 3.12 – Діаграма Deployment

3.5 Завантаження веб-додатка на хостинг

Для завантаження інтернет-платформи на хостинг нам необхідно обрати хостинг, у нашому випадку це український безкоштовний хостинг, час користування яким можливе лише 7 днів з моменту реєстрації. Переходимо на сайт хостингу, обираємо свій домен та пакет послуг для хостингу.(рисунок 3.13).

worldofwatches.zzz.com.ua

Хостинг VPS Виділений сервер

Виберіть пакет послуг (порівняти пропозиції):

Free **Pro** **VIP**

Ціна хостингу: безкоштовно

Ціна домену: безкоштовно

Рисунок 3.13 – Вибір домену та пакету послуг на сайті хостингу

Після того як ми обрали необхідні нам параметри, нам необхідно зареєструватись. Для успішної реєстрації необхідно отримати електронний лист з кодом на пошту(рисунок 3.14).

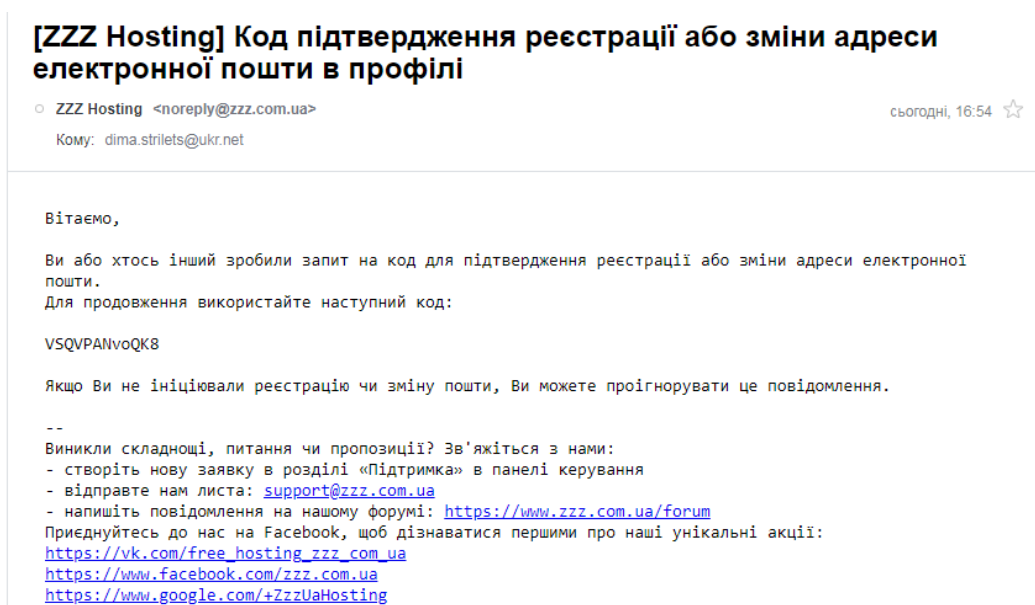


Рисунок 3.14 – Електронний лист про підтвердження реєстрації

Після реєстрації ми отримуємо повідомлення, що наш акаунт успішно створено(рисунок 3.15). Далі нам необхідно створити користувача для MySQL та для FTP сервера, за допомогою яких ми зможемо імпортувати нашу базу даних та завантажити і файли проекту на хостинг.(рисунок 3.16)

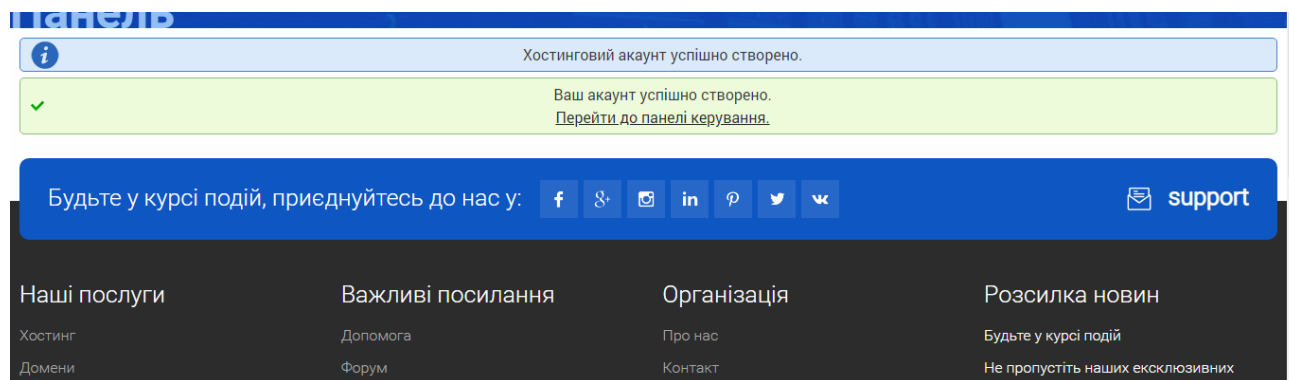


Рисунок 3.15 – Повідомлення про успішну реєстрацію

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		80

4 ТЕСТУВАННЯ ВЕБ-ДОДАТКА

4.1 Аналіз методів тестування веб-додатка

Для повноцінного функціонування веб-сайту без помилок його необхідно протестувати різними видами тестування, кожен з яких відповідає за різну частину інтернет-платформи.

Основні види тестування сайтів:

- модульне тестування;
- функціональне тестування;
- тестування юзабіліті (перевірка зручності користування);
- тестування сумісності (конфігураційне тестування);
- тест на продуктивність;
- перевірка безпеки.

Модульне тестування — це метод тестування програмного забезпечення, який полягає в окремому тестуванні кожного модуля коду програми. Модулем називають найменшу частину програми, яка може бути протестованою. У процедурному програмуванні модулем вважають окрему функцію або процедуру. В об'єктно-орієнтованому програмуванні — метод.

Функціональне тестування сайту – це один з найбільш масштабних видів робіт, з якого все починається. Основна мета – переконатися в коректності роботи основних функцій сайту. Чек-лист на цьому етапі виглядає наступним чином:

- перевірка посилань (внутрішніх, вихідних, ізольованих і т. д.);
- перевірка форм – в подальшому дозволить взаємодіяти з користувачами і отримувати від них дані;
- тест файлів cookie – необхідний для підтримки сеансів з авторизацією;
- валідація HTML/CSS – перевірка на відсутність серйозних синтаксичних помилок і доступу ресурсу для різних пошукових систем;

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		83

– перевірка бази даних – тут потрібно простежити за правильністю виконання запитів, а також вилучення та оновлення даних.

Крім цього види функціонального тестування можна розбити на такі різновиди:

– ad-hock перевірка – не припускає попередньої підготовки. Цей метод допомагає швидко ознайомитися з системою, а також виявити основні несправності;

– негативний тестінг – виявляє невірні дані і значення, які можуть привести до нестабільної роботи системи;

– проведення еквівалентних тестів – група тестів, спрямованих на виявлення конкретної помилки;

– дослідницька перевірка (exploratory testing) – методика, що дозволяє одночасно розробляти і виконувати тести, а заодно ретельно вивчити готовий продукт.

Тестування веб-ресурсів: юзабіліті.

Перевірка зручності використання допомагає оцінити сприйняття дизайну сайту і його функціонал користувачем і те, наскільки довго він готовий не закривати вкладку з ним. Сайт повинен бути послідовним і чітко структурованим. На цьому етапі критерії перевірки включають такі пункти:

– навігаційне тестування – перевірка доступу до меню, сторінок, блоків, полів і кнопок. Важливо, щоб структура була логічною і зрозумілою;

– перевірка контенту – перевіряється граматика і орфографія, розміщення заголовків, а також відповідність розмірів зображень;

– комфорт користування – перевірка структури і наявності зайвих елементів.

Типи тестування сайту на сумісність.

На цьому етапі перевіряється адаптивність сайту. Це так зване нефункціональне тестування, яке дозволяє побачити, наскільки коректно

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		84

відображається ресурс на різних розширеннях, браузерах і пристроях. розрізняють:

– кросплатформений тестінг – перевіряється робота сайту на різних операційних системах і їх версіях;

– перевірка на кросбраузерність – сайт повинен однаково добре функціонувати в усіх відомих браузерах (Google, Яндекс, Firefox, Internet Explorer і т. д.). Він також повинен правильно відображатися, на що впливає якісна верстка;

– тестування на мобільних пристроях – тут багато помилок залишаються непоміченими. Фахівці застосовують різні види тестування сайтів для їх перевірки на адаптивність.

Тест на продуктивність

Важливий етап перевірки, який допомагає оцінити стабільність роботи ресурсу в критичних ситуаціях. Включає наступні методи:

– тестування навантаження – перевіряється здатність одночасної обробки великої кількості призначених для користувача запитів. Проводиться для того, щоб визначити пропускну здатність ресурсу або окремих сторінок;

– стрес-тестування – перевірка стійкості системи в екстремальних умовах. Допомагає уникнути поломок під час сильних навантажень, а також визначити їх допустимий рівень;

– перевірка на швидкість з'єднання – перевіряється час відгуку сайту.

Слід зазначити, що перевірка може проводитися вручну або в автоматичному режимі. Автоматизація передбачає використання різних програм і сервісів, які суттєво економлять час, що особливо важливо для перевірки масштабних проектів. Але не можна недооцінювати ручний спосіб, який допомагає оцінити зручність системи і дизайн з точки зору користувача. В результаті ми отримаємо:

– стабільний, безпечний і надійно захищений ресурс;

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		85

– зручний і функціональний продукт, який буде залучати нових користувачів;

– економію ресурсів і часу на підтримку роботи сайту.

Для тестування інтернет платформи було обрано модульне та функціональне тестування.

4.2 Тестування веб-додатка

Починати тестування модулів веб-сайту необхідно з створення модульних тестів. Сценарії модульних тестів зображені у таблиці 4.1.

Таблиця 4.1 – Тестові сценарії для модульного тестування

Ідентифікатор	Модуль	Вхідні дані	Очікуваний результат
1	2	3	4
U-M-1	Авторизація	Пошук користувача по Id	Логін користувача admin
U-M-2	Авторизація	Пошук користувача по access_token	Логін користувача admin
U-M-3	Авторизація	Пошук користувача по username	Логін користувача e admin
U-M-4	Авторизація	Введення логіну та паролю admin	Повернення значення true
C-M-1	Кошик	Натискання кнопки «додати у кошик»	З'являється модальне вікно кошика
C-M-2	Кошик	Натискання кнопки «продовжити покупки»	Модальне вікно зникає
C-M-3	Кошик	Натискання кнопки «очистити кошик»	Модальне вікно не зникає, замість набору товарів з'являється напис «Кошик порожній»

Кінець таблиці 4.1

1	2	3	4
C-M-4	Кошик	Натискання кнопки «Оформити замовлення»	Перехід на сторінку оформлення замовлення
R-M-1	Відгуки	Залишити поле з рейтингом без вибору кількості зірок	Поява повідомлення про те, що поле необхідно заповнити.
R-M-2	Відгуки	Спроба не авторизованого користувача залишити відгук	Форма для публікації відгуків не відображається
R-M-3	Відгуки	Заповнення усіх полів(рейтинг, коментар)	Відгук миттєво публікується на сторінці товару
S-M-1	Пошук	Назва товару, який потрібно знайти	Товари які містять шуканий текст в назві
C-M-1	Категорії	Натискання на якусь категорію	Перехід на сторінку де виведені товари цієї категорії

За створеними сценаріями були розроблені unit-тести. Нижче приведено код unit-тестів для ідентифікаторів U-M-1-4:

```
<?php
namespace tests\models;
use app\models\User;
class UserTest extends \Codeception\Test\Unit
{
    public function testFindUserById()
    {
        expect_that($user = User::findIdentity(100));
        expect($user->username)->equals('admin');

        expect_not(User::findIdentity(999));
    }
}
```

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		87

```

public function testFindUserByAccessToken()
{
    expect_that($user = User::findIdentityByAccessToken('100-token'));
    expect($user->username)->equals('admin');

    expect_not(User::findIdentityByAccessToken('non-existing'));
}
public function testFindUserByUsername()
{
    expect_that($user = User::findByUsername('admin'));
    expect_not(User::findByUsername('not-admin'));
}
/**
 * @depends testFindUserByUsername
 */
public function testValidateUser($user)
{
    $user = User::findByUsername('admin');
    expect_that($user->validateAuthKey('test100key'));
    expect_not($user->validateAuthKey('test102key'));

    expect_that($user->validatePassword('admin'));
    expect_not($user->validatePassword('123456'));
}
}

```

Результат виконання unit-тестів для ідентифікаторів U-M-1-4 можна побачити на рисунку 4.1.

```

Дима@ДИМА-ПК c:\OpenServer\domains\shop
$ php vendor/phpunit/phpunit/phpunit tests/usertest --bootstrap tests/_bootstrap.php
PHPUnit 6.5.13 by Sebastian Bergmann and contributors.

....                                                                4 / 4 (100%)

Time: 177 ms, Memory: 4.00MB

OK (4 tests, 4 assertions)

```

Рисунок 4.1 – результат виконання unit-теста

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		88

Для функціонального тестування додатку використовувався мобільний пристрій Xiaomi Mi A1 з операційною системою Android 9 Pie, а також ПК на операційній системі Windows 10, та ноутбук Asus K53U на операційній системі Windows 7.

Опис сценаріїв функціонального тестування інтернет платформи наведено у таблиці 4.2.

Таблиця 4.2 – Тестові сценарії основного функціоналу веб-додатку

Іденти-фікатор	Функціональна вимога	Вихідні дані	Очікуваний результат
1	2	3	4
CFT-01	Користувач авторизувався	Користувач вводить логін і пароль і натискає кнопку «увійти»	Користувач авторизується і потрапляє на головну сторінку
CFT-02	Користувач зареєструвався	Користувач вводить логін та пароль та натискає кнопку «зареєструватись»	Користувача потрапляє на сторінку авторизації
CFT-03	Відкриття кошику	Користувач натискає кнопку «Кошик» у панелі навігації	Виводиться модальне вікно з вмістом кошика
CFT-04	Очищення кошику	Користувач натискає кнопку «Очистити кошик» у модальному вікні кошику	Користувач бачить надпис «Кошик порожній» у модальному вікні кошика
CFT-05	Видалення товару з кошику	Користувач натискає на хрестик навпроти назви та фото товару у кошику	Товар видаляється з кошика
CFT-06	Оформлення замовлення	Користувач натискає кнопку «Оформити замовлення» у модальному вікні кошика	Користувач бачить форму для оформлення замовлення
CFT-07	Замовлення оформлене	Користувач вводить дані у форму і натискає кнопку «Оформити»	Користувач бачить сповіщення про те, що скоро з ним зв'яжуться менеджери
CFT-08	Отримання листа на електронну пошту	Користувач оформлює замовлення	Користувачу на вказану електронну пошту приходить лист з квитанцією замовлення
CFT-9	Перегляд детальної інформації про годинник	Користувач натискає на назву товару	Користувач бачить фото та детальну інформацію про товар
CFT-10	Перегляд галереї	Користувач натискає на фото товару	Користувач бачить збільшені фото товару і кнопки для зміни фото

Кінець таблиці 4.2

1	2	3	4
CFT-11	Публікація відгука	Користувач заповнює форму для відгуків і натискає кнопку «Опублікувати»	Користувач бачить свій відгук опублікований під обраним товаром
CFT-12	Додавання товару у список бажань	Користувач натискає кнопку «Додати у список бажань»	Користувачу видається повідомлення про те, що товар успішно доданий у список бажань
CFT-13	Товари однієї категорії	Користувач натискає на кнопку будь якої з категорії	Користувач бачить усі товари обраної категорії
CFT-14	Фільтрація товарів	Користувач вводить необхідні параметри і натискає кнопку «Знайти»	Користувач бачить відфільтровані товари за введеними параметрами
CFT-15	Вихід з аккаунту	Користувач натискає кнопку «Вихід»	Користувач переходить на головну сторінку як гість
CFT-16	Пошук	Користувач вводить назву або частину назви товару який необхідно знайти	Користувач бачить товари у назві яких міститься текст введений у поле пошуку

4.3 Аналіз результатів тестування веб-додатка

Результати модульного тестування веб-додатку наведено у таблиці 4.3.

Таблиця 4.3 – Результати модульного тестування інтернет-платформи

Ідентифікатор	Модуль	Вхідні дані	Очікуваний результат	Результат
1	2	3	4	5
U-M-1	Авторизація	Пошук користувача по Id	Логін користувача admin	Правильно
U-M-2	Авторизація	Пошук користувача по access_token	Логін користувача admin	Правильно
U-M-3	Авторизація	Пошук користувача по username	Логін користувача є admin	Правильно

Кінець таблиці 4.3

1	2	3	4	5
U-M-4	Авторизація	Введення логіну та пароллю admin	Повернення значення true	Правильно
C-M-1	Кошик	Натискання кнопки «додати у кошик»	З'являється модальне вікно кошика	Правильно
C-M-2	Кошик	Натискання кнопки «продовжити покупки»	Модальне вікно зникає	Правильно
C-M-3	Кошик	Натискання кнопки «очистити кошик»	Модальне вікно не зникає, замість набору товарів з'являється напис «Кошик порожній»	Правильно
C-M-4	Кошик	Натискання кнопки «Оформити замовлення»	Перехід на сторінку оформлення замовлення	Правильно
R-M-1	Відгуки	Залишити поле з рейтингом без вибору кількості зірок	Поява повідомлення про те, що поле необхідно заповнити.	Правильно
R-M-2	Відгуки	Спроба не авторизованого користувача залишити відгук	Форма для публікації відгуків не відображається	Правильно
R-M-3	Відгуки	Заповнення усіх полів(рейтинг, коментар)	Відгук миттєво публікується на сторінці товару	Правильно
S-M-1	Пошук	Назва товару, який потрібно знайти	Товари які містять шуканий текст в назві	Правильно
C-M-1	Категорії	Натискання на якусь категорію	Перехід на сторінку де виведені товари цієї категорії	Правильно

Як можна побачити у таблиці результатів, модульне тестування пройшло успішно без помилок, це означає, що усі модулі системи працюють справно та без перебоїв.

Далі було проведено функціональні тести на різних видах пристроїв. Отримані результати відображені у таблиці 4.4

Таблиця 4.4 – Результат функціонального тестування інтернет-платформи

Ідентифікатор тесту	Тип пристрою на якому проводилось тестування		
	ПК(1920x1080)	Ноутбук Asus k53u(1366x720)	Телефон Xiaomi Mi A1(1920x1080)
1	2	3	4
CFT-1	+	+	+
CFT-2	+	+	+
CFT-3	+	+	+
CFT-4	+	+	+
CFT-5	+	+	+
CFT-6	+	+	+
CFT-7	+	+	+
CFT-8	+	+	+
CFT-9	+	+	+
CFT-10	+	+	+
CFT-11	+	+	+
CFT-12	+	+	+
CFT-13	+	+	+
CFT-14	+	+	+
CFT-15	+	+	+
CFT-16	+	+	+

Отримавши результати тестування можемо зробити висновок, що розроблений проект є оптимізованим під різні типи пристроїв. Функціонал веб-сайту справно працює на будь якому пристрої з будь якою діагоналлю.

Уданому розділі було розглянуто та проаналізовано різні типи тестування, також було обрано необхідні типи тестувань для інтернет-платформи. За результатами тестування виявлено, що розроблена платформа відповідає всім вимогам описаним у технічному завданні.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		93

ВИСНОВКИ

У процесі розробки дипломного проекту було проведено аналіз предметної області, розглянуто переваги та недоліки інтернет-магазинів. Також розглянуто схожі програмні рішення та виявлено їх слабкі та сильні сторони. Спираючись на досліджену предметну область та виявлені переваги і недоліки схожих рішень було визначено вимоги до інтернет-платформи.

Наступним етапом у розробці дипломного проекту було визначення архітектури веб-сайту. Розглянувши усі можливі типи архітектур та архітектурні шаблони, було прийнято рішення про використання клієнт-серверної архітектури та архітектурного шаблону MVC. На цьому ж етапі було розглянуто та проаналізовано типи баз даних та вибрано один тип який ідеально підійде для розробки інтернет-платформи – реляційна база даних MySQL. Також було спроектовано модулі які будуть використовуватись у роботі веб-сайту та покращать його функціональні можливості, і спроектовано інтерфейс користувача опираючись на зручність та простоту інтерфейсу для комфорту кінцевого користувача. І останнім кроком на даному етапі став вибір технологій та методів розробки. Після розгляду безлічі серверних мов та фреймворків, було зроблено вибір у користь мови програмування PHP та фреймворку Yii2.

Наступним етапом була розробка бази даних та програмних модулів. Також було описано керівництво для різних типів користувачів, а також визначено характеристики клієнтської та серверної частин а також мережі. І останнім кроком у реалізації інтернет-платформи стало завантаження проекту на хостинг та імпортування бази даних.

Після реалізації проекту було проведено різні види тестувань, які показують що система працює вірно та без помилок. І найголовніше інтернет-платформа працює на будь якому типі пристроїв

В результаті виконання дипломного проекту було розроблено інтернет-платформу для продажу годинників «The World of watches» яка допоможе

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		94

користувачам підібрати та порекомендувати годинник який зацікавить користувача, та надасть змогу зручно та без складнощів здійснити покупку обраного годинника, отримуючи при цьому електронного листа з квитанцією здійсненого замовлення.

Таким чином усі завдання та вимоги поставлені в процесі розробки дипломного проекту були успішно виконані та реалізовані з максимальним комфортом для користувача. Також закріплено отриманні практичні та теоретичні знання, які допоможуть в майбутній професійній діяльності.

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		95

10. Програмні модулі для веб-сайтів [Електронний ресурс] / Rbs Webmarket – Режим доступу до ресурсу: <http://rbs-webmarket.com/making-site/web-programming/programme-modules-site.html>

11. Офіційна документація класу Active Record [Електронний ресурс] / Yii Framework – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/ru/db-active-record>

12. Офіційна документація по генератору коду Gii [Електронний ресурс] / Yii Framework – Режим доступу до ресурсу: <https://yiiframework.com.ua/uk/doc/guide/2/start-gii/>

13. Як тестувати веб-сайт: основні види та цілі тестування [Електронний ресурс] / VOLL – Режим доступу до ресурсу: <https://voll.com.ua/uk/blog/kak-testirovat-veb-sajt-osnovnye-vidy-i-celi-testirovaniya>

					ДППЗ.190161.01.12.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		97

ДОДАТОК А
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Робота виконується в рамках проекту розробки інтернет платформи для організації продажу годинників. 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 «The World of Watches» для організації продажу годинників».

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням інтернет-платформи ж забезпечення комфортного вибору та підбору годинників для користувача, з можливістю пошуку, фільтрації, перегляду відгуків та оформлення замовлення.

2.2 Експлуатаційне призначення

Інтернет-платформа може використовуватись на будь якому пристрої де є наявний браузер та доступ до всесвітньої мережі інтернет.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

Програма повинна забезпечувати можливості виконання таких функцій:

- функція авторизації;
- функція реєстрації;
- функція перегляду кошика;
- функція очистки кошика;
- функція оформлення замовлення;
- функція перегляду товарів за певною категорією;
- функція фільтрації товарів;
- функція пошуку товарів;
- функція перегляду детальної інформації про товар;
- функція перегляду відгуків;
- можливість залишити відгук;

- можливість додати товар до списку бажань;
- можливість додати товар до порівнянь;
- можливість перегляду фото на сторінці товару;
- можливість додавати фото у опис товару;
- можливість переглядати рейтинг товару;

3.2 Вимоги до надійності

Інтернет платформа повинна забезпечувати наступні вимоги до надійності:

– усі деструктивні дії (видалення або зміна будь якого об'єкта моделі) повинні отримувати додаткове підтвердження від користувача перед тим, як ці зміни будуть застосовані;

- збереження даних кошика у масиві сесій.
- Повідомляти користувача про його не вірні дії;
- проводити валідацію введених даних;
- можливість самовідновлення після збоїв;

3.3 Умови експлуатації та вимоги до технічних засобів

Система повинна працювати без проблем на усіх пристроях де наявний браузер, то ж мінімальними характеристиками платформи є мінімальні характеристики браузера:

Windows:

- Windows 7, Windows 8, Windows 8.1, Windows 10;
- Процесор Intel Pentium 4 або новіший з підтримкою SSE3;
- наявність мережевої плати або wifi адаптера.

macOS:

- OS X El Capitan (10.11) або новішої версії.

Linux:

– Ubuntu 18.04 (64-розрядна версія) або новіша версія, Debian 10 або новіша версія, openSUSE 15.2 або новіша версія, Fedora Linux 32 або новіша версія.

- Процесор Intel Pentium 4 або новіша версія з підтримкою SSE3.

Android

– Android 6.0 Marshmallow або новіша версія.

3.4 Вимоги до інформаційної та програмної сумісності

При розробці інтернет-платформи будуть використовуватись такі технології:

- PHP - скриптова мова програмування, була створена для генерації HTML-сторінок на стороні вебсервера;
- Yii2 - це високопродуктивний вебфреймворк, написаний на PHP, реалізує парадигму модель-вид-контролер;
- jQuery - популярна JavaScript-бібліотека з відкритим кодом;
- Ajax - підхід до побудови користувацьких інтерфейсів вебзастосунків, за яких веб-сторінка, не перезавантажується, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані;
- Bootstrap - це безкоштовний набір інструментів з відкритим кодом, призначений для створення вебсайтів та вебдодатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript.

3.5 Спеціальні вимоги

Інтернет-платформа повинна мати простий та зрозумілий інтерфейс для кінцевого користувача та адаптивний під усі види пристроїв, та під будь яку діагональ екрану.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- робочий проект;
- вихідні коди з коментарями;
- керівництво користувача;
- контекстно-залежну допомогу;
- керівництво по встановленню/розгортанню системи.

5 Стадії та етапи розробки

Стадії та етапи розробки інтернет платформи для організації продажу годинників та їх підбору 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 – 14.02.22	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури
Технічний проект 15.02.22 – 28.02.22	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми
Робочий проект 01.03.22 – 10.05.22	Розробка програмного забезпечення	Реалізація програмного забезпечення; відлагодження; проведення попереднього тестування
Розробка програмної документації 11.05.22 – 20.05.22	Розробка документації до програмного забезпечення	Розробка документації, відповідно до технічного завдання
Тестування системи 21.05.22 – 30.05.22	Проведення тестування програмного забезпечення	Вибір методів тестування; проведення обраних тестів; коректування програмного забезпечення
Впровадження	Підготовка і передача програми	Завантаження веб-сайту на хостинг

6 Порядок контролю та приймання

Контроль здійснюється кінцевими користувачами системи, підключеними на етапі тестування додатку. Приймання програмної системи здійснюється після завантаження веб-сайту на хостинг.

ДОДАТОК Б
(обов'язковий)

ДІАГРАМА «СУТНІСТЬ-ЗВ'ЯЗОК»

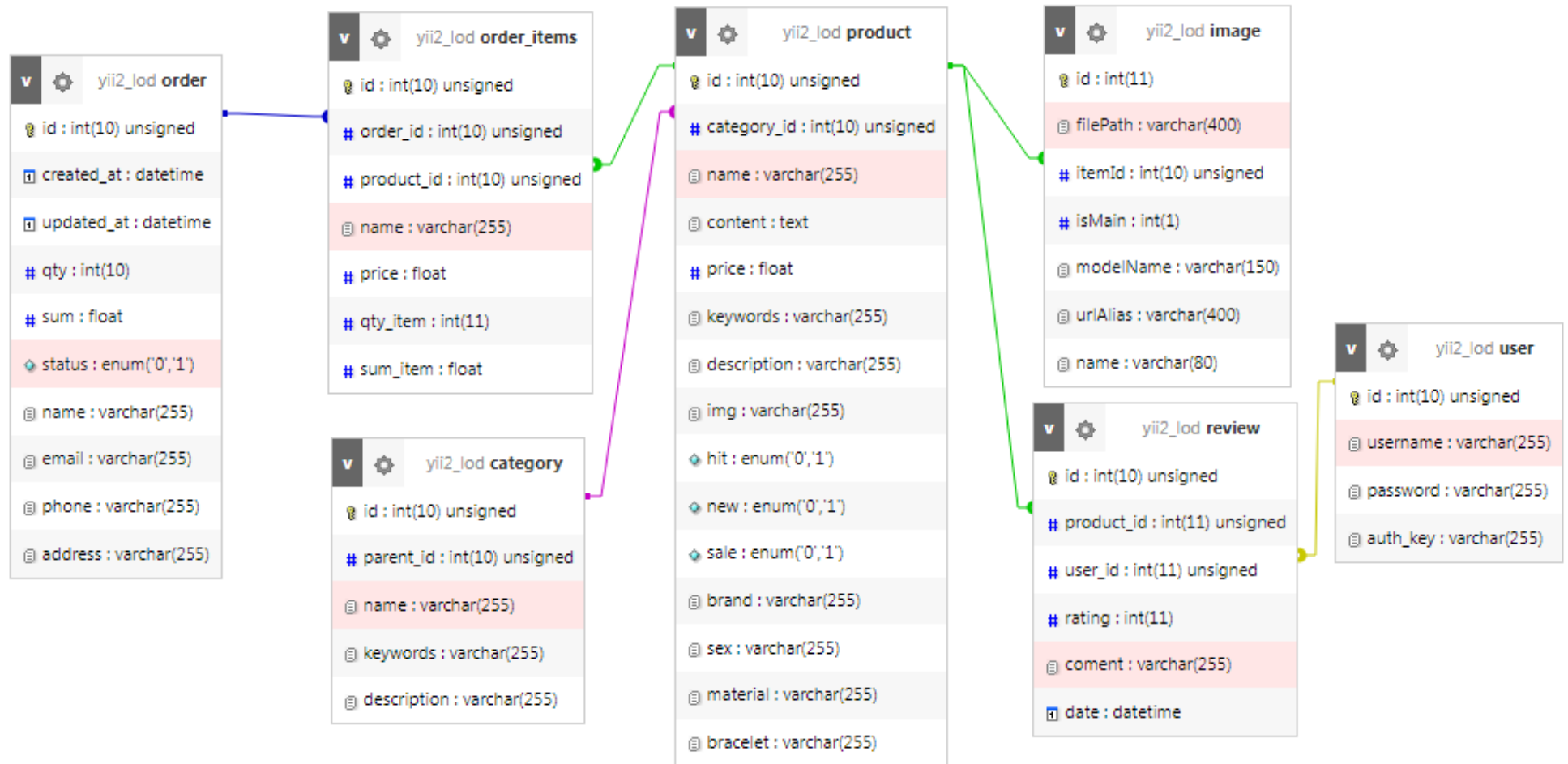


Рисунок Б.1 – Діаграма «Сутність-зв'язок»

ДОДАТОК В
(обов'язковий)

Вигляд користувацького інтерфейсу

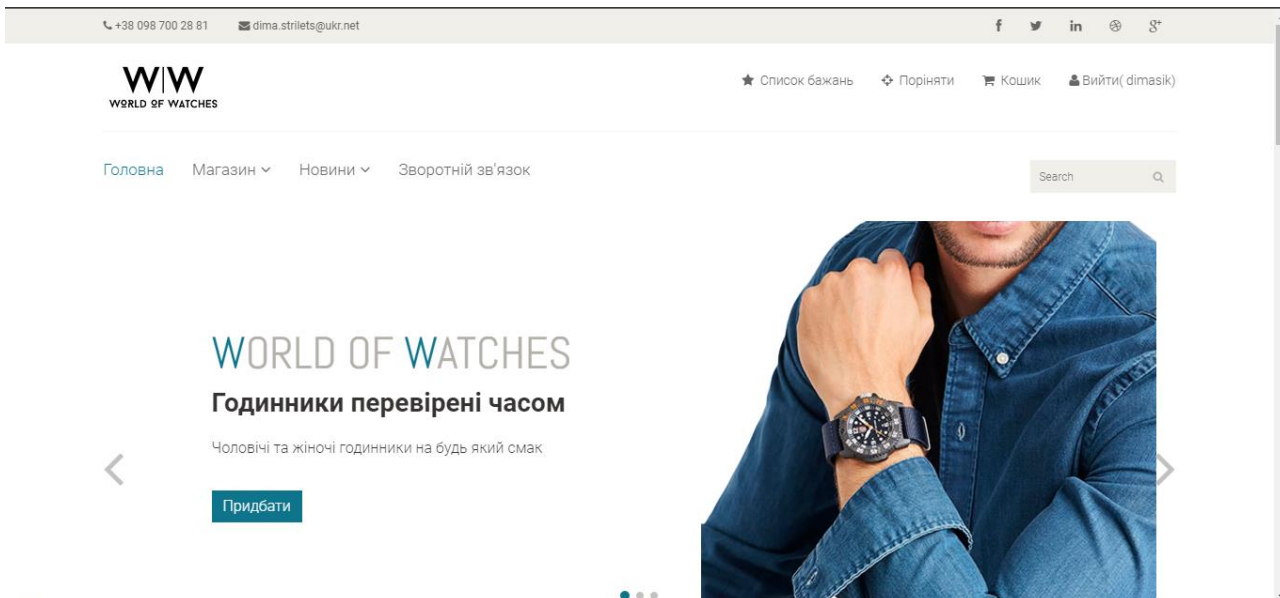


Рисунок В.1 – Перша частина інтерфейсу головної сторінки

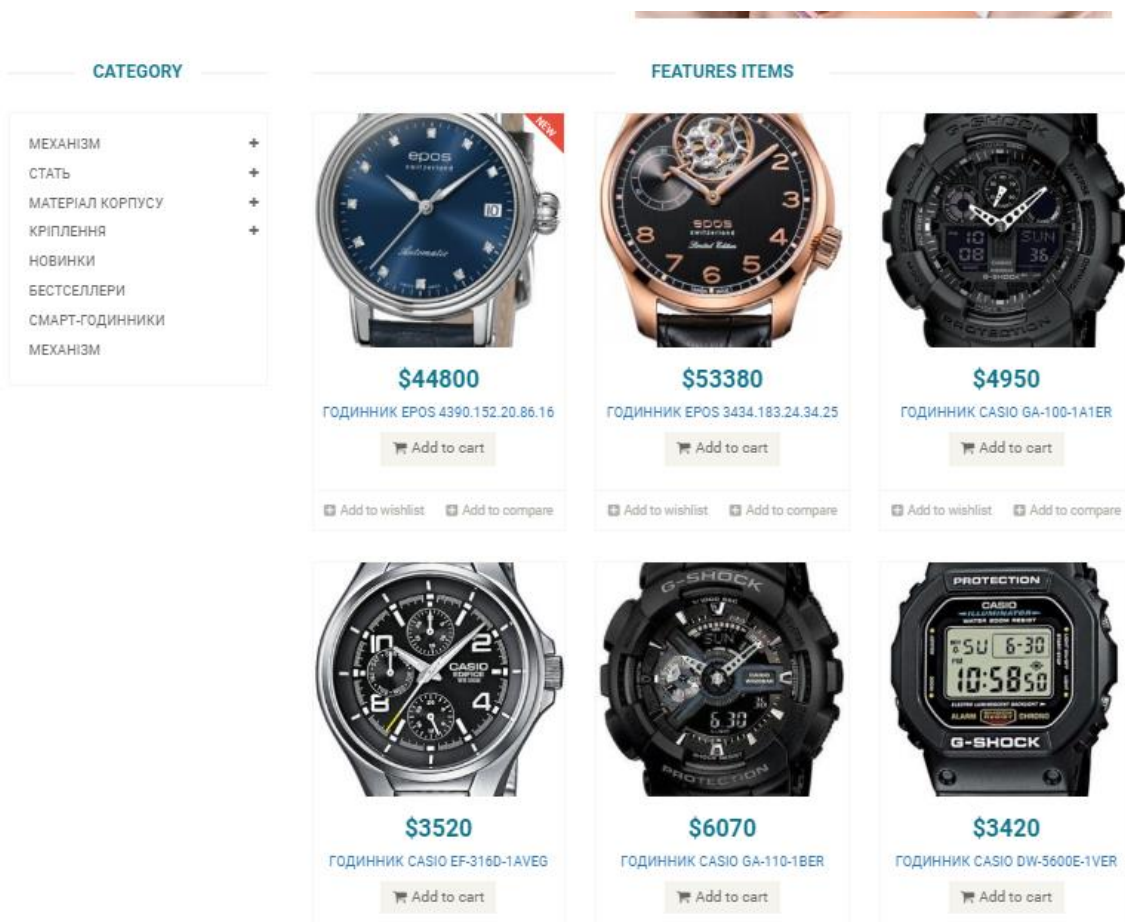


Рисунок В.2 – Друга частина інтерфейсу головної сторінки

CATEGORY


- МЕХАНІЗМ +
- СТАТЬ +
- МАТЕРІАЛ КОРПУСУ +
- КРІПЛЕННЯ +
- НОВИНКИ
- БЕСТСЕЛЛЕРИ
- СМАРТ-ГОДИННИКИ
- МЕХАНІЗМ

BRANDS

- ACNE (50)
- GRÜNE ERDE (56)
- ALBIRO (27)
- RONHILL (32)
- ODDMOLLY (5)
- BOUDESTIJN (9)
- RÖSCH CREATIVE CULTURE (4)

PRICE RANGE

ПОШУК ПО ЗАПИТУ: CASIO




\$4950

ГОДИННИК CASIO GA-100-1A1ER

Add to cart

Add to wishlist Add to compare




\$3520

ГОДИННИК CASIO EF-316D-1AVEG

Add to cart

Add to wishlist Add to compare



\$6070

ГОДИННИК CASIO GA-110-1BER

Add to cart

Add to wishlist Add to compare

1 2 >

Рисунок В.3 – Інтерфейс пошуку

FILTER

Категорія
Кварцеві

Назва

Бренд

Стать
Будь яка

Матеріал

Браслет/Ремінець
Браслет/Ремінець


Мінімальна ціна

Максимальна ціна

Знайти

ПОШУК НЕОБХІДНОГО ГОДИННИКА

Показані 1-2 із 2 записів.




\$4950

ГОДИННИК CASIO GA-100-1A1ER

Add to cart

Add to wishlist Add to compare



\$3520

ГОДИННИК CASIO EF-316D-1AVEG

Add to cart

Add to wishlist Add to compare

Рисунок В.4 – Інтерфейс перегляду товарів певної категорії

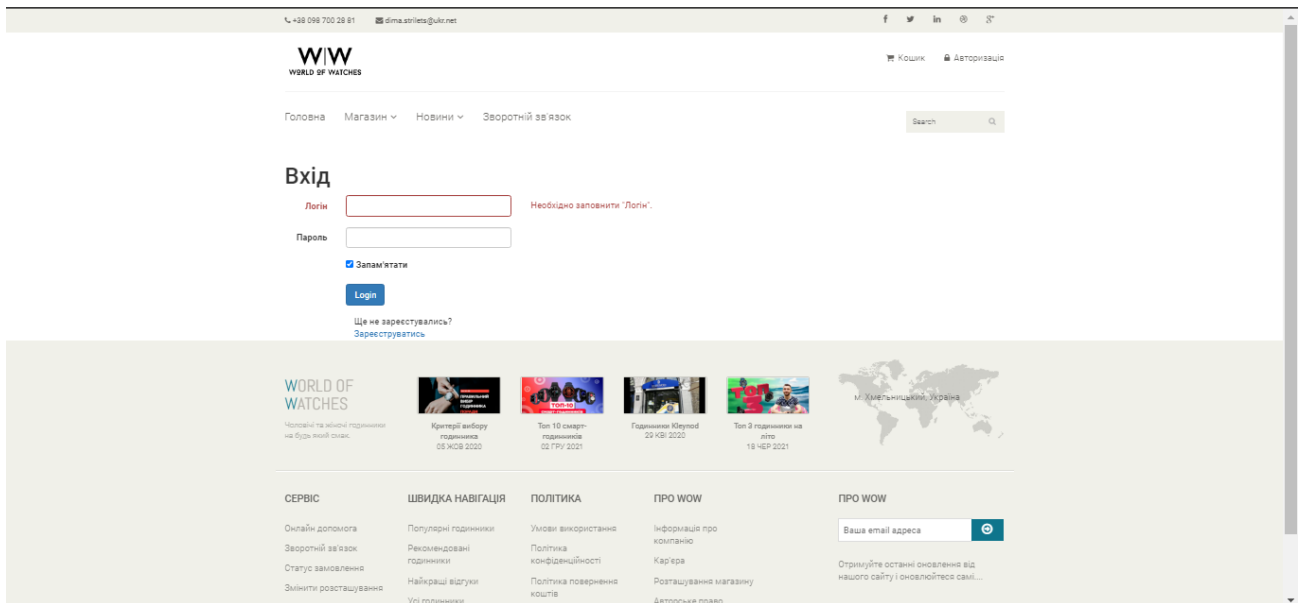


Рисунок В.5 – Інтерфейс авторизації

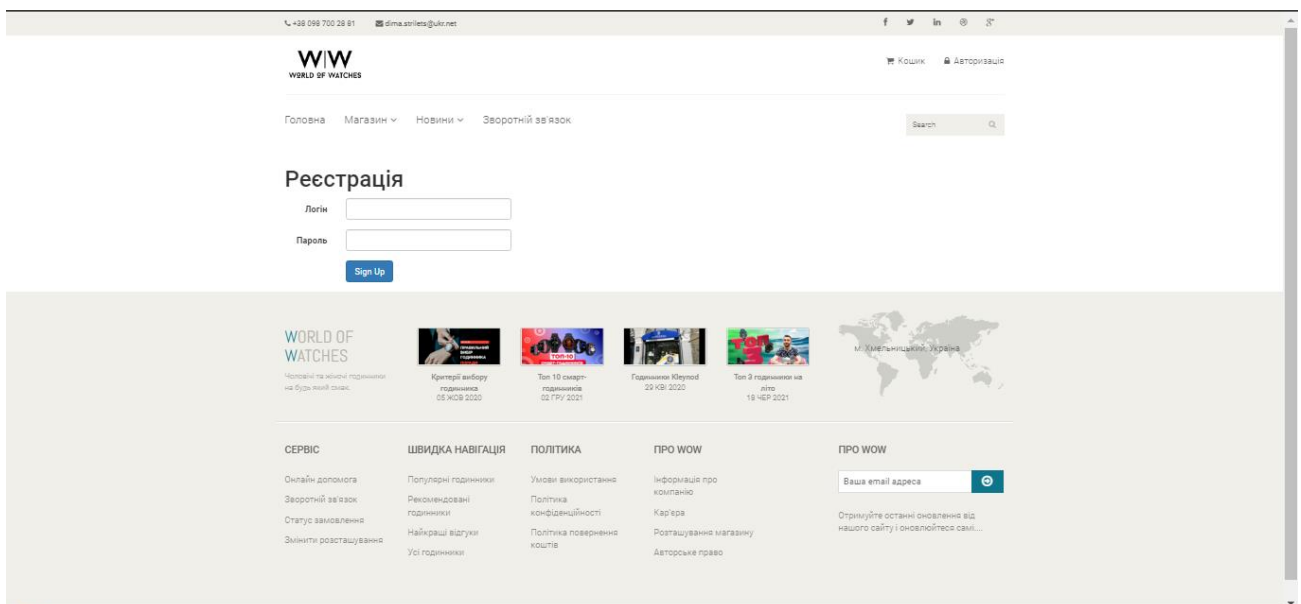


Рисунок В.6 – Інтерфейс реєстрації

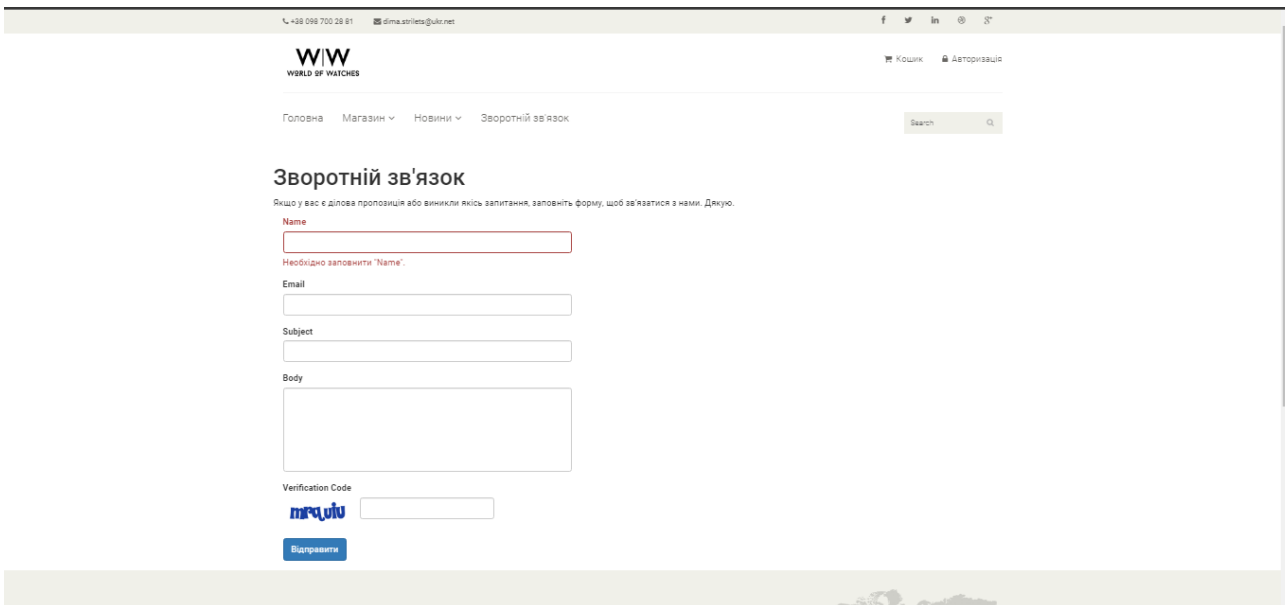


Рисунок В.7 – Інтерфейс зворотнього зв'язку

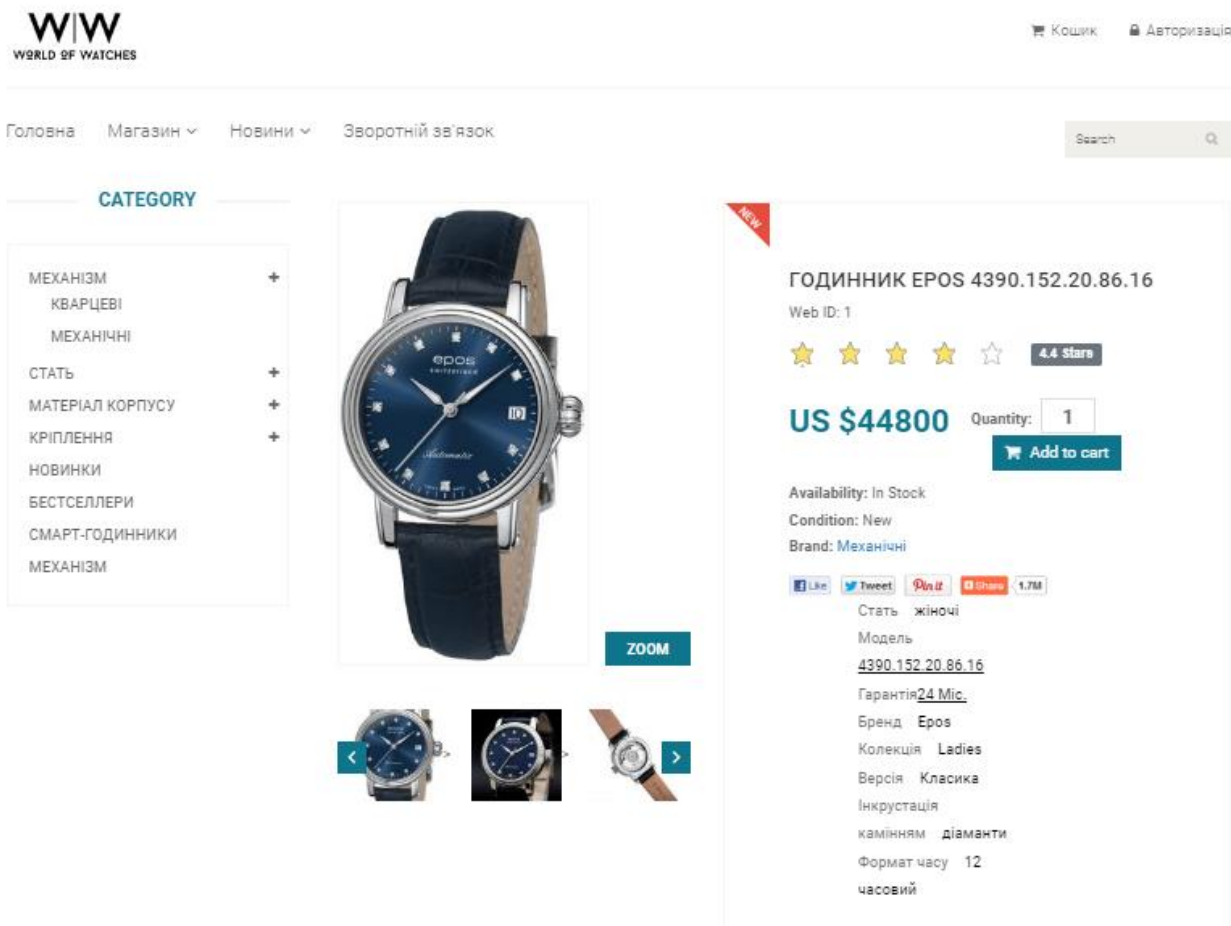


Рисунок В.8 – Інтерфейс перегляду товару

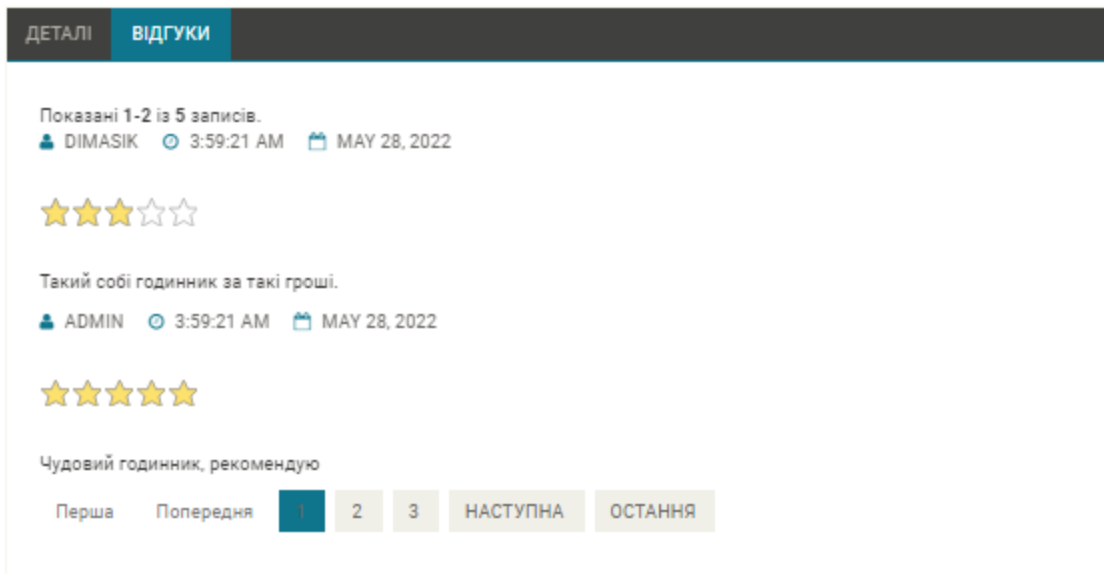


Рисунок В.9 – Інтерфейс перегляду відгуків

The screenshot shows a form titled 'Залиште свій відгук'. It includes a dropdown menu for 'Ім'я користувача' with 'dimasik' selected. Below it is a 'Рейтинг' section with five empty star icons. A 'Коментар' section features a large text input area. At the bottom left, there is a blue button labeled 'Опублікувати'.

Рисунок В.10 – Інтерфейс публікації відгуку

ДОДАТОК Г
(обов'язковий)

КОД (ЛІСТИНГ) ПРОГРАМИ

Клас Cart.php:

```

<?php

namespace app\models;
use yii\db\ActiveRecord;

class Cart extends ActiveRecord{

    public function behaviors()
    {
        return [
            'image' => [
                'class' => 'rico\yii2images\behaviors\ImageBehave',
            ]
        ];
    }

    public function addToCart($product, $qty = 1){
        $mainImg = $product->getImage();
        if(isset($_SESSION['cart'][$product->id])){
            $_SESSION['cart'][$product->id]['qty'] += $qty;
        }else{
            $_SESSION['cart'][$product->id] = [
                'qty' => $qty,
                'name' => $product->name,
                'price' => $product->price,
                'img' => $mainImg->getUrl('x50')
            ];
        }
        $_SESSION['cart.qty'] = isset($_SESSION['cart.qty']) ?
$_SESSION['cart.qty'] + $qty : $qty;
        $_SESSION['cart.sum'] = isset($_SESSION['cart.sum']) ?
$_SESSION['cart.sum'] + $qty * $product->price : $qty * $product->price;
    }

    public function recalc($id){
        if(!isset($_SESSION['cart'][$id])) return false;
        $qtyMinus = $_SESSION['cart'][$id]['qty'];
        $sumMinus = $_SESSION['cart'][$id]['qty'] *
$_SESSION['cart'][$id]['price'];
        $_SESSION['cart.qty'] -= $qtyMinus;
        $_SESSION['cart.sum'] -= $sumMinus;
    }
}

```

```

        unset($_SESSION['cart'][$id]);
    }
}

```

Клас Category.php:

```

<?php
namespace app\models;
use yii\db\ActiveRecord;
class Category extends ActiveRecord{

    public function behaviors()
    {
        return [
            'image' => [
                'class' => 'rico\yii2images\behaviors\ImageBehave',
            ]
        ];
    }

    public static function tableName(){
        return 'category';
    }

    public function getProducts(){
        return $this->hasMany(Product::className(), ['category_id' => 'id']);
    }
}

```

Клас ContactForm.php:

```

<?php

namespace app\models;

use Yii;
use yii\base\Model;

/**
 * ContactForm is the model behind the contact form.
 */
class ContactForm extends Model
{

```

```

public $name;
public $email;
public $subject;
public $body;
public $verifyCode;

/**
 * @return array the validation rules.
 */
public function rules()
{
    return [
        // name, email, subject and body are required
        [['name', 'email', 'subject', 'body'], 'required'],
        // email has to be a valid email address
        ['email', 'email'],
        // verifyCode needs to be entered correctly
        ['verifyCode', 'captcha'],
    ];
}

/**
 * @return array customized attribute labels
 */
public function attributeLabels()
{
    return [
        'verifyCode' => 'Verification Code',
    ];
}

/**
 * Sends an email to the specified email address using the information
collected by this model.
 * @param string $email the target email address
 * @return boolean whether the model passes validation
 */
public function contact($email)
{
    if ($this->validate()) {
        Yii::$app->mailer->compose()
            ->setTo($email)
            ->setFrom([$this->email => $this->name])
            ->setSubject($this->subject)
    }
}

```

```

        ->setTextBody($this->body)
        ->send();

        return true;
    }
    return false;
}
}

```

Клас LoginForm.php:

```

<?php

namespace app\models;

use Yii;
use yii\base\Model;

/**
 * LoginForm is the model behind the login form.
 */
class LoginForm extends Model
{
    public $username;
    public $password;
    public $rememberMe = true;

    private $_user = false;

    /**
     * @return array the validation rules.
     */
    public function rules()
    {
        return [
            // username and password are both required
            [['username', 'password'], 'required'],
            // rememberMe must be a boolean value
            ['rememberMe', 'boolean'],
            // password is validated by validatePassword()
            ['password', 'validatePassword'],
        ];
    }
}

```

```

public function attributeLabels(){
    return [
        'username' => 'Логін',
        'password' => 'Пароль',
        'rememberMe' => 'Запам\'ятати',
    ];
}

/**
 * Validates the password.
 * This method serves as the inline validation for password.
 *
 * @param string $attribute the attribute currently being validated
 * @param array $params the additional name-value pairs given in the rule
 */
public function validatePassword($attribute, $params)
{
    if (!$this->hasErrors()) {
        $user = $this->getUser();

        if (!$user || !$user->validatePassword($this->password)) {
            $this->addError($attribute, 'Логин/пароль введены не верно');
        }
    }
}

/**
 * Logs in a user using the provided username and password.
 * @return boolean whether the user is logged in successfully
 */
public function login()
{
    if ($this->validate()) {
        if($this->rememberMe){
            $u = $this->getUser();
            $u->generateAuthKey();
            $u->save();
        }
        return Yii::$app->user->login($this->getUser(), $this->rememberMe ?
3600*24*30 : 0);
    }
    return false;
}

```

```

/**
 * Finds user by [[username]]
 *
 * @return User|null
 */
public function getUser()
{
    if ($this->_user === false) {
        $this->_user = User::findByUsername($this->username);
    }

    return $this->_user;
}
}

```

Клас Order.php:

```

<?php

namespace app\models;

use yii\db\ActiveRecord;
use Yii;
use yii\behaviors\TimestampBehavior;
use yii\db\Expression;

/**
 * This is the model class for table "order".
 *
 * @property string $id
 * @property string $created_at
 * @property string $updated_at
 * @property integer $qty
 * @property double $sum
 * @property string $status
 * @property string $name
 * @property string $email
 * @property string $phone
 * @property string $address
 */
class Order extends ActiveRecord
{
    /**

```

```

    * @inheritdoc
    */
    public static function tableName()
    {
        return 'order';
    }

    public function behaviors(){
        return [
            [
                'class' => TimestampBehavior::className(),
                'attributes' => [
                    ActiveRecord::EVENT_BEFORE_INSERT => ['created_at',
'updated_at'],
                    ActiveRecord::EVENT_BEFORE_UPDATE => ['updated_at'],
                ],
                'value' => new Expression('NOW()'),
            ],
        ];
    }

    public function getOrderItems(){
        return $this->hasMany(OrderItems::className(), ['order_id' => 'id']);
    }

    /**
     * @inheritdoc
     */
    public function rules()
    {
        return [
            [['name', 'email', 'phone', 'address'], 'required'],
            [['created_at', 'updated_at'], 'safe'],
            [['qty'], 'integer'],
            [['sum'], 'number'],
            [['status'], 'boolean'],
            [['name', 'email', 'phone', 'address'], 'string', 'max' => 255],
        ];
    }

    /**
     * @inheritdoc
     */
    public function attributeLabels()

```

```

    {
        return [
            'name' => 'Ім\ 'я',
            'email' => 'Е-mail',
            'phone' => 'Телефон',
            'address' => 'Адреса',
        ];
    }
}

```

Клас OrderItems.php:

```

<?php

namespace app\models;

use yii\db\ActiveRecord;
use Yii;

/**
 * This is the model class for table "order_items".
 *
 * @property string $id
 * @property string $order_id
 * @property string $product_id
 * @property string $name
 * @property double $price
 * @property integer $qty_item
 * @property double $sum_item
 */
class OrderItems extends \yii\db\ActiveRecord
{
    /**
     * @inheritdoc
     */
    public static function tableName()
    {
        return 'order_items';
    }

    public function getOrder(){
        return $this->hasOne(Order::className(), ['id' => 'order_id']);
    }
}

```

```

/**
 * @inheritdoc
 */
public function rules()
{
    return [
        [['order_id', 'product_id', 'name', 'price', 'qty_item',
'sum_item'], 'required'],
        [['order_id', 'product_id', 'qty_item'], 'integer'],
        [['price', 'sum_item'], 'number'],
        [['name'], 'string', 'max' => 255],
    ];
}
}

```

Клас Product.php:

```

<?php
namespace app\models;
use yii\db\ActiveRecord;
/**
 * This is the model class for table "product".
 *
 * @property string $id
 * @property string $category_id
 * @property string $name
 * @property string $content
 * @property double $price
 * @property string $keywords
 * @property string $description
 * @property string $img
 * @property string $hit
 * @property string $new
 * @property string $sale
 * @property string $brand
 * @property string $sex
 * @property string $material
 * @property string $bracelet
 */
class Product extends ActiveRecord{

    public function behaviors()

```

```

{
    return [
        'image' => [
            'class' => 'rico\yii2images\behaviors\ImageBehave',
        ]
    ];
}

public function rules()
{
    return [
        [['category_id', 'name'], 'required'],
        [['category_id'], 'integer'],
        [['content', 'hit', 'new', 'sale'], 'string'],
        [['price'], 'number'],
        [['name', 'keywords', 'description', 'img', 'brand', 'sex',
'material', 'bracelet'], 'string', 'max' => 255],
        [['image'], 'file', 'extensions' => 'png, jpg'],
        [['gallery'], 'file', 'extensions' => 'png, jpg', 'maxFiles' => 4],
    ];
}

/**
 * @inheritdoc
 */
public function attributeLabels()
{
    return [
        'id' => 'ID товару',
        'category_id' => 'Категорія',
        'name' => 'Назва',
        'content' => 'Контент',
        'price' => 'Ціна',
        'keywords' => 'Ключові слова',
        'description' => 'Мета-опис',
        'image' => 'Фото',
        'gallery' => 'Галерея',
        'hit' => 'Хіт',
        'new' => 'Новинка',
        'sale' => 'Розпродаж',
        'brand' => 'Бренд',
        'sex' => 'Стать',
        'material' => 'Матеріал',
        'bracelet' => 'Браслет/Ремінець',
        'min_price' => 'Мінімальна ціна',
    ];
}

```

```

        'max_price' => 'Максимальна ціна'
    ];
}
public static function tableName(){
    return 'product';
}

public function getCategory(){
    return $this->hasOne(Category::className(), ['id' => 'category_id']);
}
}

```

Клас OrderItems.php:

```

<?php
namespace app\models;
use yii\db\ActiveRecord;
/**
 * This is the model class for table "product".
 *
 * @property string $id
 * @property string $category_id
 * @property string $name
 * @property string $content
 * @property double $price
 * @property string $keywords
 * @property string $description
 * @property string $img
 * @property string $hit
 * @property string $new
 * @property string $sale
 * @property string $brand
 * @property string $sex
 * @property string $material
 * @property string $bracelet
 */
class Product extends ActiveRecord{

    public function behaviors()
    {
        return [

```

```

        'image' => [
            'class' => 'rico\yii2images\behaviors\ImageBehave',
        ]
    ];
}
public function rules()
{
    return [
        [['category_id', 'name'], 'required'],
        [['category_id'], 'integer'],
        [['content', 'hit', 'new', 'sale'], 'string'],
        [['price'], 'number'],
        [['name', 'keywords', 'description', 'img', 'brand', 'sex',
'material', 'bracelet'], 'string', 'max' => 255],
        [['image'], 'file', 'extensions' => 'png, jpg'],
        [['gallery'], 'file', 'extensions' => 'png, jpg', 'maxFiles' => 4],
    ];
}

/**
 * @inheritdoc
 */
public function attributeLabels()
{
    return [
        'id' => 'ID товару',
        'category_id' => 'Категорія',
        'name' => 'Назва',
        'content' => 'Контент',
        'price' => 'Ціна',
        'keywords' => 'Ключові слова',
        'description' => 'Мета-опис',
        'image' => 'Фото',
        'gallery' => 'Галерея',
        'hit' => 'Хіт',
        'new' => 'Новинка',
        'sale' => 'Розпродаж',
        'brand' => 'Бренд',
        'sex' => 'Стать',
        'material' => 'Матеріал',
        'bracelet' => 'Браслет/Ремінець',
        'min_price' => 'Мінімальна ціна',
        'max_price' => 'Максимальна ціна'
    ];
}

```

```

    }
    public static function tableName(){
        return 'product';
    }

    public function getCategory(){
        return $this->hasOne(Category::className(), ['id' => 'category_id']);
    }
}

```

Клас Review.php:

```

<?php

namespace app\models;

use Yii;

/**
 * This is the model class for table "review".
 *
 * @property int $id
 * @property int $product_id
 * @property int $user_id
 * @property int $rating
 * @property int $coment
 * @property string $date
 */
class Review extends \yii\db\ActiveRecord
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'review';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {

```

```

return [
    [['coment', 'rating'], 'required'],
    [['product_id', 'user_id', 'rating'], 'integer'],
    [['coment'], 'string'],
    [['date'], 'safe'],
];
}

/**
 * {@inheritdoc}
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'product_id' => 'Product ID',
        'user_id' => 'Ім'я користувача',
        'rating' => 'Рейтинг',
        'coment' => 'Коментар',
        'date' => 'Date',
    ];
}

public function getUser(){
    return $this->hasOne(User::className(), ['id' => 'user_id']);
}

public function getProduct(){
    return $this->hasOne(Product::className(), ['id' => 'product_id']);
}
}

```

Клас User.php:

```

<?php

namespace app\models;
use Yii;
use yii\db\ActiveRecord;

class User extends ActiveRecord implements \yii\web\IdentityInterface
{

    public static function tableName(){
        return 'user';
    }
}

```

```
}

/**
 * @inheritdoc
 */
public static function findIdentity($id)
{
    return static::findOne($id);
}

/**
 * @inheritdoc
 */
public static function findIdentityByAccessToken($token, $type = null)
//
    return static::findOne(['access_token' => $token]);
}

/**
 * Finds user by username
 *
 * @param string $username
 * @return static|null
 */
public static function findByUsername($username)
{
    return static::findOne(['username' => $username]);
}

/**
 * @inheritdoc
 */
public function getId()
{
    return $this->id;
}

/**
 * @inheritdoc
 */
public function getAuthKey()
{
    return $this->auth_key;
}
```

```

/**
 * @inheritdoc
 */
public function validateAuthKey($authKey)
{
    return $this->auth_key === $authKey;
}

/**
 * Validates password
 *
 * @param string $password password to validate
 * @return boolean if password provided is valid for current user
 */
public function validatePassword($password)
{
    // return $this->password === $password;
    return \Yii::$app->security->validatePassword($password, $this->password);
}

public function generateAuthKey(){
    $this->auth_key = \Yii::$app->security->generateRandomString();
}

public function setUsername($name)
{
    $this->username = $name;
}

public function setPassword($pass)
{
    $hash = \Yii::$app->security->generatePasswordHash($pass);
    $this->password = $hash;
}
}

```

Контролер CartController.php:

```
<?php
```

```

namespace app\controllers;
use app\models\Product;
use app\models\Cart;
use app\models\Order;
use app\models\OrderItems;
use Yii;

class CartController extends AppController{

    public function actionAdd(){
        $id = Yii::$app->request->get('id');
        $qty = (int)Yii::$app->request->get('qty');
        $qty = !$qty ? 1 : $qty;
        $product = Product::findOne($id);
        if(empty($product)) return false;
        $session =Yii::$app->session;
        $session->open();
        $cart = new Cart();
        $cart->addToCart($product, $qty);
        if( !Yii::$app->request->isAjax ){
            return $this->redirect(Yii::$app->request->referrer);
        }
        $this->layout = false;
        return $this->render('cart-modal', compact('session'));
    }

    public function actionClear(){
        $session =Yii::$app->session;
        $session->open();
        $session->remove('cart');
        $session->remove('cart.qty');
        $session->remove('cart.sum');
        $this->layout = false;
        return $this->render('cart-modal', compact('session'));
    }

    public function actionDelItem(){
        $id = Yii::$app->request->get('id');
        $session =Yii::$app->session;
        $session->open();
        $cart = new Cart();
        $cart->recalc($id);
    }
}

```

```

$this->layout = false;
return $this->render('cart-modal', compact('session'));
}

public function actionShow(){
    $session =Yii::$app->session;
    $session->open();
    $this->layout = false;
    return $this->render('cart-modal', compact('session'));
}

public function actionView(){
    $session = Yii::$app->session;
    $session->open();
    $this->setMeta('Кошик');
    $order = new Order();
    if( $order->load(Yii::$app->request->post()) ){
        $order->qty = $session['cart.qty'];
        $order->sum = $session['cart.sum'];
        if($order->save()){
            $this->saveOrderItems($session['cart'], $order->id);
            Yii::$app->session->setFlash('success', 'Ваше замовлення
прийнято. Менеджер незабаром зв\'яжеться з вами.');
```

Yii::\$app->mailer->compose('order', ['session' => \$session])

->setFrom(['dima.strilets5079@gmail.com' => 'World Of
Watches'])

->setTo(\$order->email)

->setSubject('Замовлення')

->send();

\$session->remove('cart');

\$session->remove('cart.qty');

\$session->remove('cart.sum');

return \$this->refresh();

```

        }else{
            Yii::$app->session->setFlash('error', 'Ошибка оформления
заказа');
```

}

}

return \$this->render('view', compact('session', 'order'));

}

```

protected function saveOrderItems($items, $order_id){
    foreach($items as $id => $item){
        $order_items = new OrderItems();
```

```

        $order_items->order_id = $order_id;
        $order_items->product_id = $id;
        $order_items->name = $item['name'];
        $order_items->price = $item['price'];
        $order_items->qty_item = $item['qty'];
        $order_items->sum_item = $item['qty'] * $item['price'];
        $order_items->save();
    }
}
}

```

Контролер CategoryController:

```

<?php
/**
 * Created by PhpStorm.
 * User: Andrey
 * Date: 08.05.2016
 * Time: 10:00
 */

namespace app\controllers;
use app\models\Category;
use app\models\Product;
use app\models\ProductSearch;
use Yii;
use yii\data\ActiveDataProvider;
use yii\data\Pagination;
use yii\filters\VerbFilter;

class CategoryController extends AppController{

    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ];
    }
}

```

```

}
public function actionIndex(){
    $hits = Product::find()->where(['hit' => '1']->limit(6)->all();
    $this->setMeta('wow');
    return $this->render('index', compact('hits'));
}

public function actionView($id = NULL){
    $searchModel = new ProductSearch();
    if(isset($id)){
        $searchModel->category_id = $id;
    }
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

    return $this->render('view', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

public function actionSearch(){
    $q = trim(Yii::$app->request->get('q'));
    $this->setMeta('wow | Помык: ' . $q);
    if(!$q)
        return $this->render('search');
    $query = Product::find()->where(['like', 'name', $q]);
    $pages = new Pagination(['totalCount' => $query->count(), 'pageSize' =>
3, 'forcePageParam' => false, 'pageSizeParam' => false]);
    $products = $query->offset($pages->offset)->limit($pages->limit)->all();
    return $this->render('search', compact('products', 'pages', 'q'));
}
/**
 * @return \yii\db\ActiveQuery
 */
}

```

ДОДАТОК Д
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Хмельницький Національний Університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

Дипломний проект на тему:

Інтернет-платформа «The World of Watches» для організації продажу ГОДИННИКІВ

Студент: Стрілець Дмитро Анатолійович.

Керівник : Онишко Оксана Григорівна канд. пед. наук, доцент.

Мета та завдання проекту

Мета проекту – розробка інтернет-платформи яка дозволить користувачам підбирати та обирати необхідний їм годинник з можливістю переглядати фото у збільшеному масштабі та можливістю перегляду відгуків та рейтингу.

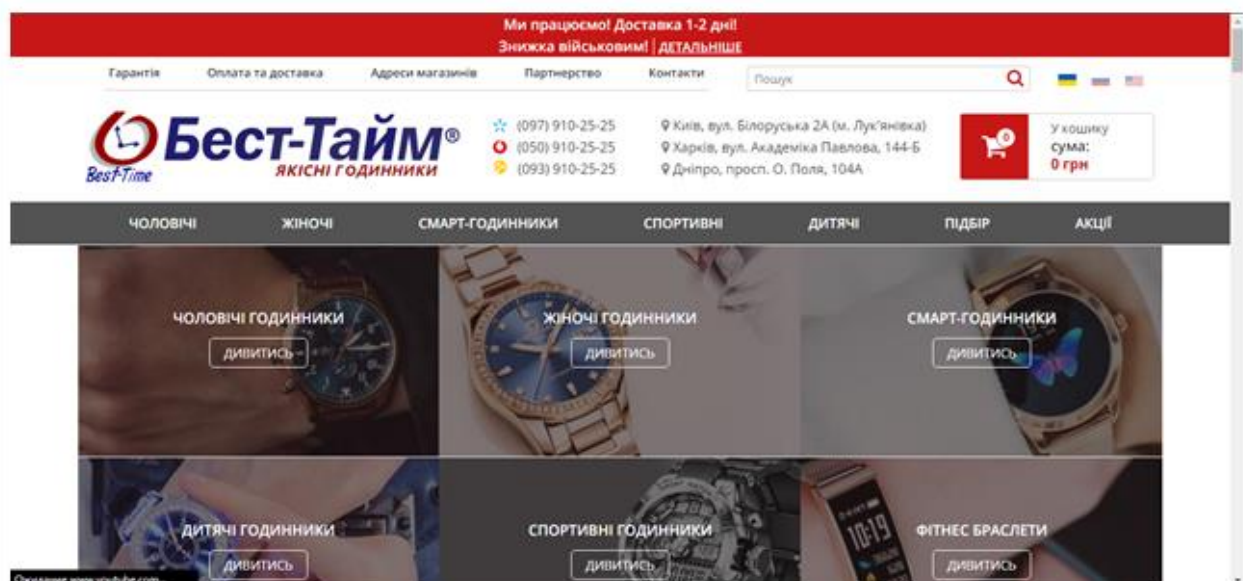
Завдання, які необхідно вирішити для досягнення мети:

- провести аналіз предметної області та визначити її особливості;
- провести аналіз існуючих програмних рішень;
- сформулювати технічне завдання;
- спроектувати архітектуру інтернет-додатку;
- ознайомитися з можливими методами та способами розробки додатку;
- виконати програмну реалізацію проекту використовуючи обраний фреймворк;
- провести тестові випробування додатку;

Актуальність теми





Актуальність теми дипломного проектування полягає в тому, що усі люди користуються годинниками, а наявні інтернет-платформи для купівлі годинників не надають усіх необхідних можливостей для комфортного вибору необхідного годинника або підбору його за параметрами які вам необхідні. Окрім того, усі наявні рішення надають адміністраторам можливість фільтрувати увесь контент на сайті та видаляти відгуки які залишили користувачі, тим самим забираючи у потенційних власників годинника можливість прочитати реальні відгуки від користувача конкретного годинника.


best-time.biz



deka.ua

0 800 211 652 0 компаний Посетители RU

DEKA БРЕНДЫ ЧАСЫ ЭЛИТНЫЕ ЧАСЫ УКРАШЕНИЯ АКСЕССУАРЫ SALE Поиск    







МАРУЧНЫЕ ЧАСЫ

LONGINES <https://deka.ua/naruchnye-chasy/brand/rado/> **TISSOT** SWISS WATCHES SINCE 1853 **RADO** SWITZERLAND **TAGHeuer** *Atlantic* **MICHAEL KORS**

art-de-luxe.com.ua

Доставка и оплата [Отзывы](#) [Скидки](#) [Блог](#) [Контакты](#)

 Самые низкие цены
 Оплата при получении
 Гарантия качества
 Отправка в день заказа

 067 684 77 11
 099 684 77 11
 063 684 77 11
 Пн-Вт: 09-19

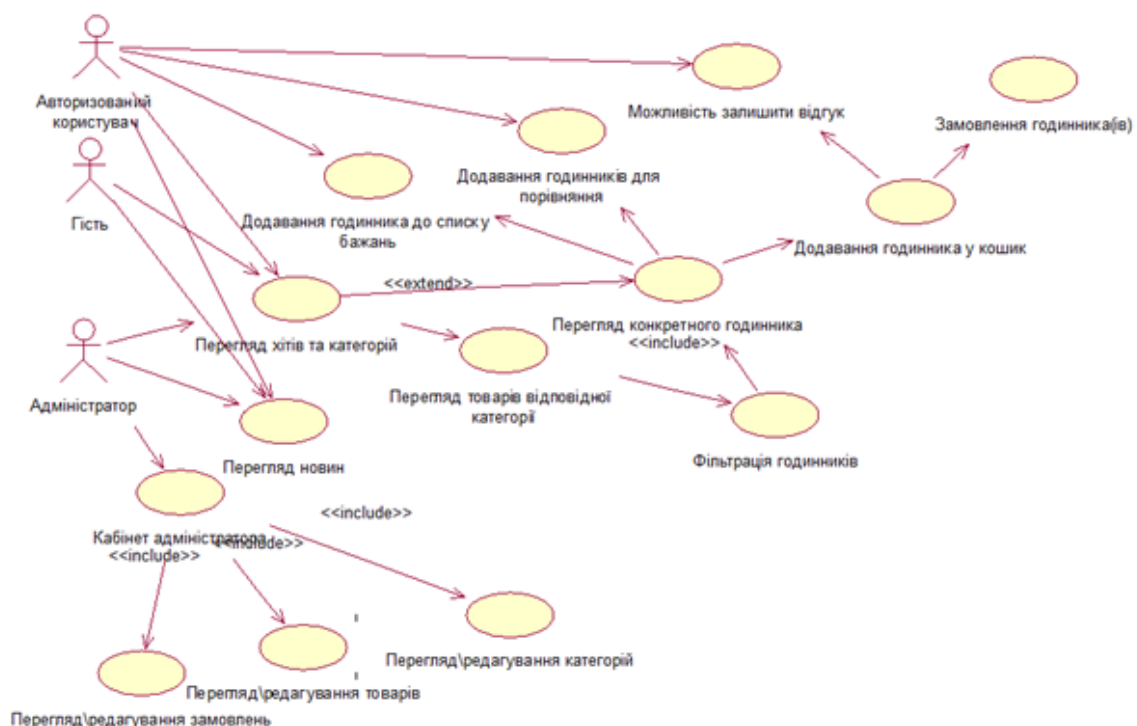
[Платья](#)[Платья](#)
[РАСПРОДАЖА](#)[Смарт-часы](#)[Фитнес-браслеты](#)[Наушники](#)[Часы](#)[Мужские часы](#)[Женские часы](#)[Смарт-часы](#)[Мужские часы](#)Сортировка: [По умолчанию](#) [По возрастанию цены](#) [По убыванию цены](#) [Популярные](#) [Новинки](#)

Результат аналізу подібного ПЗ

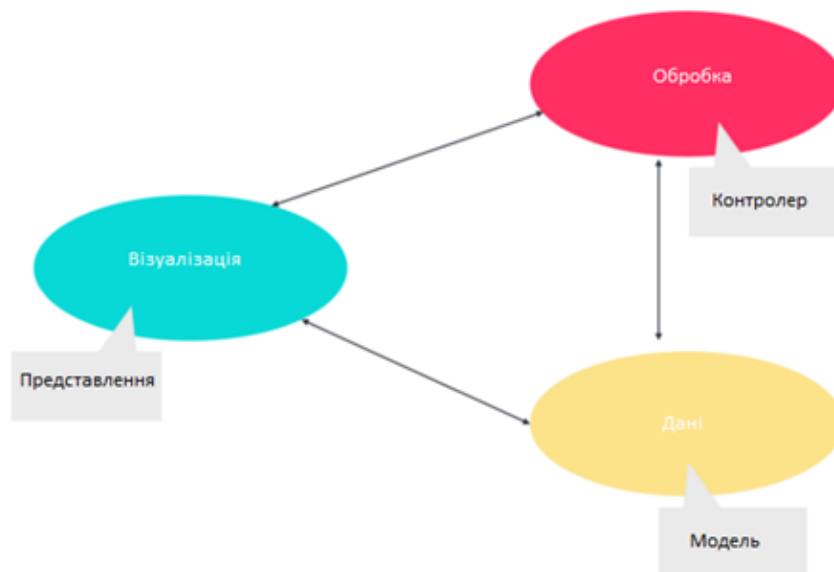
Завдяки проведеному аналізу наявного програмно-технічного забезпечення було визначено усі недоліки та переваги розглянутих рішень, що дозволило визначити, що розроблювана система повина:

- мати зручний, інтуїтивно зрозумілий та адаптивний дизайн;
- мати систему фільтрації за усіма полями.
- мати можливість залишити відгук авторизованим користувачам.
- мати зручну навігацію;
- мати можливість додати товар у список бажань;
- мати можливість додати товар для порівняння.

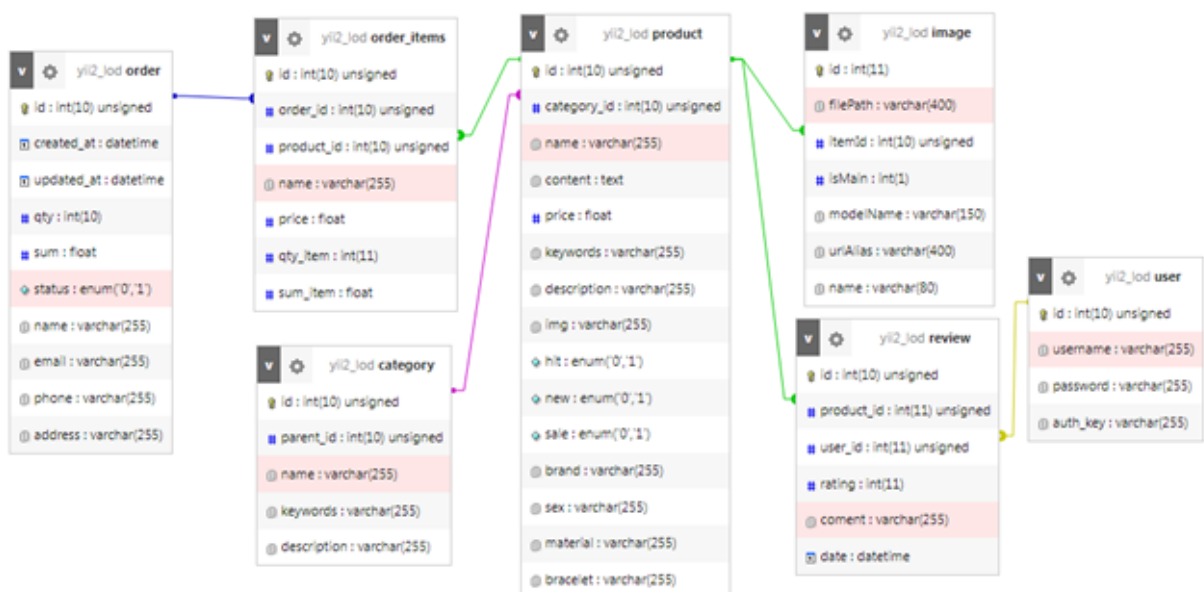
Діаграма варіантів використання



Архітектура інтернет-платформи



Логічна модель бази даних



Використані технології



yii 2.0




The screenshot displays the 'World of Watches' website. The main header includes the logo 'WW WORLD OF WATCHES' and navigation links: 'Список баначь', 'Портити', 'Кощи', and 'Вийти (logout)'. Below the header, there are links for 'Головна', 'Магазин', 'Новини', and 'Зворотній зв'язок'. A search bar is also present. The main content area features a large image of a woman in a brown jacket and the text 'WORLD OF WATCHES' and 'Годинники перевірені часом'. A 'Додати' button is visible. An inset window shows a product listing for watches, organized into a grid with columns for 'CATEGORY' and 'FEATURES ITEMS'. The listing includes six watch items with their respective prices and 'Add to cart' buttons.

CATEGORY	FEATURES ITEMS
МЕЖИМ	
СТАЛЬ	
МАТЕРІАЛ КОЖИ	
КАМЕРНИ	
НОВІНКА	
ВЕЩОВИ	
СМАРТ-ГОДИННИК	
МЕЖИМ	

\$44800	\$53380	\$4950
ГОДИННИК EPIC 4100 182.20.36.16	ГОДИННИК EPIC 1000 182.24.36.20	ГОДИННИК G-DRAGON 182.18.36.20
Add to cart	Add to cart	Add to cart
\$3520	\$6070	\$3420
ГОДИННИК G-DRAGON 182.18.36.20	ГОДИННИК G-DRAGON 182.18.36.20	ГОДИННИК G-DRAGON 182.18.36.20

CATEGORY

- МЕХАНІЗМ +
- СТАТЬ +
- МАТЕРІАЛ КОРПУСУ +
- КРИПЛЕННЯ +
- НОВИНКИ
- БЕСТСЕЛЛЕРИ
- СМАРТ-ГОДИННИКИ
- МЕХАНІЗМ



ГОДИННИК EPOS 4390.152.20.86.16
Web ID: 1

★ ★ ★ ★ ☆ **4.4 Stars**

US \$44800 Quantity: **1**
[Add to cart](#)

Availability: In Stock
Condition: New
Brand: Mexalvini

[Facebook](#) [Twitter](#) [Stika](#) [Pinterest](#) [LinkedIn](#)

[ZOOM](#)

ДЕТАЛІ **ВІДГУКИ**

Показані 1-2 із 5 записів.

DIMASIK 3:59:21 AM MAY 28, 2022

★ ★ ★ ★ ☆

Такий собі годинник за такі гроші.

ADMIN 3:59:21 AM MAY 28, 2022

★ ★ ★ ★ ★

Чудовий годинник, рекомендую

Перша Попередня **2** 3 НАСТУПНА ОСТАННЯ

FILTER

Категорія
Кварцеві

Назва

Бренд

Стать
Будь яка

Матеріал

Браслет/Ремінець
Браслет/Ремінець


Мінімальна ціна

Максимальна ціна

[Знайти](#)

ПОШУК НЕОБХІДНОГО ГОДИННИКА

Показані 1-2 із 2 записів.




\$4950

ГОДИННИК CASIO GA-100-1A1ER

[Add to cart](#)

[Add to wishlist](#) [Add to compare](#)




\$3520

ГОДИННИК CASIO EF-3160-1AVES

[Add to cart](#)

[Add to wishlist](#) [Add to compare](#)

Кошик

Фото	Назва	Кількість	Ціна	✕
	ГОДИННИК CASIO GA-100-1A1ER	1	4950	✕
Загалом:			1	
На суму:			4950	

[Продовжити покупки](#)
[Оформити замовлення](#)
[Очистити кошик](#)

Висновки

- У процесі виконання дипломного проекту було проведено аналіз предметної області, розглянуто схожі програмні рішення та виявлено їх слабкі та сильні сторони. Спираючись на досліджену предметну область та виявлені переваги і недоліки схожих рішень було визначено вимоги до інтернет-платформи. Далі було розглянуто типи архітектур та архітектурні шаблони а також типи баз даних та визначено, що для розробки інтернет-платформи ідеально підійде архітектурний шаблон MVC та реляційний тип баз даних MySQL. Також з функціональних вимог було визначено сутності предметної області які дозволили спроектувати базу даних та створити діаграму «Сутність-Зв'язок» на якій було відображено схему БД розробленого веб-додатка.
 - На основі спроектованої архітектури, інтерфейсу, модулів та БД було розроблено дипломний проект та проведено модульне та функціональне тестування в результаті якого в розробленій платформі присутні всі функції, які були визначені в технічному завданні і він готовий до використання.
 - В результаті виконання дипломного проекту було розроблено інтернет-платформу для продажу годинників «The World of watches» яка допоможе користувачам підібрати та порекомендувати годинник який зацікавить користувача, та надасть змогу зручно та без складнощів здійснити покупку обраного годинника, отримуючи при цьому електронного листа з квитанцією здійсненого замовлення.
-

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

студента групи _____

Стрільця Д.А.

Прізвище, ініціали

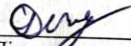
ЗАЯВА

Прошу закріпити за мною тему дипломного проекту освітнього ступеня
«бакалавр» за спеціальністю 121 «Інженерія програмного забезпечення»:

Інтернет-платформа «The World of Watches» для організації продажу
годинників

(керівник дипломного проекту – Онишко Оксана Григорівна)
Прізвище, ім'я, по батькові

01.06.2022
Дата


Підпис студента

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

здобувача вищої освіти

Стрільця Д.А.

Прізвище, ініціали

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.

01.06.2022
дата


підпис

Anti-Plagiarism v-15.257**Максимальне співпадіння з одним документом 4.0%****Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 16%**

ID: 104362 Назва: Інтернет-платформа «The World of Watches» для організації продажу годинників Додано в БД: 2022-06-02 Автора: Д.А. Стрільць Керівники: О.Г. Онишко Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	102294	945	13068 (13%)	146 (15%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра ІПЗ

ID перевірки:
1011425151

Дата перевірки:
02.06.2022 09:42:30 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
02.06.2022 09:48:48 EEST

ID користувача:
100005589

Назва документа: ДП_Стрілець_без дод

Кількість сторінок: 97 Кількість слів: 15596 Кількість символів: 120974 Розмір файлу: 5.31 MB ID файлу: 101130574

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

30.9% Схожість

Найбільша схожість: 6.1% з Інтернет-джерелом (<https://devzone.org.ua/post/nayvazhlivishi-arkhitekturni-shabloni-yaki..>)

25% Джерела з Інтернету

301

Сторінка 99

7.99% Джерела з Бібліотеки

139

Сторінка 102

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

1

Підозріле форматування

28
сторінок

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНИЙ ПРОЕКТ
освітнього ступеня «Бакалавр»

Дипломник Стрілець Дмитро Анатолійович

Тема Інтернет-платформа «The World of Watches» для організації продажу годинників

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг дипломного проекту:

Кількість листів креслень _____; кількість сторінок записки 140

1. Короткий зміст пояснювальної записки та прийнятих рішень У дипломному проекті було досліджено предметну область та визначено постановку задачі, проаналізовано наявне програмно-технічне забезпечення предметної області, та визначено вимоги до програми. На основі отриманих даних було спроектовано веб-сайт, та розроблено відповідно до вимог. Також було проведено тестування веб-сайту та доведено, що розроблена інтернет-платформа працює коректно при будь яких умовах та готовий до експлуатації.

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

3. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі описано предметну область та актуальність обраної теми. У першому розділі проведено аналіз предметної області, аналіз наявного програмно-технічного забезпечення, та визначено вимоги до інтернет-платформи. У другому розділі було розглянуто можливі архітектури веб-додатку та визначено, що веб-сайт буде використовувати клієнт-серверну архітектуру та архітектурний шаблон MVC, також спроектовано базу даних, модулі інтернет-платформи, інтерфейс користувача, та визначено методи та технології які будуть використані для розробки проекту. У третьому розділі було розроблено інтернет-платформу на основі спроектованих даних. У четвертому розділі було проведено тестування різних типів та доведено правильне функціонування інтернет-платформи.

4. Позитивні сторони проекту У проекті реалізовано усі найсильніші сторони схожого програмного забезпечення та виправлено помилки які є у схожих рішеннях. Також реалізовано сучасний адаптивний дизайн, який чудово працює на усіх типах пристроїв.

5. Негативні сторони проекту У проєкті не реалізована можливість оплати годинника у додатку

6. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення розроблене відповідно до тематики дипломного проекту, кожна діаграма або рисунок чітко відображає потрібну інформацію. Пояснювальна записка розроблена згідно усіх вимог, має чітку структуру та наповненість відповідно тематики.

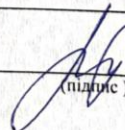
7. Відгук про дипломний проєкт в цілому Дипломний проєкт розроблений відповідно до постановки задачі, для розробки проєкту були використані сучасні технології, які забезпечують необхідну функціональність веб-сайту. Пояснювальна записка має зрозумілу структуру, та чітко викладений матеріал, який стосується теми дипломного проєкту.

8. Інші зауваження _____

9. Оцінка дипломного проєкту Дипломний проєкт виконаний згідно постановленій задачі та заслуговую на оцінку «добре»

РЕЦЕНЗЕНТ завідувач кафедрою Комп'ютерної інженерії та інформаційних систем, доктор технічних наук, професор Говорухенко Т.О.

“27” 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 «The World of Watches» для організації продажу годинників

Автор: Стрілець Дмитро Анатолійович

Спеціальність: 121 – Інженерія програмного забезпечення

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

Науковий керівник: Онишко Оксана Григорівна, кандидат педагогічних 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 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, 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 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, складає 30,9% і адресується до 440 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломної роботи.

Керівник



О. Г. Онишко

Гарант ОП



Л. П. Бедратюк

Завідувач кафедри



Л. П. Бедратюк