

Державний вищий навчальний заклад
“Прикарпатський національний університет імені Василя Стефаника”
Кафедра інформаційних технологій

УДК 004

ДИПЛОМНИЙ ПРОЕКТ

Тема: Розробка Web-сайту інтернет-магазину для продажу етно-одягу

Спеціальність 121 Інженерія програмного забезпечення
код і назва спеціальності

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДП.ПЗ-32.ПЗ

(позначення)

Рецензент

доцент Лазарович І.М.
(посада) (підпис) (дата) (розшифровка підпису)

Студент

ПЗ-41 Остафійчук Т.Д.
(шифр групи) (підпис) (дата) (розшифровка підпису)

Нормоконтролер

доцент Лазарович І.М.
(посада) (підпис) (дата) (розшифровка підпису)

Керівник дипломного проекту

професор Кузь М.В.
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту

Завідувач кафедри

доцент Козленко М.І.
(посада) (підпис) (дата) (розшифровка підпису)

2020

(рік)

Державний вищий навчальний заклад
«Прикарпатський національний університет імені Василя Стефаника»
Факультет математики та інформатики Кафедра інформаційних технологій
Спеціальність Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Завідувач кафедри Козленко М.І.

„_____” _____ 20__ р.

ЗАВДАННЯ НА ВИКОНАННЯ ДИПЛОМНОГО ПРОЕКТУ

Студенту Остафійчук Тетяні Дмитрівні

(прізвище, ім'я, по батькові студента)

1. Тема проекту Розробка Web-сайту інтернет-магазину для продажу етно-одягу

затверджена розпорядженням по факультету математики та інформатики від
„25” жовтня 2019р. № 7

2. Термін здачі студентом закінченого проекту 22 травня 2020р.

3. Вихідні дані до дипломного проекту класифікація інтернет-магазинів,
вимоги до інтернет-магазину, відомості про аналоги, технології розробки –
FuelPHP, ReactJS, MySQL.

4. Зміст пояснювальної записки (перелік питань, що їх належить опрацювати)

1. Загальний огляд інтернет-магазинів

2. Проектування інтернет-магазину

3. Програмна реалізація інтернет-магазину

4. Економічна частина

5. Перелік графічного матеріалу (з точним забезпеченням обов'язкових
креслень) 1. Титульний аркуш. 2. Мета та актуальність дипломної роботи.

3. Основні задачі. 4. Основні функції інтернет-магазину.

5. Аналоги інтернет-магазинів. 6. Структура інтернет-магазину. 7. Діаграма
варіантів використання сайту. 8. Структура бази даних. 9. Результати роботи.

10. Висновки.

6. Дата видачі завдання

11.09.2019

Керівник

(підпис)

Кузь М.В.

(розшифровка підпису)

Завдання прийняв до виконання

(підпис)

Остафійчук Т.Д.

(розшифровка підпису)

КАЛЕНДАРНИЙ ПЛАН

Номер і назва етапів дипломного проекту	Термін виконання етапів проекту	Примітка
1. Загальний огляд інтернет-магазинів	02.12.2019	виконано
2. Проектування інтернет-магазину	15.02.2020	виконано
3. Програмна реалізація інтернет-магазину	17.04.2020	виконано
4. Економічна частина	11.05.2020	виконано
5. Оформлення пояснювальної записки дипломного проекту	18.05.2020	виконано

Студент

Остафійчук Т.Д.

(підпис) (розшифровка підпису)

Керівник проекту

Кузь М.В.

(підпис) (розшифровка підпису)

РЕФЕРАТ

Пояснювальна записка: 97 сторінок (без додатків), 67 рисунків, 8 таблиць, 34 джерел, 7 додатків на 33 сторінках.

Ключові слова: інтернет-магазин, framework ReactJS, framework FuelPHP, MySQL.

Об'єктом дослідження є інтернет-магазин для продажу.

Мета роботи: проектування та розробка інтернет-магазину для продажу етно-одягу.

Стислий опис тексту пояснювальної записки:

У пояснювальній записці наявний огляд аналогів інтернет-магазинів, визначено переваги та недоліки. Також зосереджено увагу на розробці структури та функціоналу сайту. Проведено проектування бази даних. Визначено засоби розробки та реалізовано інтернет-магазин. Здійснено економічне обґрунтування розробки.

ABSTRACT

Explanatory note: 97 pages (without appendix), 67 figures, 8 tables, 34 references, 7 appendix on 33 pages.

Key words: internet-shop, framework ReactJS, framework FuelPHP, MySQL.

Object of study: is an online store for sale.

Brief description of the explanatory note:

The explanatory note provides an overview of the analogues of online stores, identifies the advantages and disadvantages. It also focuses on developing the structure and functionality of the site. The database was designed. Development tools have been identified and an online store has been implemented. The economic substantiation of development is carried out.

Зміст

ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	8
ВСТУП.....	9
1 ЗАГАЛЬНИЙ ОГЛЯД ІНТЕРНЕТ-МАГАЗИНІВ	11
1.1 Інтернет-магазин та його особливості	11
1.2 Класифікація інтернет-магазинів. Переваги та недоліки.....	12
1.3 Огляд інтернет-магазинів	16
1.4 Постановка задачі	20
2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ	22
2.1 Структура побудови інтернет-магазину	22
2.2 Моделювання основних принципів роботи web-сайту.....	27
2.3 Проектування структури бази даних	38
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ	44
3.1 Вибір та огляд програмних засобів реалізації проекту.....	44
3.2 Програмна реалізація бази даних.....	49
3.3 Розробка основних функцій магазину	53
3.4 Опис інструкції користувача	60
4 ЕКОНОМІЧНА ЧАСТИНА.....	72
4.1 Побудова мережевого графа та розрахунок його параметрів	72
4.2 Економічне обґрунтування розробки та впровадження проектного рішення	77
4.2.1 Розрахунок витрат на розробку та впровадження проектного рішення ..	77

					ДП.ІПЗ-32.ІЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	«Розробка Web-сайту інтернет-магазину для продажу етно-одягу»	Літ.	Аркуш	Аркуші
Розроб.		Остафійчук Т.Д.				н	6	97
Перев.		Кузь М. В.				ПНУ ІПЗ-41		
Н. контр.		Лазарович І. М.						
Затверд.		Козленко М. І.						

4.2.2	Обрахунки експлуатаційних витрат	86
4.2.3	Розрахунок ціни використання проектного рішення	89
4.2.4	Визначення показників економічної ефективності.....	91
ВИСНОВКИ		94
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		95
ДОДАТОК А		98
ДОДАТОК Б.....		99
ДОДАТОК В.....		100
ДОДАТОК Г		101
ДОДАТОК Д.....		110
ДОДАТОК Е.....		120
ДОДАТОК Ж.....		127

					ДП.ІПЗ-32.ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

Web – система доступу до пов'язаних між собою документів на різних комп'ютерах, підключених до Інтернету.

TIC – торгова Інтернет-система.

ЕОМ – електронно-обчислювальна машина.

ПК – персональний комп'ютер.

ПЗ – програмне забезпечення.

ЗМІ – засоби масової інформації.

UML – Unified Modeling Language.

RUP – Rational Unified Process.

БД – база даних.

СУБД – система управління базами даних.

JSX – JavaScript XML.

					ДП.ІПЗ-32.ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

З моменту свого заснування в 1989 році Всесвітня павутина торкнулася життя мільярдів людей по всьому світу і принципово змінила те, як зв'язуємося з іншими, характер роботи, як відкриваємо та ділимося новинами та новими ідеями, як розважаємося і як формуються та функціонують громади.

Мережа, як і Інтернет, створена таким чином, щоб створити бажаний ефект «від початку до кінця», при цьому приховуючи наскільки це можливо проміжну техніку, яка змушує її працювати.

Мережа стала публічною площею, бібліотекою, лікарем, магазином, школою, дизайнерською студією, офісом, кінотеатром, банком тощо. Звичайно, з кожною новою функцією, кожним новим веб-сайтом, розподілом між тими, хто перебуває в Інтернеті, і тими, хто не збільшується, що робить ще більш необхідним зробити Інтернет доступним для всіх.

Інтернет-магазини зробили революцію в діловому світі, зробивши все, що хто хотів, доступним простим клацанням кнопки миші. Інтернет-магазини за останні роки набули популярності, головним чином тому, що люди вважають зручним і легким торгувати на веб-платформах, не виходячи з дому чи офісу. Один з найбільш привабливих факторів щодо онлайн-покупок, особливо в період свят чи карантину, – це полегшення необхідності чекати в довгих чергах або шукати від магазину до магазину певний предмет. Процес прийняття рішень покупця за останні роки різко змінився. Покупці проводять великі дослідження в Інтернеті, перш ніж коли-небудь звернутися за консультацією до продавця. Інтернет робить бізнес набагато простішим та швидшим. Це призвело до змін у тому, як люди ведуть бізнес із швидко зростаючою світовою тенденцією до інтернет-магазинів чи електронної комерції.

Інтернет-магазин – це тип веб-сайту, який має функціонал, необхідний для розміщення та обробки замовлень в Інтернеті. Створення веб-сайту означає,

					ДП.ІПЗ-32.ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

що розміщення бізнесу або особистої інформації на веб-сайті, доступному для всіх, хто має Інтернет.

Все вище сказане є свідченням актуальності питання дослідження інтернет-магазину та його розробки з використанням сучасних Internet-технологій. В рамках дипломного проекту було поставлено завдання розробити Web-сайт «інтернет-магазин» для продажу етно-одягу.

Мета роботи. Розробка інтернет-магазину для продажу етно-одягу.

Новизна роботи. В процесі виконання дипломного проекту розроблено вебсайт для продажу етно-одягу з можливістю визначення точного розміру онлайн. Вибір предметної області зумовлений актуальністю веб-технологій, областю їх застосування та стрімким розвитком інтернет-представництв онлайн бізнесу.

Завдання на дипломну роботу:

- здійснити пошук інформації по тематиці дипломної роботи, виконати її аналіз та дослідження;
- ознайомитись з особливостями функціонування інтернет-магазинів, його позитивними та негативними якостями, та визначити їх класифікацію;
- розглянути структуру побудови та сучасні програмні засоби по розробці інтернет-магазинів;
- проаналізувати маркетингове дослідження та розглянути економічні передумови створення інтернет-магазину;
- розробити вебсайт, базу даних «інтернет-магазину» та метод який надасть можливість визначити розмір та зробити оцінку щодо параметрів тіла;
- протестувати роботу вебсайт.

					ДП.ІПЗ-32.ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНИЙ ОГЛЯД ІНТЕРНЕТ-МАГАЗИНІВ

1.1 Інтернет-магазин та його особливості

Через швидке зростання технологій бізнес-організації перейшли від традиційного методу продажу товарів до електронного способу продажу товарів. Бізнес-організації використовують Інтернет як основний засіб для здійснення комерційних операцій.

Інтернет-магазин – це сервіс, який дозволяє здійснювати покупки в інтернеті, є одним з найпопулярніших видів онлайн торгівлі у сфері B2C (business to customer) - продаж товару або послуг фізичним особам [1].

Інтернет-магазини не мають обмежень у просторі, і на web-сайтах може відображатися широка різноманітність товарів. Це допомагає покупцям-аналітикам придбати товар після хорошого пошуку.

Інтернет-покупки – це форма електронної комерції, яка дозволяє споживачам безпосередньо купувати товари чи послуги у продавця через інтернет за допомогою web-браузера. Споживачі знаходять товар, що цікавить, безпосередньо відвідуючи web-сайт роздрібною торгівлі або здійснюючи пошук серед альтернативних постачальників за допомогою торгової пошукової системи, яка відображає доступність та ціни одного товару у різних електронних торгових мереж [2].

Головною особливістю інтернет-магазинів є те, що вони можуть запропонувати клієнту набагато більшу кількість товарів чи послуг, а ніж offline магазини і надавати користувачам достатньо великий обсяг інформації, який буде необхідний для того щоб прийняти рішення чи необхідно здійснити покупку. При використанні інтернет-технологій є можливість персоналізації інтернет-магазину як шлях до підвищення продажів, тобто вміти запропонувати правильний товар у необхідний момент відповідному споживачеві.

					ДП.ІПЗ-32.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

Інтернет-магазини не потребують суттєво великих витрат на утримання та організацію роботи, тому що необхідна досить невелика кількість працівників та матеріальна база (приміщення для роботи).

А також інтернет-магазини містять недоліки. Найбільш поширеними є види шахрайства при здійсненні грошових транзакцій та комісії платіжних систем, можливі проблеми з службами доставки, висока конкуренція на ринку та спори з клієнтами на рахунок чи відповідає товар основним параметрам якості.

Споживачі інтернет-магазинів ставлять до них ряд основних вимог, серед них такі як:

- доступно зрозумілий інтерфейс для будь-якого користувача та зручна навігаційна панель на сайті;
- отримання необхідної інформації для користувача за невеликий обсяг часу;
- найменша кількість здійснених операцій користувачем при замовленні товару.

1.2 Класифікація інтернет-магазинів. Переваги та недоліки

Існує кілька систем класифікації інтернет-магазинів [3]:

- за методом роздрібного продажу товарів у мережі: інтернет-магазини; веб-вітрини, торгові системи; торгові ряди; контентні проекти (споживацькі енциклопедії, системи інтернет -замовлень товарів тощо);
- за бізнес-моделлю: повністю онлайн-магазин та поєднання офлайн-бізнесу з онлайн-бізнесом (коли інтернет-магазин створюється на основі вже діючої реальної торгової структури);
- за взаємовідносинами з постачальниками: магазини, які володіють власним складом (наявність реальних товарних запасів);

						ДП.ІПЗ-32.ПЗ	Арк.
							12
Зм.	Арк.	№ докум.	Підпис	Дата			

- магазини, що працюють за договорами з постачальниками (відсутність значних товарних запасів);
- за ступенем автоматизації серед торгових систем електронних магазинів розрізняють веб-вітрини, власне інтернет-магазини та торгові інтернет-системи [3].

Характерною рисою інтернет-магазину є повна автоматизація системи обробки замовлень, завдяки чому можна працювати індивідуально з кожним зареєстрованим клієнтом. Спільною рисою для та інтернет-магазину та ТІС є можливість здійснювати повний торговий цикл у режимі підключення до мережі. При цьому ТІС додатково інтегрована в систему внутрішнього документообігу компанії. Неавтоматизованими для інтернет-магазину та ТІС залишаються системи доставки товару. Приблизна структура основних форм організації інтернет-магазинів представлена на рис. 1.1 [3].

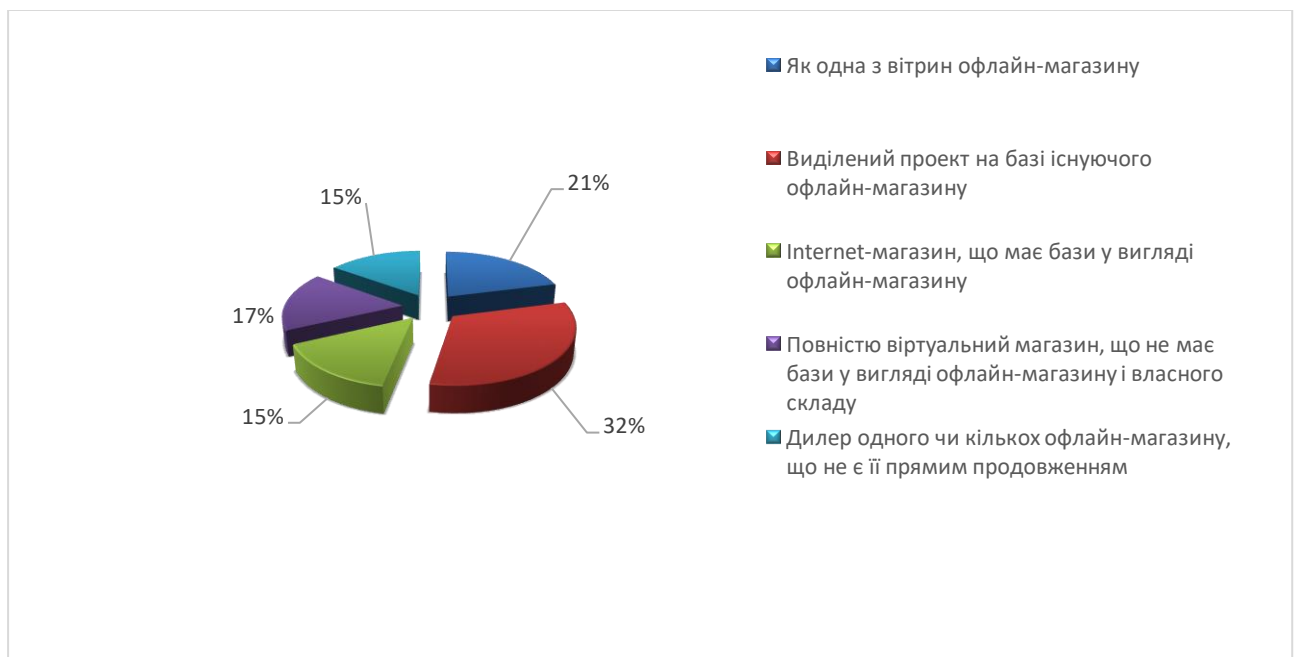


Рисунок 1.1 – Форми організації інтернет-магазинів в українському сегменті мережі Інтернет

На основі здійснення аналізу електронної комерції , було визначено основні причини здійснення покупок через Інтернет:

					ДП.ІПЗ-32.ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

- економить час та зусилля;
- зручність покупки вдома;
- доступна широка різноманітність / асортимент продукції;
- хороші знижки / нижчі ціни;
- отримайте детальну інформацію про товар;
- можливість порівняти різні моделі / марки.

Покупки в інтернет-магазинах надають ряд переваг:

- зручність. Інтернет-магазини дають можливість робити покупки коли завгодно, де завгодно, і це доступно 24/7;
- кращі ціни. Товари, який знаходяться в Інтернеті, мають кращі ціни. Це тому, що вони надходять безпосередньо від продавця, і посередника немає. Це також дає можливість порівняти ціни, щоб знайти вигіднішу пропозицію, і багато web-сайтів також пропонують купони зі знижкою та знижки, якщо здійснюємо покупку безпосередньо у них. Найкраще, що покупець економить кошти на податку. Інтернет-магазини збирають лише податок з продажу;
- більше різноманіття. Широкий вибір продукції в інтернеті. Є можливість знайти майже будь-яку марку чи товар, який шукається. Можна ввійти в останні світові тенденції, не витрачаючи гроші на авіа перельоти. Здійснення покупки в області роздрібної торгівлі в інших частинах держави, країни чи навіть світу, а не обмежуватись власною державою. Деякі інтернет-магазини приймають замовлення на товари, що не продаються, і відправляють їх, коли товар є в наявності. Іноді доставка безкоштовна, залежно від вашого географічного розташування;
- легке порівняння цін. Порівнювати та досліджувати продукти та їх ціни в Інтернеті набагато простіше. Наприклад якщо здійснювати покупку техніки, можна знайти відгуки споживачів та порівняння товарів для всіх варіантів на ринку із посиланнями на найкращі ціни. Також є

					ДП.ІПЗ-32.ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

- покупці в Інтернеті не можуть скористатися сезонними торговими заходами, що не є загальнодержавними.

Загальні ризики ведення електронної комерції, яких потрібно остерігатися:

- безпека в Інтернеті;
- надійність системи;
- питання конфіденційності;
- шахрайство з кредитними картками;
- проблеми з доставкою товару;
- недоліки в програмному забезпеченні;
- є деякі проблеми в залученні покупців.

1.3 Огляд інтернет-магазинів

Одним з найпопулярніших сайтів по продажі етно-одягу є – «ETNODIM» [4]. Даний інтернет-магазин має ненав'язливий дизайн. Для зручнішого використання є можливість фільтрації товарів за багатьма параметрами.

Сторінка про товар містить декілька фотографій з хорошою якістю та короткий опис, ціну (рис. 1.2).

					ДП.ІПЗ-32.ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

Гарантія



Обмін або повернення впродовж 31 дня



При кур'єрській доставці по Києву, даємо декілька моделей/розмірів на вибір



Доставляємо в день замовлення

Рисунок 1.4 – Гарантія

Також, одним із популярних інтернет-магазинів по продажі вишивки є магазин «ЕТНО-СІТІ» [5]. Сайт містить зручну та зрозумілу навігацію для користувача, дизайн сайту є приємним до людського ока. Для швидкого обрання необхідного товару є пошук.

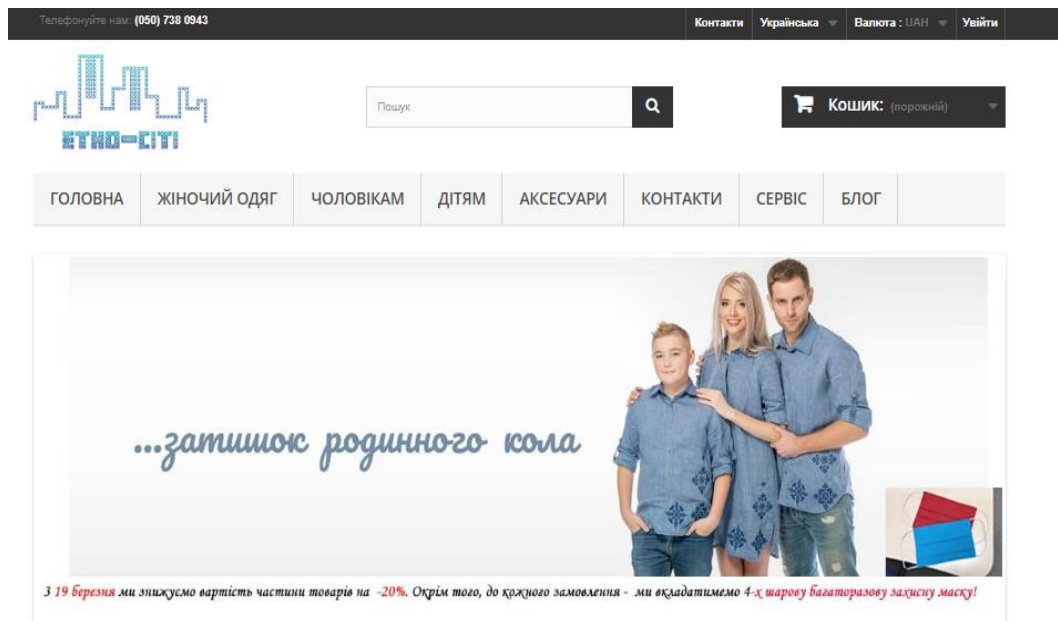


Рисунок 1.5 – Головна сторінка «ЕТНО-СІТІ»

Сторінка про товар містить декілька фотографій, ціну, додати виріб до списку обраних товарів, поле для обрання розміру, а також поле для того щоб вказати кількість одиниць при замовленні товару (рис. 1.6).

Для більш детальної інформації про виріб, яка надається користувачеві - характеристика товару, крім того є можливість залишити відгук про товар, а також є онлайн підтримка за допомогою якої можна здійснити консультацію.

					ДП.ІПЗ-32.ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

1.4 Постановка задачі

Основною задачею дипломної роботи є розроблення інтернет-магазину для продажу етно-одягу.

Об'єктом дослідження є інтернет-магазин етно-одягу.

Завдання дослідження – розробка структури для функціонування інтернет-магазину

Предметом дослідження є засоби та методи програмної реалізації Web-сайтів – інтернет-магазинів, з метою врахування ймовірно очікуваної аудиторії:

- доступність, простоту подачі;
- швидке завантаження інформації;
- логічне розташування елементів;
- ефективно вирішення актуальних проблем;
- відсутність складнощів в процесі здійснення яких-небудь дій.

Мета роботи – здійснити аналіз електронної комерції, а саме проаналізувати діяльність інтернет-магазину, їх відмінності від звичайних магазинів, розглянути аналоги на ринку та розробити Інтернет-магазин «Вишиванка» для продажу етно-одягу.

Результатом даного проекту є інтернет-магазин, який за своєю практичністю відзначається, як засіб електронної комерції реального магазину.

Необхідно виконати ряд завдань проекту, для того щоб досягти поставленої мети:

- проектування та розробка структури інтернет-магазину як з сторони клієнта так і менеджера – адміністратора;
- проектування та розробка основних UML діаграм, які дозволять визначити основні класи, атрибути, методи та зобразити процеси роботи сайту;

					ДП.ІПЗ-32.ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

- розробка основних модулів магазину, наприклад таких, як: перегляд товару, віджет визначення розміру, додавання в кошик, оформлення замовлення, форму зв'язку, а також адміністративної панелі для кращого управління;
- створення необхідної кількості категорій та інформаційних блоків;
- розроблення інтерфейсу користувача;
- наповнення магазину контентом;
- розрахунок витрат на розробку та впровадження проектного рішення.

					ДП.ІПЗ-32.ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ

2.1 Структура побудови інтернет-магазину

В наш час існує безліч онлайн платформ для електронної комерції, відповідно онлайн клієнти мають різноманітний вибір Інтернет ресурсів. Обравши один з них клієнт не буде довго затримуватися на ньому в очікуванні відповіді чи від сервера, чи від менеджера, а скоріш за все просто закрий вкладку і обере більш зручніший та швидкісний для себе веб-ресурс для купівлі товару. Тому чим більше часу клієнт заощадить при роботі з сайтом, тим є більша ймовірність того що він стане саме покупцем даного магазину.

Не менш важливим є те щоб покупцеві було досить практично працювати з магазином, а для цього потрібно спроектувати зручну структуру. Це свого роду є «каркас», який дає зрозуміти що саме повинно бути на сайті для зручного використання. Наприклад, якщо розглядати веб-ресурс на якому розміщена корисна інформація для користувача, але текст не буде мати жодного оформлення, яке приваблювало б людське око, то з ймовірністю у сто відсотків користувач відмовиться використовувати даний портал. Такий самий випадок може трапитися з Інтернет-магазинами, клієнт відмовиться використовувати ресурс, якщо довго буде виконувати пошук необхідних йому даних. Отже можна зробити висновок, що розробка структури інтернет-магазину є одним із важливих етапів.

Структура сайту - продукт ретельного мислення, навмисного дизайну та чіткої організації.

Структура сайту – це є побудова деревоподібного розподілу блоків сайту, які дають його зручну та логічну структуру [6]. Якщо брати більш точніше до магазину то це є схематичне зображення розміщення блоків з категорією,

					ДП.ІПЗ-32.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

Більшість складних web-сайтів можуть бути організовані як логічна послідовність, але кожна сторінка в послідовності може містити посилання на одну або кілька сторінок, а також посилання на інформацію на інших web-сайтах. Приклад такого типу структури - це коли користувач намагається придбати щось або проходить курс онлайн. Послідовні структури припускають, що існує деяке оптимальне впорядкування вмісту, яке пов'язане з більшою ефективністю чи успіхом [7].

Ієрархічна структура. Даний тип структур є найбільш поширений, а також найкращий спосіб організації найскладніших інформаційних масивів. Оскільки вебсайти зазвичай організовані навколо однієї домашньої сторінки, яка потім посилається на сторінки підтеми меню, ієрархічна архітектура особливо підходить для організації вебсайтів. Найпростіша форма ієрархічної структури сайту - це зірка або набір сторінок, які розходяться від головної домашньої сторінки. Сайт по суті є однорівневою ієрархією. Навігація є простим списком підсторінок, а також посиланням на домашню сторінку [7].

Багато користувачів знайомі з структуруванням інформації в ієрархіях, оскільки вони щоденно бачать ці структури в тому, як у підприємств формується їх керівна структура керівництва, спосіб створення планів проектів тощо. Більшість web-сайтів приймають певну форму деревоподібної архітектури. Таке розташування основних категорій і підкатегорій має досить велику перевагу для складної організації сайтів, оскільки більшість людей знайомі з ієрархічними організаціями і можуть легко формувати ментальні моделі структури сайту [7].

Мережева структура. Такий вид архітектури полягає у побудові системи навігаційної мапи сайту, коли між всіма елементами існує зв'язок та можливість зручного та швидкого переходу між сторінками сайту без додаткового завантаження проміжних сторінок. Даний тип структур також є поширений на просторах Інтернету, одним з прикладів такої структури є сайти – великі інформаційні портали [7].

					ДП.ІПЗ-32.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

Комбінована структура. Даний тип структури – поєднання двох різних видів структур. Наприклад, сайт може мати спроектовану ієрархічну структуру але при проектуванні деяких сторінок використати дану структуру буде не логічно, тому можна такий процес замінити, обравши іншу структуру, яка буде більш доречною, наприклад лінійну структуру. Але в більшості випадків сайти з такою структурою здаються абсолютно неорганізованими. Використовуючи такий сайт, можна помітити що здійснювати переходи між сторінками можна цілком у різний спосіб, і не є обов'язковим повертатися до бажаної сторінки по однаковому шляху [7].

Отже, здійснивши аналіз типів структур сайтів, було обрано комбіновану структуру, так як можна створити просту та зручну для відвідувача структуру. На рис. 2.1 спроектована структура інтернет-магазину для продажу етно-одягу.

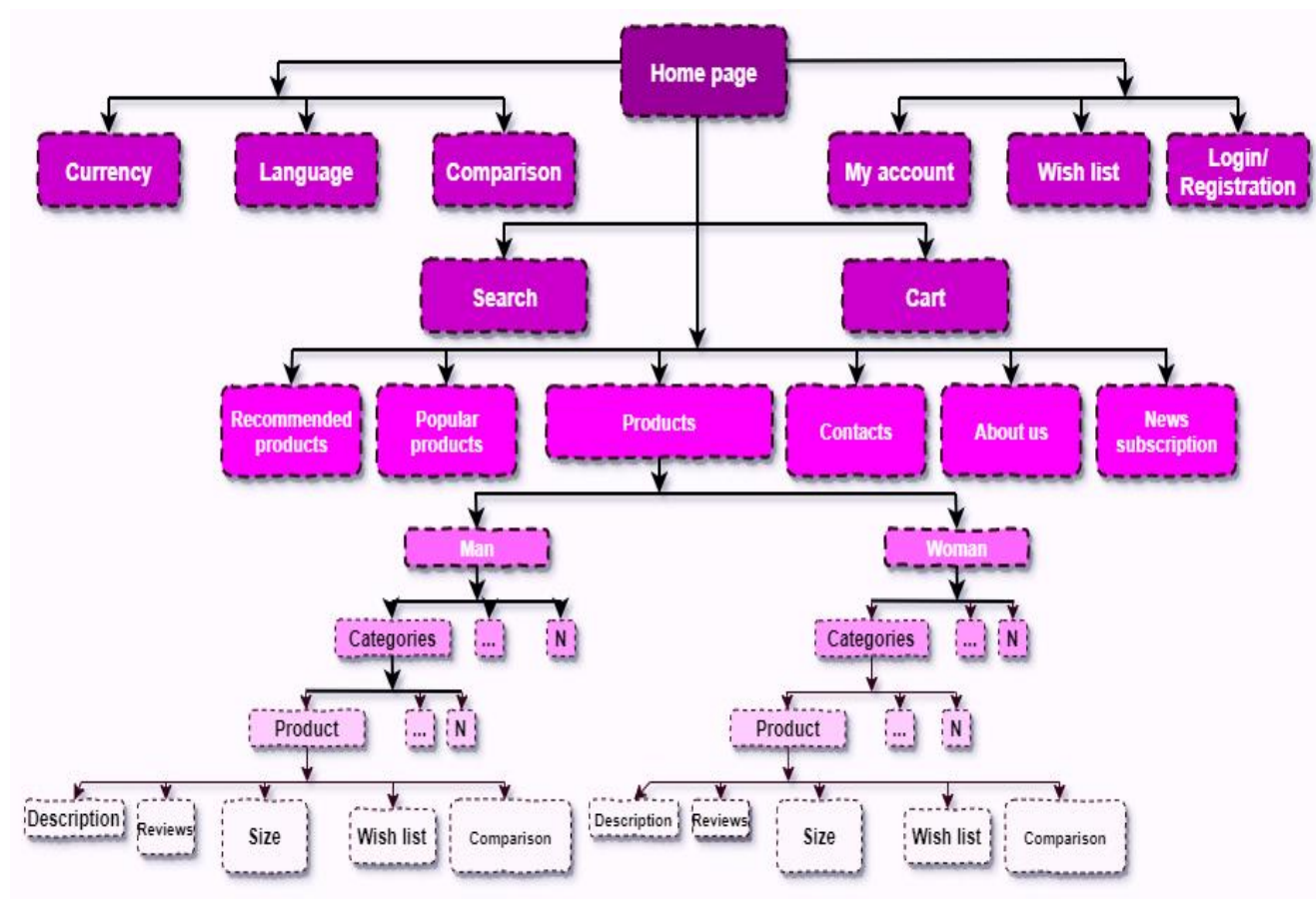


Рисунок 2.1 – Проектування структури Інтернет магазину для покупця

Головна сторінка є невід'ємною складовою сайту, тому її розробці приділяють особливу увагу. В більшості випадків таку сторінку забезпечують адаптованою навігацією по сайті, а також надають користувачеві різноманітні зручні функції для роботи. Проте одним із важливих запитів до головної сторінки – розміщення корисного та інформативного контенту у графічній та текстовій формі.

Сайт охоплює основні елементи – хедер, навігаційну панель, основний інформаційний блок та футер. Верхня частина головної сторінки сайту містить хедер це так звана «шапка», яка в більшій мірі дублюється на всіх решта сторінках. Переважно за допомогою цього блоку користувач розпізнає веб-ресурс, адже в цій частині розміщують назву магазину – логотип, який а також найбільш важливі компоненти сайту, наприклад для інтернет-магазину – зміна мови, валюти, порівняння товару, список побажань, корзина, вхід/реєстрація та особистий кабінет.

Наступним не менш важливим елементом є навігаційна панель, яка містить гіперпосилання на основні розділи сайту. Горизонтальна навігаційна панель є найбільш поширеним різновидом розташування головного меню. Адже меню – найважливіший елемент сторінки сайту, на який користувач достатньо звертає увагу, тому меню повинне бути помітним, зручним та зрозумілим, бо в іншому випадку користувач не зрозуміє, як користуватися ресурсом для того щоб потрапити в потрібний розділ, а просто покине сайт.

Основний блок інформації містить в більшій мірі контент, який є складовою для будь якої електронної комерції. Велику роль відіграє розроблення банерів для реклами, тому що це підвищить відсоток продажу в Інтернеті.

Наступним елементом є футер, який розміщується в нижній частині сайту. Та містить інформацію про авторські права, контактну інформацію, включає посилання на соціальні мережі, підписка на розсилку новин, а також на самі основні розділи сайту.

					ДП.ІПЗ-32.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

документування компонентів програмного забезпечення, бізнес-процесів та інших систем. Мова UML заснована на деяких базових поняттях, які можуть бути вивчені й застосовані більшістю програмістів і розробників, з об'єктно-орієнтованого аналізу і проектування [8].

UML – це стандартна мова моделювання, а не процес розробки програмного забезпечення. Дане твердження було підтверджено за специфікацією UML, адже процес проектування пояснений, як:

- надає вказівки щодо порядку діяльності команди;
- визначає, які артефакти слід розробляти;
- спрямовує завдання окремих розробників та команди в цілому;
- пропонує критерії моніторингу та вимірювання продукції та діяльності проекту;
- служить, як спосіб для вирішення задач об'єктно-орієнтованого моделювання систем [8].

UML є незалежним від процесу та може бути застосована у контексті різних процесів. Тим не менш, найбільш підходить для використання залежних від конкретних випадків, ітераційних та поступових процесів розвитку. Прикладом такого процесу є Rational Unified Process (RUP).

Здебільшого проектування UML діаграм застосовуються на всіх стадіях життєвого циклу бізнес-процесів та розроблення прикладних програм. Графічна мова моделювання підтримує різні різновиди діаграм, а також містить найрізноманітніший набір можливостей описування та представлення різних аспектів складної системи, це все робить дану мову універсальним засобом описування програмних проектів. UML може бути застосований до різних областей застосувань (наприклад, банківська справа, фінанси, Інтернет, аерокосмічна, охорона здоров'я тощо) [33].

UML – діаграма, являє собою часткове графічне представлення (вид) моделі системи при проектуванні, реалізації або вже існуючої. Діаграма UML містить графічні елементи (символи) – вузли UML, пов'язані з ребрами (також

					ДП.ІПЗ-32.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

Таблиця 2.1 – Опис класів інтернет-магазину

Параметр	Значення
Клас Admin	
Коментар	Клас, що є адміністратором інтернет-магазину
Атрибути	username – ім'я адміністратора password – пароль для входу email – поштова адреса адміністратора для входу
Операції	addToCategories() – додавання нової категорії role change() – зміна ролі користувача, тобто надання прав менеджера, тощо. connection with store() – перегляд форми «Зв'язатися з нами»
Клас Users	
Коментар	Клас, що є користувачем та клієнтом інтернет-магазину
Атрибути	username – ім'я покупця password – пароль для входу в особистий кабінет email – поштова скринька для входу
Операції	change password() – зміна паролю
Клас Email news	
Коментар	Клас, що відповідає за розсилку новин
Атрибути	email – поштова адреса
Операції	addEmail() – додавання поштової скриньки
Клас Orders	
Коментар	Клас, що є замовленням, яке здійснює покупець
Атрибути	id_order – код замовлення username – ім'я клієнта email – поштова скринька клієнта method – метод доставик city_name – назва міста status – статус доставки замовлення price – загальна сума покупки
Операції	addToOrders() – здійснити замовлення ViewOrder() – перегляд замовлення EditOrder() – редагування замовлення DeleteOrder() – видалення замовлення
Клас Authorization	
Коментар	Клас, що відповідає за авторизацію користувача
Атрибути	login – поштова адреса password – пароль для входу в обліковий запис
Операції	Authorization() – метод авторизації
Клас Registration	
Коментар	Клас, який відповідає за реєстрацію користувача

Продовження таблиці 2.1

Атрибути	email – пошта для реєстрації password – пароль username – ім'я користувача
Операції	Registration() – метод реєстрації
Клас Cart	
Коментар	Клас, що є корзиною
Атрибути	product_id – код продукту
Операції	Delete product() – видалення товару з корзини Add product() – додавання товару в корзину Update cart() – оновлення товару в корзині View content() – перегляд вмісту Calculation_amount() – розрахунок суми до оплати
Клас Products	
Коментар	Клас, який відповідає за товар
Атрибути	categories_id – код категорії product_id – код товару parameters – параметри title – назва продукту old_price – стара ціна товару new_price – нова ціна товару sizes – розміри images – фотографія
Операції	AddToComment() – додати коментар до товару addToProduct() – додати товар deleteProduct() – видалити товар editProduct() – редагування товару
Клас Categories	
Коментар	Клас, що відповідає за категорії
Атрибути	categories – категорії
Операції	ViewCategories() – вигляд категорії
Клас Product filter	
Коментар	Клас, який відповідає за фільтрацію товару за параметрами
Атрибути	price – ціна size – розмір
Операції	Filter by price ()– фільтр по ціні Filter by size() – фільтр по розміру
Клас Product search	
Коментар	Клас, який відповідає за пошук необхідного товару
Атрибути	title_product – назва продукту
Операції	Product search() – пошук товарів
Клас Wish list	
Коментар	Клас, який відповідає за надання можливості порівняти товар

- створити обліковий запис (особистий кабінет).

У можливості зареєстрованого користувача – Customer включаються всі можливості не зареєстрованого користувача, а також додаються нові функції:

- авторизація;
- редагування профілю;
- список побажань;
- порівняльна характеристика товару;
- управління корзиною.

Розглянемо одну із функцій – управління корзиною. До функції управління корзиною входять ще декілька функцій, а саме: додати товар у корзину, управління корзиною, переглянути корзину, очистити корзину та оформити замовлення. Додати виріб у корзину можна з декількох сторінок, як з головної так і з сторінки з детальною інформацією про товар, натиснувши на кнопку «додати до корзини». Під управлінням корзиною розуміється можливість її оновлення, та видалення непотрібного товару з корзини, а також додати виріб до списку побажань.

Адміністратор (менеджер) – Administrator має всі функції зареєстрованого користувача, а також додаються нові функції:

- керування контентом;
- керування замовленнями.

Розглянемо функцію керування контентом. До даної функції входять ще декілька, а саме: аналіз статистики, додавання категорії, додавання товару, керування користувачами, видалення та перегляд категорії, товару, перегляд зв'язку.

Перейдемо до розгляду функції керування замовленнями. Під керуванням замовленнями розуміється можливість видалення замовлення, якщо те відпрацьоване чи некоректне, зміна статусу та перегляду замовлень.

					ДП.ІПЗ-32.ІЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

2.3 Проектування структури бази даних

Щоб організувати чітку роботу з даними необхідно спроектувати базу даних, для того необхідно здійснити аналіз предметної області розглянути всі можливі проблеми та на основі цього розробити концептуальну модель бази даних.

Концептуальне проектування – на даному етапі проводиться збір, аналіз та формуються вимоги до даних. Тобто здійснюється аналіз предметної області, визначаються найважливіші сутності (таблиці) і атрибути та зв'язки між ними. В результаті одержуємо концептуальну модель, яка є симетрична до структури бази даних. Переважно дана модель представлена у вигляді моделі «сутність - зв'язок». Дана модель містить таблиці, їх структуру та логічні зв'язки. Таблиці складаються з стовпців, атрибутів – назва, яких не повинна повторюватися, а необхідною вимогою є унікальність для певного типу сутностей, ключів таблиць, а також типів даних [12, 30].

Предметна область інтернет-магазину «Вишиванка» являє собою комерційний портал для продажу етно-одягу, яка спрямована на те щоб користувач легко міг знайти необхідний йому товар. Інтернет-магазин є ланкою, яка забезпечує зв'язок між продавцем, менеджером – власником магазину і їх клієнтами – покупцями. Тому для швидкої та оптимальної взаємодії обох сторін, існує необхідність автоматизації інформаційних процесів, що призведе до їх швидкої, злагодженої та якісної роботи.

Проектування структури даної предметної області призначене для введення обліку всіх користувачів та їх вподобання, а також для зберігання інформації про товар. У магазині електронної комерції є основне завдання «змушувати» користувача переміщати товари до кошика замовлень, тому на першому етапі проектування бази даних, розглядається три основні частини:

					ДП.ІПЗ-32.ІЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

«username» – ім'я покупця, «password», «profile_fields» – це поля, які присвоюються користувачеві, «last_login» – дата останньої авторизації у форматі UNIX, «login_hash» – хеш авторизації. Ключовим атрибутом даної сутності є «Код клієнта».

Сутність «discount_cards» – визначає набір атрибутів, які містять інформацію про дисконтну картку. Основними атрибутами даної сутності є «id» – порядковий номер коду картки, «user_id» – порядковий номер користувача, «code» – унікальний код картки, «percent» – значення знижки дисконтної картки у вигляді відсотків.

Поле «user_id» – виступає як вторинний ключ, який служить для встановлення зв'язку «один до одного» між сутністю «discount_cards» і сутністю «users»: одна дисконтна картка може бути задіяна до одного покупця (рис. 2.8).

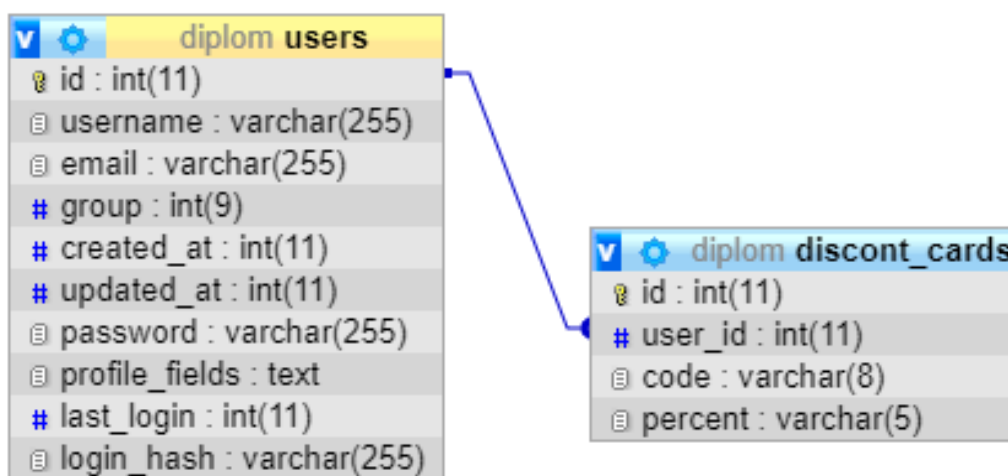


Рисунок 2.8 – ER-діаграма екземплярів сутностей «users», «discount_cards»

Сутність «products» – містить інформацію про продукти і має ряд атрибутів: «id» – код продукту, «category_id» – порядковий номер відповідної категорії, «title» – назва товару, «old_price» – стара ціна товару, «new_price» – нова ціна товару, «sizes» – масив розмірів товару, «description» – опис товару. «Код товару» – є ключовим атрибутом.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ

3.1 Вибір та огляд програмних засобів реалізації проекту

Вибір типу архітектури веб-додатків та складової моделі є одним із найважливіших, але складних у розробці веб-додатків. Архітектура веб-додатків – це схема взаємодії між компонентами веб-додатків. Спосіб планування цієї взаємодії визначає стійкість, ефективність та безпеку майбутнього веб-додатка [13].

Архітектура інтернет-магазину Етно-одягу, базується на трирівневій архітектурі – це архітектура веб-додатків, яка широко використовується у всьому світі. В основному вона містить 3 рівня: Клієнт, Сервер та База даних [29].

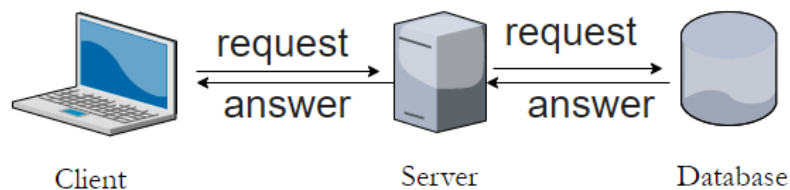


Рисунок 3.1 – Архітектура «Клієнт-сервер» інтернет-магазину

В архітектурі клієнт/сервер відносини комп'ютерів розділені на дві ролі:

- Клієнт, який вимагає конкретних послуг або ресурсів.
- Сервер, який призначений для виконання запитів, відповідаючи (або намагаючись відповісти) на запитувані послуги чи ресурси.

Його особливість полягає в тому, що сам веб-додаток розташований і виконується на сервері, тоді як клієнт отримує лише результати своєї роботи. Робота такої програми заснована на отриманні запитів від користувача,

					ДП.ІПЗ-32.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

передавати на сервер всі команди. В якості мови роботи з БД використовується широко відомий SQL [19].

Особливості використання phpMyAdmin:

- простий спосіб створення таблиць: за допомогою інтерфейсу користувача чи за допомогою SQL-запиту;
- перегляд таблиць;
- підказки при обробці SQL і підсвічування синтаксису;
- зручний і зрозумілий імпорт і експорт БД, таблиць і записів;
- підказки при обробці SQL і підсвічування синтаксису;
- відображення структури таблиці (імена полів таблиці, її атрибути, типи, відношення);
- легкий спосіб занесення даних: за допомогою інтерфейсу користувача чи за допомогою SQL-запиту;
- зручно реалізовано набір функцій, таких, як: функція пошуку, створення, редагування, видалення, які відносяться до рядків таблиць [30].

Для того, щоб створити таблицю використовують SQL-запити, при створенні яких необхідно вказати ім'я таблиці, імена і типи даних для кожного стовпця.

Наприклад, розглянемо створення однієї із таблиць бази даних – таблицю «Products» (рис.3.2).

```
CREATE TABLE `diplom`.`products` (  
  `id` int(11) NOT NULL,  
  `views` int(11) NOT NULL DEFAULT 0,  
  `category_id` int(11) DEFAULT NULL,  
  `parameters` text NOT NULL DEFAULT '[]',  
  `title` varchar(255) NOT NULL,  
  `old_price` int(11) DEFAULT NULL,  
  `new_price` int(11) NOT NULL,  
  `sizes` varchar(255) DEFAULT NULL,  
  `description` text DEFAULT NULL,  
  `colors` varchar(255) DEFAULT NULL,  
  `updated_at` int(11) NOT NULL,  
  `created_at` int(11) NOT NULL,  
  PRIMARY KEY (`id`);  
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Рисунок 3.2 – Створення таблиці бази даних за допомогою SQL-запиту

										ДП.ІПЗ-32.ПЗ	Арк.
											50
Зм.	Арк.	№ докум.	Підпис	Дата							

FuelPHP. Розглянемо основні функції із сторони клієнта і з сторони сервера та їх взаємодію між собою.

На рисунку 3.8 зображено фрагмент коду реалізації методу авторизації на стороні клієнта. Використано асинхронну функцію авторизації – `async()`, для отримання успішного/ не успішного результату операції. Наступним етапом іде перевірка умови на введення email та паролю. Якщо ці поля заповнені то переходимо до перевірки чи є такий користувач у БД, в іншому випадку виводимо повідомлення про те що необхідно заповнити всі поля. Якщо поля заповнені коректно то відповідно відправляється POST запит на сервер з відповідними полями. На рисунку 3.9 функцію перевірки даних при авторизації, здійснюється перевірка умови, якщо логін і пароль співпадають то повертається список який містить usertoken, в іншому випадку повертається список з error. Коли отримали відповідь з сервера то також здійснюється перевірка, якщо виникли помилки – заносимо в глобальний масив, інакше, якщо результатом запиту є usertoken то авторизацію проведено успішно. Збереження usertoken в cookie, для наступного POST запиту на сервер та отримання даних по користувачеві, тобто визначення ролі.

```
DoLogin = async () => {
  if (this.state.email && this.state.password) {
    const user = {
      email: this.state.email,
      password: this.state.password,
      login_errors: undefined
    }
    await axios.post('http://diplom:8080/server/user/login', JSON.stringify({
      email: user.email,
      password: user.password
    }))
    .then(async (res) => {
      var data = res.data
      if (data.errors) {
        this.setState({ login_errors: data.errors })
      } else if (data.usertoken) {
        cookie.save('usertoken', data.usertoken, { path: '/' })
        await this.props.setUserToken(data.usertoken)
        await axios.post('http://diplom:8080/server/user/data', JSON.stringify({
          token: cookie.load('usertoken')
        }))
        .then(async (res) => {
          this.setState({
            user_group: res.data[0].group
          })
          cookie.save('user', res.data[0], { path: '/' })
          await this.props.compareUpdate()
        })
        this.props.setLoginPopup([false])
      }
    })
  } else {
    this.setState({ errors: "Some of inputs are empty" })
  }
}
```

Рисунок 3.8 – Метод авторизації на стороні клієнта

						ДП.ІПЗ-32.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			54

дані подаються у вигляді JSON. При успішній авторизації чи реєстрації користувачеві повертається сторінка з особистим кабінетом, в іншому випадку відображається повідомлення про не коректне внесення даних.

АВТОРИЗАЦІЯ

Email address *

Password *

LOGIN

[Forgot your password?](#)

РЕЄСТРАЦІЯ

Your First and Last name *

Email address *

Password *

REGISTER

Рисунок 3.25 – Форма авторизації, реєстрації

Авторизовані користувачі на наступному етапі переходять до свого облікового запису (рис. 3.26).

🏠 > Dashboard

My Account

- ▶ Dashboard
- ▶ Information
- ▶ Orders
- ▶ Wishlist

Halyna Oliynyk

email: informat222@gmail.com

Discount Card: [9a8726b9](#) - 3 %

Рисунок 3.26 – Обліковий запис користувачів

За допомогою свого облікового запису клієнт може змінити основні дані про себе, пароль, переглянути історію замовлень (рис. 3.27).



Рисунок 3.31 – Сторінка відображення детальної інформації про товар

Для зручності замовлення товару на даному сайті клієнтові надається можливість визначити розмір за своїми замірами (рис. 3.32).

Дана форма дає користувачеві чітко дізнатися свій розмір виробу згідно розмірної сітки інтернет-магазину «Вишиванка», яка закладена при розробці визначення розміру, що і є перевагою над іншими аналогами в електронній комерції. В додатку Б зображено розмірну сітку.

XS S M L

ВАШІ ПАРАМЕТРИ

ПАРАМЕТРИ	СМ	РОЗМІР: L
Довжина рукава	61,5	62 см ✓
Ширина плечей	33,8	34,5 см !
Обхват грудей	95,7	96 см ✓
Обхват стегон	99	102 см ✗

Рисунок 3.32 – Визначення розміру за своїми замірами

На рисунку 3.32 містяться позначення, які дають зрозуміти де одяг буде сидіти по фігурі згідно введених параметрів, наприклад кружечок зеленого кольору – означає що введені параметри збігаються з розмірною сіткою та показано що відхилення в межах норми, жовтий кружечок з знаком оклику – означає, що параметри не вписуються у діапазон але не є критичними, а от червоний кружечок з хрестиком – означає, що велике відхилення від допустимих значень розмірної сітки відповідного розміру.

Обравши розмір клієнт переходить до наступного кроку – оформлення замовлення, для цього необхідно додати товар до кошика (рис. 3.33). Дана функція доступна тільки авторизованим користувачам. Якщо користувач не авторизований, на сторінці з'являється форма для реєстрації. Коли користувач здійснив вхід в систему, на сторінці відображається товар, який був доданий до кошика. Звісно для заохочення клієнтів на сайті діє знижка – промокод яку можна отримати при реєстрації, відображення коду та самої відсоткової знижки знаходиться в особистому кабінеті та початковий вигляд кошика містить форму, де потрібно ввести код на отримання знижки.

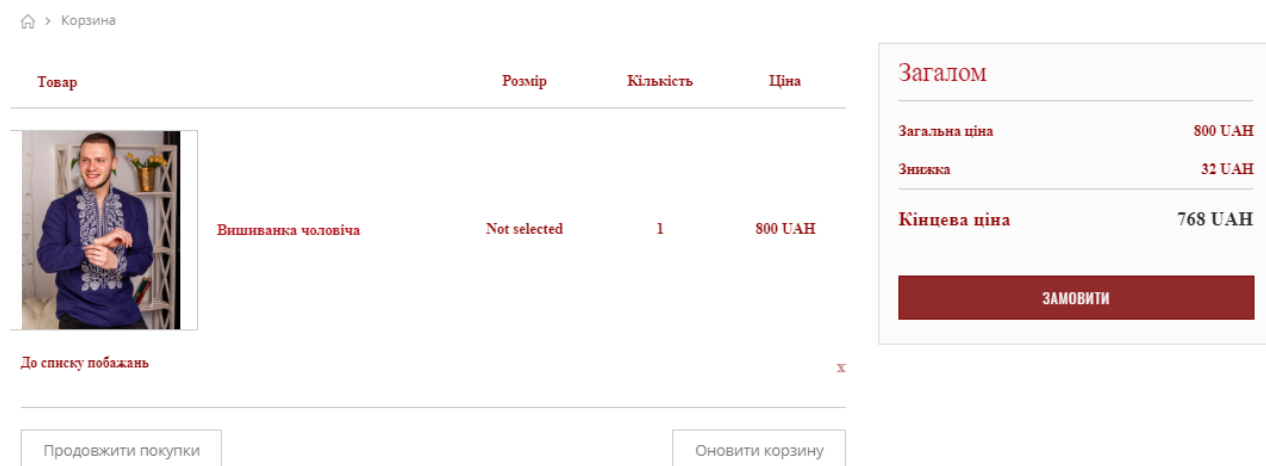


Рисунок 3.33 – Сторінка «Корзина» - авторизованого користувача

Для оформлення замовлення необхідно заповнити форму ввівши свої персональні дані, а також не менш важливим є обрати спосіб доставки, який є двох видів: само вивіз та нова пошта.

					ДП.ІПЗ-32.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

Замовити

Ім'я *

Налупа

Прізвище *

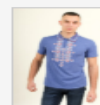
Олійник

Email адреса *

informat222@gmail.com

Загалом

1 PRODUCTS IN CART ^



Футболка чоловіча
420 UAH

Методи замовлення



Фіксована

Самовивіз



За тарифом

Нова пошта

Виберіть місто

Абазівка (Полтавський р-н, Полтавська обл)

ПІДТВЕРДИТИ

Рисунок 3.34 – Заповнення форми «Оформлення замовлення»

На сайті можна залишити відгук про товар (рис. 3.35).

ДОДАЙТЕ ВАШ КОМЕНТАР

Як ви оціните даний товар? *

	1 ЗІРКА	2 ЗІРКИ	3 ЗІРКИ	4 ЗІРКИ	5 ЗІРОК
Якість	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Оцінка	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ціна	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ім'я *

Налупа Олійник

Email адреса *

informat222@gmail.com

Що ви думаєте? *

ВІДПРАВИТИ

Рисунок 3.35 – Відгуки про товар

Клієнтам надаються інформаційні розділи, де можна якщо виникли запитання заповнити форму «Напишіть нам», а також прочитати більш детально про особливості та підхід магазину «Вишиванка».

										ДП.ІПЗ-32.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата							68

користувачів, вести статистику кількості замовлень, коментарів тощо. А основним завданням є наповнення контентом. Тому авторизувавшись на сайті, як адміністратор – надається додаткова вкладка адміністративна панель, здійснивши перехід на яку менеджер-адміністратор має ряд функцій. Більш детально розглянемо їх, спершу відкривається сторінка із статистикою: кількість користувачів на сайті, здійснених замовлень, кількість коментарів (рис. 3.38).

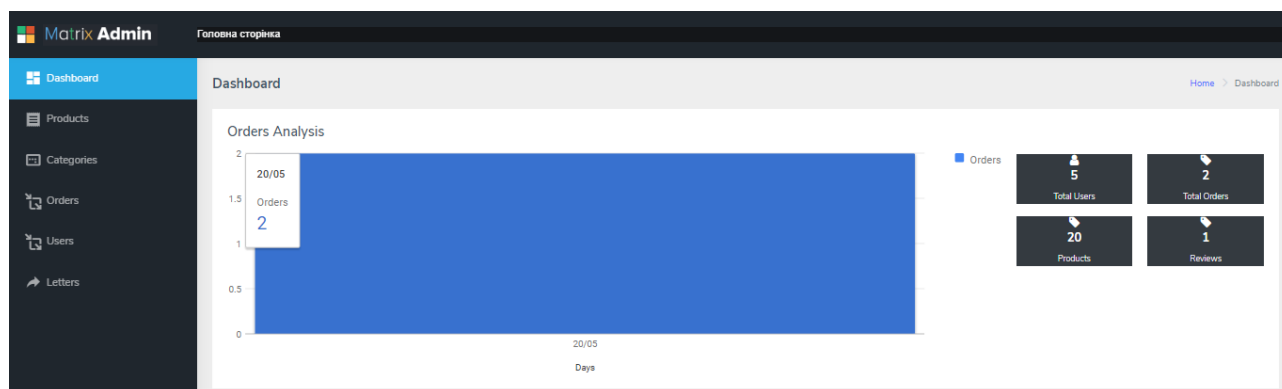


Рисунок 3.38 – Головна сторінка «Адмін панелі»

Для зручної роботи адміністратора з сайтом, було розроблено адміністративну панель за допомогою якої можна додати нові категорії (рис. 3.39), а відповідно до них товар (рис. 3.40). Окрім даної функції звісно ще можна переглядати, редагувати та видаляти.

The screenshot shows the 'Create new category' form in the Matrix Admin dashboard. The form has the following fields: 'Title' with a text input field containing 'Enter product title'; 'Parent Category' with a dropdown menu showing 'Man'; and 'Parameters' with a large text area. A 'Submit' button is located at the bottom left of the form.

Рисунок 3.39 – Сторінка додавання нової категорії

Рисунок 3.40 – Сторінка додавання товару

Як зазвичай основним в інтернет-магазині є клієнт та його замовлення, які обслуговуються адміністратором. Основними функціями до замовлення – перегляд, редагування, видалення, зміна статусу замовлення (рис. 3.41).

ID	Sum	Status	Created At	Actions
27	600	Замовлення прийнято	05/20/2020 15:49	👁️ ✎️ 🗑️
26	450	Замовлення прийнято	05/20/2020 15:48	👁️ ✎️ 🗑️

Рисунок 3.41 – Сторінка замовлення

Реалізований інтерфейс для перегляду інформації з форми «Напишіть нам» (рис. 3.42). З якої адміністратор аналізує роботу інтернет-магазину та виконує побажання клієнтів для підвищення рейтингу магазину.

ID	Name	Email	Created At	Actions
9	Танюша Остафійчук	tanjushaostafijchuk@gmail.com	05/07/2020 14:55	👁️ 🗑️

Рисунок 3.42 – Сторінка відображення запитань клієнтів

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Побудова мережевого графа та розрахунок його параметрів

Як правило, графік проекту розробляється менеджером проекту з залученням людей, які є експертами в тій або іншій сфері. В результаті будується план послідовності переліку робіт, який є структурований за певним ознаками, тобто будується структурна декомпозиція робіт (СДР). Та призначаються відповідальні виконавці на точно вказані терміни та завдання при плануванні складу робіт.

Завдання, як статична категорія з їх послідовністю переводяться в динамічну категорію стану робіт, які приводять до результату проекту. Ці наведені в формат робіт завдання зводяться в графік і/або таблицю. Завдання виражають суть результатів за критерієм виконання («вирішені / не вирішені»).

Для того щоб підтримувати роботу виконавців на проекті та не відставати від графіку використовують календарний план. А також календарний план є зручний і для замовника, адже дає чітко зрозуміти коли, можна буде очікувати проект на стадії завершення.

Календарний план простий у використанні, формат якого відображає ключові завдання проекту, дати початку та закінчення, тривалість завдання та відповідні панелі завдань, які оновлюються неминучими змінами проекту. Налаштування панелі завдань різними кольорами покращить спосіб розмежувати ролі, команди чи інші категорії візуально організованим чином.

Цей єдиний документ показує всі завдання, їх терміни та відповідальних за виконання. Надає всім членам команди цілодобовий доступ до шкали управління проектами. Це, в свою чергу, робить весь проект більш прозорим.

					ДП.ІПЗ-32.ІЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

Кінець таблиці 4.1

Навчання замовника.	2 дні	29.04.20	30.04.20	8	Контент-менеджер
Вибір хостинга та доменного ім'я.	1 день	04.05.20	04.05.20	10	Веб-програміст
Перенесення сайту на хостинг.	1 день	05.05.20	05.05.20	10	Веб-програміст
Тестування сайту.	2 дні	06.05.20	07.05.20	11;12	Тестувальник
Передача сайту замовникам. Підписування необхідних документів при прийманні-здачі роботи.	1 день	08.05.20	08.05.20	13	Аналітик

На виконання проектного рішення було витрачено 49 днів. Для планування і керування розробкою проекту побудовано мережеву діаграму, яка відповідає переліку робіт в таблиці 4.1. В мережевому графі детально показано, що, в якій послідовності, коли, за який час, для чого необхідно виконати, щоб забезпечити закінчення всіх робіт не пізніше заданого терміну.

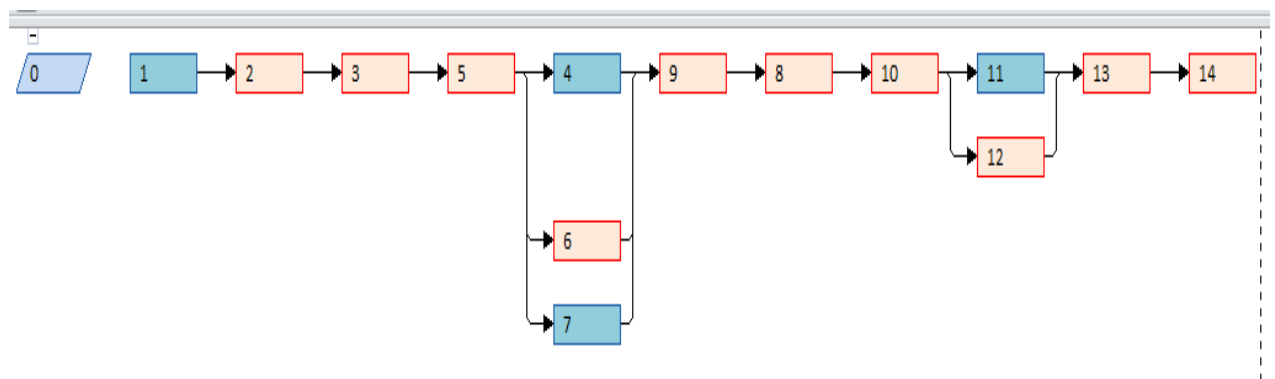


Рисунок 4.1 Мережевий граф

Тому послідовність етапів, на яких з'ясується, який найменший проміжок часу необхідний для виконання завдання з найменшим запасом часу. Отже, критичний шлях - це дійсно найдовший проміжок часу, який знадобиться для виконання проектних завдань. Величина критичного шляху

Створена діаграма Ганта в MS Project складається з налаштованим спринтами, кожен з яких може складатися з чотирьох елементів:

- дати початку кожної роботи;
- дати закінчення робіт;
- тривалість робіт;
- ресурси необхідні для виконання робіт [21].

На рисунку 4.4 зображено діаграму Ганта. На даній діаграмі видно повний список робіт необхідних для виконання. Тільки після реалізації всіх представлених робіт можна досягти мети проекту. Так само представлено планові дати початку і закінчення робіт і послідовність виконання робіт.

На діаграмі Ганта відображені резерви часу некритичних робіт їх виконання. На виконання проектної частини було витрачено 49 днів.

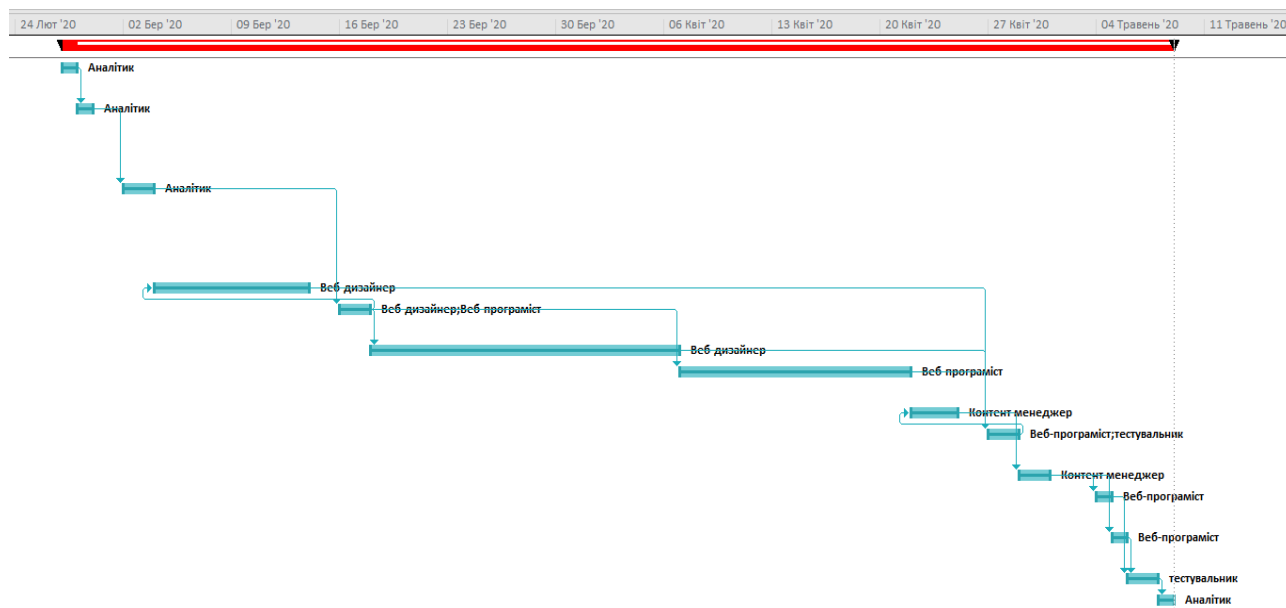


Рисунок 4.4 Діаграма Ганта

4.2 Економічне обґрунтування розробки та впровадження проектного рішення

Для економічного обґрунтування розробки та впровадження проектного рішення обчислюється ряд економічних показників – витрати на розробку та впровадження проектного рішення.

Розрахунок витрат на розробку та впровадження проектного рішення включає в себе наступні складові:

- витрати коштів для нарахування оплати праці розробникам;
- єдиний соціальний внесок;
- вартість додаткових виробів, що закуповуються;
- транспортно-заготівельні витрати;
- накладні витрати;
- інші витрати;
- витрати на від лагодження і дослідну експлуатацію програмного забезпечення на ЕОМ;
- експлуатаційні витрати;
- ціна споживання проектного рішення;
- показники економічної ефективності.

4.2.1 Розрахунок витрат на розробку та впровадження проектного рішення

Розрахунок витрат на розробку та впровадження проектного рішення, що визначається за формулою (4.1) та включає в себе наступні складові [22]:

$$K_{\text{заг.}} = K_1 + K_2, \quad (4.1)$$

					ДП.ІПЗ-32.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		77

де K_1 – витрати на розробку програмного засобу, грн.;

K_2 – витрати на відлагодження і досліду експлуатацію програмного забезпечення на ЕОМ, грн.

Визначення витрат на оплату праці та відрахувань на соціальні заходи.

Відповідно до Закону України « Про оплату праці» заробітна плата – це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу» [23].

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, наслідків його праці та господарської діяльності підприємства. Заробітна плата складається з базової та допоміжної оплати праці [23].

Базова заробітна плата як відомо нараховується на виконану роботу за тарифною сіткою, договірними розцінками чи посадовими окладами і не залежить від наслідків господарської діяльності підприємства.

Допоміжна заробітна плата – це частинка заробітку працівників, до якої входять витрати на оплату праці, не пов'язані з виплатами за нормований відпрацьований час. Нараховують допоміжну заробітну плату залежно від виконаних і запланованих показників, умов виробництва, кваліфікаційної підготовки робітників. Джерелом допоміжної оплати праці є фонд матеріального заохочення, який створюється за рахунок доходів.

Для обчислення середньоденної заробітної плати кожного з працівників можна використати наступну формулу [22]:

$$ЗП_{дi} = \frac{ЗП_i}{\Phi_m}, \quad (4.2)$$

де $ЗП_i$ – основна місячна заробітна плата розробника 1-ої спеціальності, грн.;

Φ_m – місячний фонд робочого часу, днів (24 дні).

										ДП.ІПЗ-32.ПЗ	Арк.
											78
Зм.	Арк.	№ докум.	Підпис	Дата							

Нарахування витрат на оплату праці усіх розробників проекту можна здійснити за формулою [22]:

$$B_{оп} = \sum_{i=1}^N n_i \cdot t_i \cdot ЗП_{ди}, \quad (4.3)$$

де n_i – чисельність розробників проекту i -ої спеціальності, чол.;

t_i – час, витрачений на розробку проекту працівником i -ої спеціальності, дні;

$ЗП_{ди}$ – денна заробітна плата розробника i -ої спеціальності, грн.

Середньоденна заробітна плата працівників:

$$ЗП_{ди} = \frac{4008,75}{24} = 167,03 \text{ грн.}$$

Тоді витрати для нарахування зарплати розробників проекту являють:

$$B_{оп} = (1 \cdot 167,03 \cdot 5) + (1 \cdot 167,03 \cdot 14) + (1 \cdot 167,03 \cdot 21) + (1 \cdot 167,03 \cdot 5) + (1 \cdot 167,03 \cdot 4) = 8184,44 \text{ грн.}$$

Додаткова зарплатня становить 10–15 % від суми основної зарплатні [22].

$$З_{дод.} = З_{осн.} \cdot K_{додл.}, \quad (4.4)$$

де $K_{додл.}$ – коефіцієнт додаткових виплат працівникам, 0,1–0,15.

Аналітик : $З_{осн.} = 167,03 \cdot 5 = 835,15 \text{ грн.}$

$$З_{дод.} = 835,15 \cdot 0,15 \text{ грн.}$$

					ДП.ІПЗ-32.ІЗ	Арк.
						79
Зм.	Арк.	№ докум.	Підпис	Дата		

Веб-програміст: $З_{осн.} = 167,03 \cdot 14 = 2338,42 \text{ грн.}$

$$З_{дод.} = 2338,42 \cdot 0,15 = 350,76 \text{ грн.}$$

Веб-дизайнер: $З_{осн.} = 167,03 \cdot 21 = 3507,63 \text{ грн.}$

$$З_{дод.} = 3507,63 \cdot 0,15 = 526,14 \text{ грн.}$$

Контент-менеджер: $З_{осн.} = 167,03 \cdot 5 = 835,12 \text{ грн.}$

$$З_{дод.} = 835,12 \cdot 0,15 = 125,27 \text{ грн.}$$

Тестувальник: : $З_{осн.} = 167,03 \cdot 4 = 668,12 \text{ грн.}$

$$З_{дод.} = 668,12 \cdot 0,15 = 100,22 \text{ грн.}$$

Отже всі витрати на оплату праці ($B_{o.n.}$) визначаються за формулою(4.5)
[22]:

$$B_{o.n.} = З_{осн.} + З_{дод.} \cdot \quad (4.5)$$

Аналітик : $B_{o.n.} = 835,15 + 125,27 = 960,42 \text{ грн.}$

Веб-програміст: $B_{o.n.} = 2338,42 + 350,76 = 2689,18 \text{ грн.}$

Веб-дизайнер: $B_{o.n.} = 3507,63 + 526,14 = 4033,77 \text{ грн.}$

Контент-менеджер: $B_{o.n.} = 835,12 + 125,27 = 960,39 \text{ грн.}$

					ДП.ІПЗ-32.ПЗ	Арк.
						80
Зм.	Арк.	№ докум.	Підпис	Дата		

Тестувальник: $B_{o.n.} = 668,12 + 100,22 = 768,34$ грн.

Всього: $B_{o.n.} = 960,42 + 2689,18 + 4033,77 + 960,39 + 768,34 = 9412,1$ грн.

Витрати на оплату зарплатні працівникам тягнуть за собою додаткові зобов'язання підприємства по сплаті ЄСВ ($B_{есв}$).

Аналітик: $B_{есв} = 960,42 \cdot 0,22 = 211,29$ грн.

Веб-програміст: $B_{есв} = 2689,18 \cdot 0,22 = 591,62$ грн.

Веб-дизайнер: $B_{есв} = 4033,77 \cdot 0,22 = 887,43$ грн.

Контент-менеджер: $B_{есв} = 960,39 \cdot 0,22 = 211,29$ грн.

Тестувальник: $B_{есв} = 768,34 \cdot 0,22 = 169,04$ грн.

Загальний ЄСВ: $B_{есв} = 9604,19 \cdot 0,22 = 2070,66$ грн.

Нарахування єдиного внеску складають 22 відсотки до бази нарахування згідно із Законом України «Про внесення змін до Податкового кодексу України та деяких законодавчих актів України щодо забезпечення збалансованості бюджетних надходжень у 2016 році» № 909-VIII від 24 грудня 2015 року, яким внесено зміни, зокрема, до Закону України від 08 липня 2010 року № 2464 «Про збір та облік єдиного внеску на загальнообов'язкове державне соціальне страхування» [23].

Проведені розрахунки витрат на оплату праці зводять у таблицю 4.2.

					ДП.ІПЗ-32.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

Визначення транспортно-заготівельних витрат. Транспортно-заготівельні витрати (B_{mp}) для розрахунку даного типу витрат становлять 11% від суми придбання закуплених виробів.

$$B_{mp1} = 180 \cdot 0,11 = 19,8 \text{ грн.}$$

$$B_{mp1} = 140 \cdot 0,11 = 15,4 \text{ грн.}$$

$$B_{mp1} = 90 \cdot 0,11 = 9,9 \text{ грн.}$$

Визначення накладних витрат. Накладні витрати (B_n) проектних організацій передбачають витрати на керування, загальногосподарські, невиробничі витрати. Вони становлять 25% витрат на оплату праці.

$$B_n = 9412,10 \cdot 0,25 = 2353,03 \text{ грн.}$$

Визначення інших витрат. Інші витрати ($B_{ін}$) — це витрати, які не враховані в попередніх статтях витрат. Їх розраховують за встановленими відсотками до витрат на оплату праці (6%).

$$B_{ін} = 9412,10 \cdot 0,06 = 564,73 \text{ грн.}$$

Визначення витрат на розробку проектного рішення.

Після вирахування всіх витрат, можна виконати підрахунок витрат на розробку проектного рішення, використовуючи формулу (4.6) [22]:

$$K_1 = B_{он} + B_{есв} + B_o + B_n + B_{н} + B_{ін} \quad , \quad (4.6)$$

					ДП.ІПЗ-32.ІЗ	Арк.
						83
Зм.	Арк.	№ докум.	Підпис	Дата		

де B_{on} – витрати на оплату праці розробників;

$B_{есв}$ – єдиний соціальний внесок;

B_o – вартість додаткових виробів, що закупаються;

$B_{тp}$ – транспортно-заготівельні витрати;

B_n – накладні витрати;

$B_{ін}$ – інші витрати.

$$K_1 = 9412,10 + 2070,68 + 410 + 45,1 + 2353,03 + 564,73 = 14855,64 \text{ грн.}$$

Визначення витрат на налагодження і використанні програмного забезпечення визначаються згідно з формули (4.7) [22]:

$$K_2 = S_{м.г.} \cdot t_{від} , \quad (4.7)$$

де $S_{м.г.}$ – вартість однієї години роботи ПК, грн./год.;

$t_{від}$ – кількість годин роботи ПК на налагодження програми, год.

Загальна кількість днів роботи на комп'ютері рівна 50 днів. Середній щоденний час роботи на ЕОМ - 6 год., тому:

$$t_{від} = 49 \cdot 6 = 294 \text{ грн.}$$

Витрати використання, що становлять 1 год машинного часу, можуть бути обчисленні за витратою електроенергії (4.8) [22] :

$$e_2 = P_{сн} \cdot C_{вод} , \quad (4.8)$$

де $P_{сн} = 104$ Вт - споживана потужність ЕОМ;

					ДП.ІПЗ-32.ІЗ	Арк.
						84
Зм.	Арк.	№ докум.	Підпис	Дата		

$C_{\text{вод}} = 1.68$ - вартість 1 кВт/год електроенергії для споживачів.

Отже, за формулою (4.8):

$$e_z = 0,104 \cdot 1,68 = 0,17 \text{ грн./год.}$$

Отже:

$$K_2 = 0,17 \cdot 294 = 49,98 \text{ грн.}$$

Виходячи з формули обрахунку витрат на розробку та використання проектного рішення одержуємо подальші результати:

$$K_{\text{заг.}} = 14855,64 + 49,98 = 14905,62 \text{ грн.}$$

Підготовка кошторису на розробку проектного рішення. Результати розрахунків для формування кошторису зводяться у таблицю 4.4.

Таблиця 4.4 – Кошторис витрат на розробку проектного рішення

Найменування елементів витрат	Сума витрат, грн.
Витрати на розробку проектного рішення, у т.ч.:	
Витрати на оплату праці	9 412,10
Єдиний соціальний внесок	2 070,68
Витрати на додаткові вироби, що закуповуються	410
Транспортно-заготівельні витрати	45,1
Накладні витрати	2 353,03
Інші витрати	564,73
Витрати на налагоджену і дослідну експлуатацію системи	49,98
Всього	14 905,62

4.2.2 Обрахунки експлуатаційних витрат

Підготовка даних (E_1) і вартість години роботи комп'ютера (E_2) включається у вартість використання витрат при порівнянні програмних засобів.

Експлуатаційні одноразові витрати обчислюються за формулою (4.9) [24]:

$$E_{\Pi(A)} = E_{1\Pi(A)} + E_{2\Pi(A)}. \quad (4.9)$$

Значущість підготовки даних для використання проектного рішення (аналогу) (E_1) розраховуємо згідно формули (4.10) [24]:

$$E_1 = \sum_{i=1}^N n_i t_i \cdot 3\Pi z_i. \quad (4.10)$$

Середня ставка годин веб-розробника визначається за формулою (4.11) [24]:

$$3\Pi z_i = \frac{3\Pi z_{oi}(1+b)}{\Phi z}, \quad (4.11)$$

де b – коефіцієнт відрахувань на соціальні заходи, становить 0.22;

Φ – фонд робочого часу (год)

Отже: окупність підготовки даних для застосування проектного рішення (аналогу):

$$E_{\Pi} = 78,12 \text{ грн.}$$

$$E_{\Pi} = 136,36 \text{ грн.}$$

					ДП.ІПЗ-32.ПЗ	Арк.
						86
Зм.	Арк.	№ докум.	Підпис	Дата		

Окупність машини години для роботи ПК визначена попередньо і становить 0,17 грн./год.

$$E_{II} = 0,17 \cdot 5 = 0,85 .$$

Витрати на налагодження і апробаційне використання аналога становлять:

$P_{cn} = 150$ Вт - споживана потужність ЕОМ;

$C_{вод} = 1,68$ - вартість 1 кВт/год електроенергії для споживачів.

$$e_z = 0,150 \cdot 1,68 = 0,28 \text{ грн./год.}$$

Отже:

$$E_A = 0,28 \cdot 5 = 1,40 .$$

Одноразові витрати використання на розробку проектного рішення становлять:

$$E_{II} = 78,12 + 0,85 = 78,97 .$$

Одноразові витрати використання на розробку аналога становлять:

$$E_A = 136,36 + 1,40 = 137,76 \text{ грн.}$$

Визначення річних витрат використання проектного рішення та аналогу.

Річні витрати використання обраховуються за формулою (4.12):

					ДП.ІПЗ-32.ПЗ	Арк.
						87
Зм.	Арк.	№ докум.	Підпис	Дата		

$$B_{(e)П(A)} = E_{П(A)} \cdot N_{П(A)}, \quad (4.12)$$

де $N_{П(A)} = 52$ тижні.

$$B_{(e)П} = 78,97 \cdot 52 = 4106,44 \text{ грн.}$$

$$B_{(e)A} = 137,76 \cdot 52 = 7163,52 \text{ грн.}$$

Початкові дані та результати обрахунків витрат на підготовку даних для використання проектного рішення зводяться у таблицю 4.5.

Таблиця 4.5 – Обчислень витрат на підготовку даних для роботи на ПК

Категорія персоналу	Чисельність співробітників і-ої категорії, чол.	Час роботи співробітників і-ої категорії, год.	Середньо годинна ЗП співробітника і-ої категорії, грн.	Витрати на підготовку даних, грн.
Проектне рішення				
Програміст	1	2	39,06	78,12
Всього	1	2	39,06	78,12
Аналог				
Програміст	1	2	68,18	136,36
Всього	1	2	68,18	136,36

Отже, проектне рішення було розроблено із меншими фінансовими затратами порівняно із досліджуваним аналогом, то можна зробити висновок, що розроблена модель є більш рентабельною і доступнішою ніж аналог.

$$B_{(e)NPV} = \sum_{t=1}^T \frac{B_{(E)Nt}}{(1+R)^t}, \quad (4.15)$$

де $B_{(E)Nt}$ – річні витрати використання в t -ому році, грн.;

При умові, що впродовж всього строку використання $B_{(E)Nt} = const$ – тоді розрахунки здійснюються за формулою (4.16) [24]:

$$B_{(E)NPV} = \sum_{t=1}^T B_{(T)nt} \cdot \frac{1}{(1+R)^t} = pvB_{(E)n}, \quad (4.16)$$

де pv – ставка дисконтна на період T , яка визначається залежно від процентної ставки R і періоду експлуатації T .

При $R=0,14$ і $T=4$ роки ставка дисконту має таке значення.

Таблиця 4.6 – Ставка дисконту

T	0	1	2	3	4
PV	1,0	0,88	0,77	0,68	0,59

$$B_{(E)NPV} = 0,88 + 0,77 + 0,68 + 0,59 = 2,92.$$

У результаті розрахунків ціна споживання проектного рішення склала:

$$Ц_{c(П)} = 18781,08 + 2,92 \cdot 4106,44 = 30771,88 \text{ грн.}$$

Ціна споживання аналогу склала:

$$Ц_{c(A)} = 24000 + 2,92 \cdot 7163,52 = 44917,49 \text{ грн.}$$

					ДП.ІПЗ-32.ПЗ	Арк.
						90
Зм.	Арк.	№ докум.	Підпис	Дата		

4.2.4 Визначення показників економічної ефективності

Економічне відображення в області експлуатації (грн.) визначається за формулою (4.17) [25]:

$$E_{екс} = B_{(е)а} - B_{(е)н} . \quad (4.17)$$

$$E_{екс} = 7163,52 - 4106,44 = 3057,08 \text{ грн.}$$

Економічне відображення в області проектування (грн.) визначається за формулою (4.18) [25]:

$$E_{нр} = Ц_a - Ц_n . \quad (4.18)$$

$$E_{нр} = 44917,49 - 30771,88 = 14145,61 \text{ грн.}$$

Якщо $E_{нр} > 0$ та $E_{екс} > 0$, то розраховується додатковий економічний ефект в сфері експлуатації (грн.) (4.19) [25]:

$$E_{ексД} = \sum_{i=1}^T E_{екс} (1 + R)^{T-i} . \quad (4.19)$$

$$E_{ексД} = 3057,08 \cdot (1 + 0,14)^3 + 3057,08 \cdot (1 + 0,14)^2 + 3057,08 \cdot (1 + 0,14)^1 = 11983,75 \text{ грн.}$$

Допоміжний економічний ефект в області проектування (грн.) (4.20) [25] :

$$E_{нрД} = E_{нр} \times (1 + R)^T . \quad (4.20)$$

$$E_{нрД} = 14145,61 \cdot (1 + 0,14)^4 = 23906,08 \text{ грн.}$$

					ДП.ІПЗ-32.ПЗ	Арк.
						91
Зм.	Арк.	№ докум.	Підпис	Дата		

експлуатаційні витрати, економічний ефект та термін окупності. Значення були порівняні із значеннями існуючого аналогу, що довело переваги над готовим рішенням. Даний проект є економічно доцільний і вигідний, це пояснюється меншими затратами на розробку і експлуатацію без втрати якості та надійності продукту, великим економічним ефектом та коротким терміном окупності на проектне рішення становить 6 місяців.

					ДП.ІПЗ-32.ПЗ	Арк.
						93
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті даного дипломного проекту було розроблено інтернет-магазин для продажу етно-одягу. Для створення веб-сайту також здійснено порівняльний аналіз аналогів інтернет-магазинів, визначено основні переваги та недоліки, на основі аналізу розроблено функціональну структуру сайту з урахуванням всіх вимог споживача.

Відповідно спроектовано UML-діаграми: діаграму класів – з блоку структурні діаграми та діаграму варіантів використання, діаграму послідовності процесів з категорії – діаграми поведінки. Також було зроблено проект структури бази-даних складається з сутностей, які пов'язані між собою зв'язками.

З аналізу технологій для створення Web-сайтів, було обрано framework ReactJS – клієнтської частини та FuelPHP – для серверної частини, як СУБД обрано MySQL, для запуску сервера – OpenServer, а середовище розробки веб-застосунку було обрано Microsoft Visual Studio. Інтернет-магазин для продажу етно-одягу має зручний та зрозумілий інтерфейс користувача.

Економічні розрахунки проведені в процесі виконання дипломного проекту, а саме проведення розрахунків витрат на розробку програмного забезпечення відображають доцільність розробки даного програмного продукту.

Використавши дані технології було розроблено інтернет-магазин на замовлення для реального магазину «Вишиванка», що знаходиться в місті Надвірна Івано-Франківської області, про що свідчить «Акт впровадження результатів».

					ДП.ІПЗ-32.ПЗ	Арк.
						94
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А

Таблиця відповідності розмірів

Розмір		Висота рукава	Ширина плечей	Обхват грудей	Обхват бедер
42	XS	59	46	84	90
44	S	60	47	88	94
46	M	61	49	92	98
48	L	62	51	96	102
50	XL	63	52	100	106
52	XXL	64	55	104	110
54	3XL	65	57	108	114
56	4XL	66	59	112	118
58	5XL	67	61	116	122
60	6XL	68	64	120	126

ДОДАТОК Б

ФОП

Тороус Наталія Василівна

78400 м. Надвірна, вул.

Чорновола, 4

тел. (067) 682-47-72

АКТ ВПРОВАДЖЕННЯ

результатів дослідження за темою дипломної роботи

в практику діяльності підприємця

Підстави для здійснення науково-дослідної роботи студенткою

Замовник Тороус Наталія Василівна, ФОП доручає студентці Прикарпатського національного університету імені Василя Стефаника Остафійчук Тетяні Дмитрівні, напряму підготовки спеціальності 121 «Інженерія програмного забезпечення» виконати дослідження та розробку за темою «Розробка Web-сайту інтернет-магазину для продажу етно-одягу» в термін з жовтня 2019 р. по травень 2020 р. під керівництвом д.т.н., проф. Кузь М.В..

За результатами проведеного дослідження визнано доцільним впровадити інтернет-магазин «Вишиванка».

ФОП магазину «Вишиванка»
(підпис)

_____ Тороус Н.В.

ДОДАТОК В

ФОП

Тороус Наталія Василівна
78400 м. Надвірна, вул. Чорновола, 4
тел. (067) 682-47-72

ДОВІДКА ПРО ВПРОВАДЖЕННЯ

19.05.20

м. Надвірна

№_

Видана студентці четвертого курсу напряму підготовки спеціальності 121 «Інженерія програмного забезпечення» Прикарпатського національного університету імені Василя Стефаника Остафійчук Тетяні Дмитрівні в тому, що вона дійсно виконала дипломну роботу на тему: «Розробка Web-сайту інтернет-магазину для продажу етно-одягу» на реальних даних магазину «Вишиванка», м. Надвірна, вул. Чорновола 4.

Практичні пропозиції студентки щодо запровадження, яка наведена у дипломній роботі, має практичну значущість і впроваджені у діяльність.

ФОП магазину «Вишиванка»
(підпис)

_____ Тороус Н.В.

ДОДАТОК Г

Лістинг програми, Main.js

```

import React, { PureComponent } from 'react';
import { BrowserRouter, Route, Switch } from 'react-router-dom'
import cookie from 'react-cookies'
import axios from 'axios'
import Popup from 'reactjs-popup'
import Header from './components/blocks/Header'
import Footer from './components/blocks/Footer'
import Index from './components/pages/Index'
import History from './History'
import About from './components/pages/About'
import Contact from './components/pages/Contact'
import Products_View from './components/pages/Product_View'
import Categories_Index from './components/pages/Categories_Index'
import Account from './components/pages/Account';
import Cart_View from './components/pages/Cart_View'
import Mobile from './Mobile.js'
import Error from './components/blocks/Error'
import i18n from './i18n'
import Fade from 'react-reveal/Fade';
import Compare from './components/pages/Compare';
import $ from 'jquery'
class Main extends PureComponent {
  // конструктор головного класу
  constructor(props) {
    super(props)

    this.state = {
      usertoken: cookie.load('usertoken'), // токен користувача
      user: undefined, // користувач
      cart: cookie.load('cart') ? cookie.load('cart') : [], // корзина
      cart_count: cookie.load('cart_count') ? cookie.load('cart_count') : 0, // сум
а корзини
      lang: cookie.load('lang') ? cookie.load('lang') : 'en', // мова користувача
      popup_show: false, // попап вікно
      popup_message: "", // попап повідомлення
      login_popup: false
    }
    // бінд основних функцій, які будуть передаватися в підкомпоненти
    this.onCartDelete = this.onCartDelete.bind(this)
    this.onClose = this.onClose.bind(this)
    this.deleteCompares = this.deleteCompares.bind(this)
    this.addCompare = this.addCompare.bind(this)
    this.compareUpdate = this.compareUpdate.bind(this)

    this.setModal = this.setModal.bind(this)

```

Продовження додатку Г

```

this.setLoginPopup = this.setLoginPopup.bind(this)

    this.cartAllRemove = this.cartAllRemove.bind(this)
  }

  // початкова функція
  componentDidMount = async () => {

    $('.admin-links').remove()

    i18n.changeLanguage(this.state.lang)

    document.documentElement.lang = this.state.lang

    if (this.state.usertoken) this.compareUpdate()

    await axios.get(`https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?js
n&valcode=USD`)
      .then(res => {
        this.setState({
          usd_rate: res.data[0].rate
        })
      })
  }

  // функція встановлення токена користувача
  setUserToken = (token, r) => {
    this.setState({ usertoken: token })
  }

  // функція додавання товару до корзини
  onCartAdd = async (e, id, quantity, size) => {

    e.preventDefault()
    // отримання інформації про товар з сервера для збереження
    await axios.get(`http://diplom:8080/server/products/single?id=` + id)
      .then(res => {
        var cart = cookie.load('cart') ? cookie.load('cart') : []

        var repeated = false

        var data = res.data

        cart.map((item, key) => {
          if (item.id === id) {
            repeated = true
            cart[key]['quantity']++
          }
        })
      })
  }

```

Продовження додатку Г

```

    }
  })

  if (repeated === false) {
    // записання товару в перемінну
    data.quantity = quantity ? quantity : 1
    data.size = size
    cart.push(data);

  }

  this.setState({ cart: cart, cart_count: Number(this.state.cart_count) + Number(data.new_price * data.quantity) }, () => {
    //збереження корзини та суми в куки
    cookie.save('cart', cart, { path: '/' })
    cookie.save('cart_count', this.state.cart_count, { path: '/' })
  })
})

}

// функція додавання товару до списку побажань, приймає event та id товару
addToWish = async (e, id) => {
  e.preventDefault()

  var msg

  if (this.state.usertoken === undefined) {
    msg = "Login or register to add this item"
  } else {
    await axios.post(`http://diplom:8080/server/wishlist/add`, JSON.stringify({
      model_id: id,
      user_id: cookie.load('user').id
    })).then(res => {

      if (res.data.code === 'success') {
        msg = "You have successfully added an item to wishlist"
      } else {
        msg = "This product is already in your wishlist"
      }
    })
  }

  }

  // показати попап вікно з потрібним повідомленням
  this.setState({
    popup_message: msg,
    popup_show: true
  })
}

```

Продовження додатку Г

```

    })
        }

    // функція видалення товару з корзини, приймає event та ключ товару в масиві state
    onCartDelete(e, key) {
        e.preventDefault()
        // видалений елемент за ключом
        var deleted = this.state.cart.splice(key, 1)

        var newcart = this.state.cart

        this.setState({
            cart: newcart,
            cart_count: Number(this.state.cart_count) -
            Number(deleted[0].new_price * deleted[0].quantity)
        }, function () {
            cookie.save('cart', this.state.cart, { path: '/' })
            cookie.save('cart_count', this.state.cart_count, { path: '/' })
        })
    }

    cartAllRemove() {
        this.setState({
            cart: [],
            cart_count: 0
        }, function () {
            cookie.save('cart', this.state.cart, { path: '/' })
            cookie.remove('cart_discont', { path: '/' })
            cookie.save('cart_count', this.state.cart_count, { path: '/' })
        })
    }

    onClose() {
        this.setState({ popup_show: false })
    }

    // оновити список порівняння
    compareUpdate = async () => {
        await axios.get(`http://diplom:8080/server/user/compares?user_id=` + cookie.load(
        'user').id)
            .then(res => {
                this.setState({ compares: res.data })
            })
    }

    // видалити з списку побажань, приймає event та id товару
    deleteCompares = async (e, id) => {

```


Продовження додатку Г

```

e.preventDefault()

var new_c = this.state.compares.data.filter(item => item.id !== id)

this.setState({
  compares: {
    data: new_c,
    count: new_c.length
  }
}, async () => {
  var list = []
  this.state.compares.data.map((item) => list.push(item.id))

  await axios.post(`http://diplom:8080/server/user/set_compare`, JSON.stringify
({
  new_list: list,
  user_id: cookie.load('user').id
}))
  .then(res => {
    console.log(res.data)
  })
})

}

// додати товар до списку порівнять
addCompare = async (e, id, cat_id) => {
  e.preventDefault()

  if (this.state.compares.count >= 2) {
    this.setState({
      popup_show: true,
      popup_message: "Limit of items to compare is 2"
    })
  } else {
    if (cookie.load('user')) {
      if (this.state.compares.data.length === 1 && this.state.compares.data[0].
category_id !== cat_id) {
        this.setState({
          popup_show: true,
          popup_message: "You can not add two products with different categ
ories"
        })

        return
      }

      await axios.get(`http://diplom:8080/server/user/add_compare?id=` + id + '
&user_id=' + cookie.load('user').id)
        .then(async (res) => {

```

Продовження додатку Г

```

        // оновити список
        await this.compareUpdate()
    })
} else {
    this.setState({
        popup_show: true,
        popup_message: "Log in or register to add this product to compare lis
t"
    })
}
}

setModal(value, msg) {
    this.setState({
        popup_show: value,
        popup_message: msg ? msg : ""
    })
}

setLoginPopup(e) {
    this.setState({
        login_popup: e
    })
}

render() {

    return (

        <div className="App">

            <BrowserRouter>
                <History>

                    <Header
                        setUserToken={this.setUserToken}
                        link={this.state.link}
                        cart={this.state.cart}
                        cart_count={this.state.cart_count}
                        lang={this.state.lang}
                        compares={this.state.compares}
                        compareUpdate={this.compareUpdate}
                        deleteCompares={this.deleteCompares}
                        usd_rate={this.state.usd_rate}
                        login_popup={this.state.login_popup}
                        setLoginPopup={this.setLoginPopup}
                    />

```

Продовження додатку Г

```

<Switch>

  <Route exact path="/">
    <Fade big>
      <Index onCartAdd={this.onCartAdd} addToWish={this.addToWish}
        ToWish} usd_rate={this.state.usd_rate} />
    </Fade>
  </Route>

  <Route exact path="/cart">
    <Fade big>
      <Cart_View
        onCartAdd={this.onCartAdd}
        onCartDelete={this.onCartDelete}
        addToWish={this.addToWish}
        usd_rate={this.state.usd_rate}
        setLoginPopup={this.setLoginPopup}
        setModal={this.setModal}
      />
    </Fade>
  </Route>

  <Route path="/product/:id">
    <Fade big>
      <Products_View
        onCartAdd={this.onCartAdd}
        addCompare={this.addCompare}
        addToWish={this.addToWish}
        usd_rate={this.state.usd_rate}
        setLoginPopup={this.setLoginPopup}
      />
    </Fade>
  </Route>

  <Route exact path="/categories/:id?/:title?">
    <Fade big>
      <Categories_Index
        onCartAdd={this.onCartAdd}
        addToWish={this.addToWish}
        setLoginPopup={this.setLoginPopup}
      />
    </Fade>
  </Route>

  <Route exact path="/about">
    <Fade big>
      <About />
    </Fade>
  </Route>

```

Продовження додатку Г

```

</Route>

    <Route exact path="/contact">
      <Fade big>
        <Contact />
      </Fade>
    </Route>

    {this.state.usertoken &&
    <>
      <Route exact path="/compare">
        <Fade big>
          <Compare
            compares={this.state.compares}
            compareUpdate={this.compareUpdate}
            onCartAdd={this.onCartAdd}
            // addCompare={this.addCompare}
            addToWish={this.addToWish}
            usd_rate={this.state.usd_rate}
            setLoginPopup={this.setLoginPopup}
          />
        </Fade>
      </Route>

      <Route path="/account">
        <Fade big>
          <Account usd_rate={this.state.usd_rate} cartA
llRemove={this.cartAllRemove} />
        </Fade>
      </Route>
    </>
  }

  <Route component={Error} />

</Switch>
<Footer setModal={this.setModal} />
<Popup open={this.state.popup_show} closeOnDocumentClick onClose=
{this.onClose} >
  <div className="text-center">
    <h4 className="mt-4 mb-4">
      {this.state.popup_message}
    </h4>
  </div>
</Popup>

</History>

```

Продовження додатку Г

```
        </BrowserRouter>
        <div id="mobile-wrapper">
            <Mobile />
        </div>
        <a id="scroll-top" href="#top" role="button"><i className="icon-angle-
up"></i></a>
    </div>

    )    }
}

export default Main
```

ДОДАТОК Д

Лістинг програми, Header.js

```

import React, { PureComponent } from 'react'
import Login from '../forms/Login'
import Register from '../forms/Register'
import CartDropdown from './CartDropdown'
import axios from 'axios'
import { Link } from 'react-router-dom'
import cookie from 'react-cookies'
import Popup from 'reactjs-popup'
import { withRouter } from 'react-router-dom'
import { LazyLoadImage } from 'react-lazy-load-image-component';
import { Trans, withTranslation } from 'react-i18next';
import i18n from '../i18n'
class Header extends PureComponent {
  // конструктор класу шапки сайту
  constructor(props) {
    super(props)
    this.state = {
      user_group: cookie.load('user') ? cookie.load('user').group : undefined, // роль\група користувача
      cart: cookie.load('cart') ? cookie.load('cart') : [], // корзина
      email: undefined,
      password: undefined,
      login_errors: undefined,
      register_errors: undefined,
      cart_count: cookie.load('cart_count') ? cookie.load('cart_count') : 0, // сума корзини
      logout_popup: false,
      lang: this.props.lang, // мова користувача
      money: cookie.load('money') ? cookie.load('money') : 'uah', // валюта,
      login_popup: false
    }
  }

  // метод пошуку по слову і категорії
  doSearch = async (e) => {
    e.preventDefault()
    var text = e.target.search.value,
        cat = e.target.cat.value ? e.target.cat.value : ""
    //редирект за ключовим словом і категорією
    await this.props.history.push('/categories/' + cat + '?search=' + text)
  }

  // підтвердження форми авторизації
  onSubmitLogin = async (e) => {
    e.preventDefault()

```

Продовження додатку Д

```

this.setState({
  email: e.target.email.value,
  password: e.target.password.value
})
// виконати авторизацію
await this.DoLogin();
}
// метод оновлення компоненту, потрібен для оновлення корзини при додаванні\видаленні
товару
componentDidUpdate = async (prevProps, prevState) => {
  if (prevState === this.state && prevProps.location.pathname === this.props.location.pathname) {

    this.setState({
      cart: this.props.cart,
      cart_count: 0
    }, () => {
      var price = 0;
      // перерахування суми корзини, щоб уникнути помилок
      this.state.cart.map((item) => price += Number(item.new_price * item.quantity))

      this.setState({ cart_count: price })
      cookie.save('cart_count', price, { path: '/' })
    })
  }

  // початкова функція програми, отримує всі категорії, щоб занести у список для пошуку
  componentDidMount = async () => {
    await axios.get(`http://diplom:8080/server/admin/all_categories`)
      .then(async (res) => {
        this.setState({ categories: res.data })
        // якщо користувач авторизований показати список порівняння товарів
        if (this.state.user_loaded) await this.props.compareUpdate()
      })
  }

  // видалення товару з корзини
  deleteCartProduct = (e, id) => {
    e.preventDefault()
    var deleted = this.state.cart.splice(id, 1)
    var newcart = this.state.cart
    this.setState({
      cart: newcart,
      cart_count: Number(this.state.cart_count) -
        Number(deleted[0].new_price * deleted[0].quantity)
    }, function () {
      cookie.save('cart', this.state.cart, { path: '/' })
      cookie.save('cart_count', this.state.cart_count, { path: '/' })
    })
  }
}

```

Продовження додатку Д

```

// метод видалення сесії авторизації користувача
logout = async () => {
  cookie.remove('usertoken', { path: '/' })
  cookie.remove('user', { path: '/' })

  this.setState({ user_group: undefined, logout_popup: true })
}

componentWillUpdate(prevProps, prevState) {
  if (prevState.user_group === undefined && this.state.user_group !== undefined) {
    this.props.history.push('/')
  }
}

// метод авторизації
DoLogin = async () => { // асинхронна функція авторизації
  if (this.state.email && this.state.password) { // якщо введено email і пароль
    const user = { // введення константи користувача з двома полями і полем можли
вих помилок
      email: this.state.email,
      password: this.state.password,
      login_errors: undefined
    }
    //відправлення POST запиту на сервер з потрібними полями
    await axios.post(`http://diplom:8080/server/user/login`, JSON.stringify({
      email: user.email,
      password: user.password
    }))
    .then(async (res) => {
      var data = res.data

      if (data.errors) { // якщо є помилки - занести їх в глобальний масив
        this.setState({ login_errors: data.errors })
      } else if (data.usertoken) { // якщо результатом запиту є токен автор
изації, все успішно
        cookie.save('usertoken', data.usertoken, { path: '/' }) // збереж
ення токену в куки
        await this.props.setUserToken(data.usertoken) // передача токену
в батьківський клас для обробки
        await axios.post(`http://diplom:8080/server/user/data`, JSON.stri
ngify({ //отримання даних користувача по токену
          token: cookie.load('usertoken')
        }))
        .then(async (res) => {
          this.setState({

```


Продовження додатку Д

```

        user_group: res.data[0].group // збереження номеру гру
упи, якщо адмін, то в меню буде цей пункт
    })
    cookie.save('user', res.data[0], { path: '/' })

    await this.props.compareUpdate()
  })
  this.props.setLoginPopup(false)
}
  })
} else {
  this.setState({ errors: "Some of inputs are empty" }) // якщо серед полів є п
усте поле
}
}
// метод для відправки форми реєстрації
onSubmitRegister = async (e) => {
  e.preventDefault()
  // константа користувача з введеними даними
  var user = {
    email: e.target.email.value,
    password: e.target.password.value
  };
  await axios.post(`http://diplom:8080/server/user/register`, JSON.stringify({
    username: e.target.username.value,
    email: e.target.email.value,
    password: e.target.password.value
  })))
  .then(async (res) => {
Продовження додатку Д
    var data = res.data
      if (data.errors) { // якщо є помилки занести у масив state
        this.setState({ register_errors: data.errors })
      } else {

        this.setState({
          email: user.email,
          password: user.password,
          register_errors: undefined
        })
        // виконати авторизацію
        await this.DoLogin();
      }
    })
  }

onChange = (e) => {
  this.setState({
    [e.target.name]: e.target.value
  })
}

```

Продовження додатку Д

```

    })
  }
  onChangeLanguage = async (e, lang) => {
    e.preventDefault()
    await i18n.changeLanguage(lang);
    cookie.save('lang', lang, { path: '/' })
    this.setState({ lang: lang })
    document.documentElement.lang = lang;
  }
  onChangeMoney = (e, money) => {
    e.preventDefault()
    cookie.save('money', money, { path: '/' })
    this.setState({ money: money })
  }
  componentWillMount() {
    document.addEventListener('mousedown', this.handle, false)
  }
  componentWillUnmount() {
    document.removeEventListener('mousedown', this.handle, false)
  }
  handle = (e) => {
Продовження додатку Д
    if (this.node.contains(e.target)) {
      return
    }
    this.props.setLoginPopup(false)
  }
  render() {
    return (
      <header className="header">
        <Popup
          open={this.state.logout_popup}
          closeOnDocumentClick
        >
          <div className="text-center">
            <h4 className="mt-4 mb-4">
              <Trans i18nKey='blocks.logged-out-modal'></Trans>
            </h4>
          </div>
        </Popup>
        <div ref={node => this.node = node}>
          <Popup open={this.props.login_popup} closeOnDocumentClick modal >
            <>
              <style dangerouslySetInnerHTML={{ __html: ` .popup-
content{ width: 85% !important; } ` }}></style>
              <div className="mfp-content" style={{ width: "100%" }}>

```


Продовження додатку Д

```

<div className="header-left header-dropdowns">
  <div className="header-dropdown">
    <a href="">{this.state.money === 'usd' ? 'USD' : 'UAH'}</
a>

    <div className="header-menu">
      <ul>
        <li><a href="" onClick={(e) => this.onChangeMoney
(e, 'uah')}>UAH</a></li>
        <li><a href="" onClick={(e) => this.onChangeMoney
(e, 'usd')}>USD</a></li>
      </ul>
    </div>
  </div>

  <div className="header-dropdown">
    <a href="#">
      <LazyLoadImage
        Продовження додатку Д
        src={this.state.lang === 'en' ? "/assets/images/flags/en.png" : "/assets/images/flag
s/uk.png"}
        effect="blur"
      />
      {this.state.lang === 'en' ? 'ENGLISH' : 'UKRAINIAN'}
    </a>
    <div className="header-menu">
      <ul>
        <li><a href="" onClick={(e) => this.onChangeLangu
age(e, 'en')}>ENGLISH</a></li>
        <li><a href="" onClick={(e) => this.onChangeLangu
age(e, 'ua')}>UKRAINIAN</a></li>
      </ul>
    </div>
  </div>
  {cookie.load('usertoken') && this.props.compares &&
  <div className="dropdown compare-dropdown">

    <a href="#" className="dropdown-
toggle" role="button" data-toggle="dropdown" aria-haspopup="true"
    aria-expanded="false" data-display="static">
      <i className="icon-retweet"></i>
      <Trans i18nKey="blocks.compare_link"></Trans> ({t
his.props.compares.count})
    </a>

    <div className="dropdown-menu">

```

Продовження додатку Д

```

<div className="dropdownmenu-wrapper">
  <ul className="compare-products">
    {this.props.compares.data.map((item, key)
=>
      <li className="product" key={key}>
        <a onClick={(e) => this.props.deleteCompares(e, item.id)} style={{ cursor: "pointer" }} className="btn-remove" title="Remove Product"><i className="icon-cancel"></i></a>
        <h4 className="product-title"><Link to={"/product/" + item.id}>{item.title}</Link></h4>
      </li>
    )}
  </ul>

  <div className="compare-actions">
    <Link to="/compare" className="btn btn-primary"><Trans i18nKey="blocks.to_compare_link"></Trans></Link>
  </div>
</div>
</div>
}
</div>

<div className="header-right">
  <div className="header-dropdown dropdown-expanded">
    <a href="#"><Trans i18nKey='blocks.links'></Trans></a>
    <div className="header-menu">
      <ul>
        {!cookie.load('usertoken') ?
        <>
          <li><Link to="" onClick={(e) => this.props.setLoginPopup(true)}><Trans i18nKey='blocks.my_account'></Trans></Link></li>
          <li><Link to="" onClick={(e) => this.props.setLoginPopup(true)}><Trans i18nKey='blocks.wishlist'></Trans></Link></li>
          <li><a onClick={(e) => this.props.setLoginPopup(true)} style={{ cursor: "pointer" }}><Trans i18nKey='blocks.login'></Trans></a></li>
        </>
        :
        <>
          <li><Link to="/account"><Trans i18nKey='b
locks.my_account'></Trans></Link></li>

```

Продовження додатку Д

```

18nKey='blocks.wishlist'></Trans></Link></li>
<li><Link to="/account/wishlist"><Trans i
ns i18nKey='blocks.logout'></Trans></a></li>
</>
}
</ul>
</div>
</div>
</div>
</div>
<div className="header-middle">
  <div className="container">
    <div className="header-left">
      <Link to="/" className="logo">
        <LazyLoadImage
          src="/assets/images/logo_100.png"
          effect="blur"
        />
      </Link>
    </div>
    <div className="header-center">
      <div className="header-search">
        <a href="#" className="search-
toggle" role="button"><i className="icon-magnifier"></i></a>
        <form action="#" onSubmit={this.doSearch}>
          <div className="header-search-wrapper">
            <input type="search" className="form-
control" name="search" id="q" placeholder={this.props.t('blocks.search_place')} required
/>
            <div className="select-custom">
              <select id="cat" name="cat">
                <option value="">{this.props.t('blocks.al
l_cats')}</option>
                {this.state.categories &&
                  this.state.categories.map((item, key)
=>
                    <option key={key} value={item.id}
>{item.title}</option>
                )
              }
            </select>
          </div>
          <button className="btn" type="submit"><i classNam
e="icon-magnifier"></i></button>

```

Продовження додатку Д

```

        </div>
      </form>
    </div>
  </div>
  <div className="header-right">
    <button className="mobile-menu-toggler" type="button">
      <i className="icon-menu"></i>
    </button>
    <div className="header-contact">
      <span><Trans i18nKey='blocks.call_us'></Trans></span>
      <a href="tel:#"><strong>+38 097 987 7745</strong></a>
    </div>
    <CartDropdown
      cart={this.state.cart}
      usd_rate={this.props.usd_rate}
      deleteCartProduct={this.deleteCartProduct}
      setLoginPopup={this.props.setLoginPopup}
      cart_count={this.state.cart_count}
    />
  </div>
</div>
</div>
<div className="header-bottom sticky-header">
  <div className="container">
    <CartDropdown cart={this.state.cart} deleteCartProduct={this.delet
eCartProduct} cart_count={this.state.cart_count} />
    <nav className="main-nav">
      <ul className="menu sf-arrows">
        <li id="home"><Link to="/"><Trans i18nKey='blocks.home_li
nk'></Trans></Link></li>
        {this.state.user_group &&
          this.state.user_group === '2' &&
          <li><a href="/admin"><Trans i18nKey='blocks.admin_lin
k'></Trans></a></li>
        }
        <li><Link to="/categories" id="contacts"><Trans i18nKey='
blocks.products_link'></Trans></Link></li>
        <li><Link to="/contact" id="contacts"><Trans i18nKey='blo
cks.contacts_link'></Trans></Link></li>
        <li><Link to="/about" id="about"><Trans i18nKey='blocks.a
bout_link'></Trans></Link></li>
      </ul>
    </nav>
  </div>
</div>
</header>
)  }}
export default withTranslation()(withRouter(Header))

```

ДОДАТОК Е

Лістинг програми, Admin.php

```

<?php
use Fuel\Core\Input;
class Controller_Admin extends Controller_Rest
{
    protected $format = 'json';
    protected $status = [
        0 => 'Замовлення прийнято',
        1 => 'Замовлення в обробці',
        2 => 'Замовлення відправлено',
        3 => 'Замовлення успішно'
    ];
    public function post_update_order()
    {
        $json = file_get_contents('php://input');
        $data = json_decode($json);
        $result = DB::update('orders')
            ->set(array(
                'username' => $data->username,
                'email' => $data->email,
                'status' => $data->status
            ))
            ->where('id', '=', $data->id)
            ->execute();
        return $this->response(['id' => $data->id], 200);
    }
    public function get_set_parameters_category()
    {
        return $this->response(['hello' => 1], 200);
    }
    public function get_all_statuses()
    {
        return $this->response($this->status, 200);
    }
    public function get_all_role_users()
    {
        $users = DB::select('id', 'email', 'group')->from('users')->execute();
        $list = [];
        foreach($users as $k => $u){
            $list[$k] = $u;
        }
        return $this->response($list, 200);
    }
    public function get_set_role()
    {
        $result = DB::update('users')
            ->set(array(

```


Продовження додатку Е

```

        'group' => Input::get('group')
    ))
    ->where('id', '=', Input::get('id'))
    ->execute();
    return $this->response(['code' => 'success'], 200);
}
public function get_all_products()
{
    $list = DB::select_array()->from('products')->as_assoc()->order_by('created_at',
'desc')->execute();
    $arr = [];
    foreach ($list as $k => $l) {
        $arr[$k] = $l;
    }
    return $this->response($arr, 200);
}
public function get_all_categories()
{
    $list = DB::select_array()->from('categories')->as_assoc()->execute();
    $arr = [];
    foreach ($list as $k => $l) {
        $arr[$k] = $l;
    }
    return $this->response($arr, 200);
}
public function get_all_colors()
{
    $list = DB::select_array()->from('colors')->as_assoc()->execute();
    $arr = [];
    foreach ($list as $k => $l) {
        $arr[$k] = $l;
    }
    return $this->response($arr, 200);
}
public function get_all_letters()
{
    $list = DB::select_array()->order_by('created_at', 'DESC')->from('letters')->
as_assoc()->execute();
    $arr = [];
    foreach ($list as $k => $l) {
        $arr[$k] = $l;
        $arr[$k]['created_at'] = Date::forge($arr[$k]['created_at'])-
>format("%m/%d/%Y %H:%M", true);
    }
    return $this->response($arr, 200);
}

public function get_all_orders()
{

```

Продовження додатку E

```

        $list = DB::select_array()->order_by('created_at', 'DESC')->from('orders')-
>as_assoc()->execute();
        $arr = [];
        foreach ($list as $k => $l) {
            $arr[$k] = $l;
            $arr[$k]['created_at'] = Date::forge($arr[$k]['created_at'])-
>format("%m/%d/%Y %H:%M", true);
            $arr[$k]['status'] = $this->status[$arr[$k]['status']];
        }
        return $this->response( $arr , 200);
    }
    public function get_all_sizes()
    {
        $arr = [
            'XS', 'S', 'M', 'L', 'XL', 'XXL', '3XL', '4XL', '5XL', '6XL'
        ];
        return $this->response( $arr , 200);
    }
    public function post_delete_color()
    {
        $json = file_get_contents('php://input');
        $data = json_decode($json);
        $list = DB::delete('colors')->where('id', $data->id)->execute()[0];
        return $this->response( $list , 200);
    }
    public function post_create_color()
    {
        $json = file_get_contents('php://input');
        $data = json_decode($json);
        // var_dump($data);die;
        list($id, $rows) = DB::insert('colors')->set(
            (array) $data
        )->execute();
        return $this->response( ['id' => $id] , 200);
    }
    public function get_get_order()
    {
        $id = Input::get('id');
        $list = DB::select_array()->from('orders')->where('id', $id)->as_assoc()-
>execute()[0];
        $list['created_at'] = Date::forge($list['created_at']->format("%m/%d/%Y %H:%M",
true);
        $list['status'] = $this->status[$list['status']];
        return $this->response( $list , 200);
    }

    public function get_get_letter()
    {
        $id = Input::get('id');

```

Продовження додатку Е

```

        $list = DB::select_array()->from('letters')->where('id', $id)->as_assoc()-
>execute()[0];
        $list['created_at'] = Date::forge($list['created_at']->format("%m/%d/%Y %H:%M",
true);
        return $this->response( $list , 200);
    }
    public function get_get_color()
    {
        $id = Input::get('id');
        $list = DB::select_array()->from('colors')->where('id', $id)->as_assoc()-
>execute()[0];
        return $this->response( $list , 200);
    }
    /**
     *
     * PRODUCTS ADMIN
     *
     */
    public function get_get_product()
    {
        $id = Input::get('id');
        $list = DB::select_array()->from('products')->where('id', $id)->as_assoc()-
>execute()[0];
        $image = DB::select()->from('images')->where('model_id', $list['id'])-
>execute()[0];
        $list['image'] = $image;
        return $this->response( $list , 200);
    }
    public function get_get_category()
    {
        $id = Input::get('id');
        $list = DB::select_array()->from('categories')->where('id', $id)->as_assoc()-
>execute()[0];
        return $this->response( $list , 200);
    }
    public function post_image()
    {
        if (!empty($_FILES)) {
            $config = array(
                'path' => DOCROOT . 'assets/img/products',
                'randomize' => true,
                'ext_whitelist' => array('img', 'jpg', 'jpeg', 'gif', 'png'),
            );
            Upload::process($config);
            if (Upload::is_valid()) {
                Upload::save();
                $value = Upload::get_files();
                return $this->response( ['path' => '/assets/img/products/' .
$value[0]['saved_as']] , 200);

```

Продовження додатку Е

```

    }
    }
}

public function post_delete_product()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $list = DB::delete('products')->where('id', $data->id)->execute()[0];
    $img = DB::delete('images')->where('model_id', $data->id)->execute()[0];
    return $this->response( $list , 200);
}

public function post_delete_letter()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $list = DB::delete('letters')->where('id', $data->id)->execute()[0];
    return $this->response( $list , 200);
}

public function post_delete_category()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $list = DB::delete('categories')->where('id', $data->id)->execute()[0];
    return $this->response( $list , 200);
}

public function get_parameters_by_category()
{
    $id = Input::get('id');
    $parameters = DB::select('parameters')->from('categories')->where('id', $id)-
>execute()[0];
    return $this->response($parameters['parameters'], 200);
}

public function post_update_category()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $result = DB::update('categories')
        ->set(array(
            'title' => $data->title,
            'main' => $data->main,
            'parameters' => json_encode($data->parameters)
        ))
        ->where('id', '=', $data->id)
        ->execute();
    return $this->response( ['id' => $data->id] , 200);
}

public function post_create_contact()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $contact = $data->contact;

    list($insert_id, $rows_affected) = DB::insert('letters')->set(array(
        'name' => $contact->name,

```

Продовження додатку Е

```

        'email' => $contact->email,
        'phone' => !empty( $contact->phone ) ? $contact->phone : null,
        'text' => $contact->text,
        'created_at' => Date::forge()->set_timezone('Europe/Kiev')->get_timestamp()
    ))->execute();
    return $this->response( ['id' => $insert_id] , 200);
}

public function post_update_product()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $product = $data->product;
    $old = DB::select('parameters')->from('products')->where('id', $data->product->product->id)->execute()[0]['parameters'];
    unset($product->parameters->show_c);
    unset($product->parameters->parameters);
    $old = json_decode($old, true);
    $new = [];
    foreach($product->parameters as $k => $arr){
        if(!is_int($k)) $new[$k] = $arr;    }
    $parameters = json_encode( array_merge($old, $new) );
    $result = DB::update('products')
        ->set(array(
            'title' => isset($product->title) ? $product->title : $product->product->title,
            'old_price' => !empty($product->old_price) ? $product->old_price : $product->product->old_price,
            'title' => isset($product->title) ? $product->title : $product->product->title,
            'sizes' => !empty($product->sizes) ? json_encode( $product->sizes ) : $product->product->sizes,
            'parameters' => !empty($product->parameters) ? $parameters : $product->product->parameters,
            'colors' => json_encode($product->colors),
            'category_id' => isset($product->category_id) ? $product->category_id : $product->product->category_id,
            'new_price' => isset($product->new_price) ? $product->new_price : $product->product->new_price,
            'description' => isset($product->description) ? $product->description : $product->product->description,
        ))
        ->where('id', '=', $data->product->product->id)
        ->execute();
    if (isset($product->imagePath)) {
        $image = DB::update('images')
            ->set(array(
                'src' => $data->imagePath,
            ))
            ->where('model_id', '=', $data->product->product->id)

```

Продовження додатку Е

```

        ->execute();    }
    return $this->response( ['id' => $product->product->id] , 200);    }
public function post_create_product()

{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $product = $data->product;
    unset($product->parameters->show_c);
    unset($product->parameters->parameters);
    $parameters = json_encode( $product->parameters );
    list($insert_id, $rows_affected) = DB::insert('products')->set(array(
        'title' => $product->title,
        'old_price' => !empty($product->old_price) ? $product->old_price : null,
        'sizes' => json_encode($product->sizes),
        'parameters' => $parameters,
        'colors' => json_encode($product->colors),
        'category_id' => $product->category_id,
        'new_price' => $product->new_price,
        'description' => !empty($product->description) ? $product->description : "",
        'created_at' => Date::forge()->set_timezone('Europe/Kiev')->get_timestamp(),
        'updated_at' => Date::forge()->set_timezone('Europe/Kiev')->get_timestamp()
    ))->execute();
    list($id, $rows) = DB::insert('images')->set(array(
        'model_id' => $insert_id,
        'src' => $data->imagePath,
        'type' => 'single'
    ))->execute();
    return $this->response( ['id' => $insert_id] , 200);    }
public function post_create_category()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $data->parameters = json_encode(array_filter($data->parameters, 'strlen'));
    list($id, $rows) = DB::insert('categories')->set(
        (array) $data
    )->execute();
    return $this->response( ['id' => $id] , 200);    }
public function get_email_news()
{
    $email = Input::get('email');
    $result = '';

    if (DB::select()->from('email_news')->where('email', $email)->execute()[0]) {
        $result = 'already';
    } else {
        list($id, $rows) = DB::insert('email_news')->set([
            'email' => $email
        ])->execute();
        $result = 'created';
    }
}
return $this->response( $result , 200);}}

```

ДОДАТОК Ж

Лістинг програми, User.php

```
<?php

use Fuel\Core\Input;

class Controller_User extends Controller_Rest
{
    protected $format = 'json';

    public function get_get_discont()
    {
        $user_id = Input::get('user_id');
        $code = Input::get('code');
        $card = DB::select('id', 'code', 'percent')->from('discont_cards');

        if(!empty($user_id)) $card = $card->where('user_id', $user_id);
        else $card = $card->where('code', $code);

        $card = $card->execute()[0];

        $code = !empty($card) ? true : false;
        return $this->response(['code' => $code, 'card' => $card], 200);
    }

    public function post_login()
    {
        $json = file_get_contents('php://input');

        $data = json_decode($json);

        $email = $data->email;

        $password = $data->password;

        $list = [];

        if (Auth::login($email, $password)) {
            $list['usertoken'] = Auth::get('login_hash');
        } else {
            $list['errors'] = 'User not found or something went wrong!';
        }

        return $this->response($list, 200);
    }
}
```

Продовження додатку Ж

```

public function post_register()
{
    $json = file_get_contents('php://input');
    $list = [];
    $data = json_decode($json);
    try {
        $user = Auth::create_user(
            $data->username,
            $data->password,
            $data->email,
            1,
            []
        );
        list($insert_id, $rows_affected) = DB::insert('discont_cards')->set(array(
            'code' => substr(md5(time()), 0, 8),
            'user_id' => $user
        ))->execute();

    } catch (SimpleUserUpdateException $e) {
        $list['errors'] = 'Username or Email already exists';
    }
    return $this->response($list, 200);
}

public function post_data()
{
    $json = file_get_contents('php://input');
    $list = [1];
    $data = json_decode($json);

    $list = DB::select()->from('users')->where('login_hash', $data->token)-
>execute();
    return $this->response($list, 200);
}

public function get_compares_cat()
{
    $arr = [];
    $list = DB::select('products')->from('compares')->where('user_id',
Input::get('user_id'))->execute()[0];

    $products = json_decode($list['products']);

    if (!empty($products)) {
        foreach ($products as $k => $l) {
            $arr[$k] = DB::select('id', 'title', 'category_id', 'parameters',
'new_price')->from('products')->where('id', $l)->execute()[0];
            $arr[$k]['image'] = DB::select('src')->from('images')->where('model_id',
$arr[$k]['id'])->execute()[0]['src'];
        }
    }
}

```


Продовження додатку Ж

```

$parameters = DB::select('parameters')->from('categories')->where('id',
$arr[0]['category_id']->execute()[0]['parameters'];

$compares = [];

foreach (json_decode($parameters) as $k => $p) {

    foreach ($arr as $a => $val) {

        foreach (json_decode($val['parameters'], true) as $kk => $j) {
            $compares[$kk]['values'][$a] = $j;
        }
    }
}

$compares = array_values($compares);
foreach (json_decode($parameters) as $k => $p) {
    $compares[$k]['name'] = $p;
}

return $this->response( ['data' => $arr, 'count' => count($arr), 'compares' =>
$compares] , 200);
}

public function get_comparares()
{
    $arr = [];
    $list = DB::select('products')->from('compares')->where('user_id',
Input::get('user_id'))->execute()[0];

    $products = json_decode($list['products']);

    if (!empty($products)) {
        foreach ($products as $k => $l) {
            $arr[$k] = DB::select('id', 'title', 'category_id')->from('products')->where('id', $l)->execute()[0];
        }
    }

    return $this->response( ['data' => $arr, 'count' => count($arr)] , 200);
}

public function get_add_compare()
{
    $id = Input::get('id');
    $user_id = Input::get('user_id');

    $arr = DB::select()->from('compares')->where('user_id', $user_id)->execute()[0];
    if (!empty($arr)) {

```

Продовження додатку Ж

```

    $products = json_decode($arr['products']);
    if (!in_array($id, $products)) {
        $products[] = $id;
        $products = json_encode($products);
        $result = DB::update('compares')->value("products", $products)
            ->where('user_id', $user_id)
            ->execute();
    }
} else {
    list($insert_id, $rows_affected) = DB::insert('compares')->set(array(
        'user_id' => $user_id,
        'products' => "[" . json_encode( $id ) . "]"
    ))->execute();
}
return $this->response( ['code' => 'success'] , 200);
}
public function post_set_compare()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $result = DB::update('compares')
        ->value("products", json_encode($data->new_list))
        ->where('user_id', $data->user_id)
        ->execute();
    return $this->response( ['code' => 'success'] , 200);
}
public function post_edit_info()
{
    $json = file_get_contents('php://input');
    $data = json_decode($json);
    $data = $data->data;

    $result = DB::update('users')
        ->set(array(
            'username' => $data->first_name . ' ' . $data->last_name,
            'email' => $data->email,
        ))
        ->where('id', '=', $data->user->id)
        ->execute();

    if (!empty($data->pass)) {
        $result = DB::update('users')
            ->set(array(
                'password' => \Auth::instance()->hash_password($data->pass)
            ))
            ->where('id', '=', $data->user->id)
            ->execute();
    }
}
var_dump($result);    }}

```