

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-07

Дипломний проєкт

здобувачки освіти денної форми навчання

РП.07.06.000.ДП

***КУРІЛЕЦЬ НАТАЛІЇ
АНДРІЇВНИ***

м. Одеса
2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Інженерія програмного забезпечення»

Група: 4РП-07

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту на тему:

Розробка веб-орієнтованої системи для ведення кав'ярного бізнесу

Проектний матеріал складається з пояснювальної записки на 78 сторінках та графічного (презентаційного) матеріалу на 12 аркушах (слайдах).

Дипломник _____ (Курілець Н. А.)

Керівник _____ (Жадан А. С.)

Консультанти:

з економічного розділу _____ (Іванченков В. С.)

з розділу охорони праці та техніки безпеки _____ (Чорновол Н. І.)

з нормоконтролю _____ (Петрашова В. І.)

старший консультант _____ (Кривченко Ю. В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю. В.)

Завідувач відділення _____ (Скорнякова О. В.)

Захист 25 » 06 2024 р.

Протокол ЕК № 2

Оцінка ЕК 5 (Врештало) / 95

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І. В.
“ 16 ” 07 2024 року

ЗАВДАННЯ
на дипломний проєкт

Здобувачі освіти Курілець Наталії Андріївни

1. Тема проєкту Розробка веб-орієнтованої системи для ведення кав'ярного бізнесу

Затверджена наказом по коледжу від “ 02 ” листопада 2023 р., наказ № 244-А2-ОД

2 Термін здачі закінченого проєкту 10.06.2024.

3. Вихідні дані до проєкту

1. Застосувати фреймворк Bootstrap для графічного інтерфейсу

2. Використовувати СУБД MySQL для бази даних

3. Застосовувати Micro-framework PHP для створення застосунку

4. Застосовувати принципи ООП для підтримування коду

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Аналіз предметної області. 2. Технології та засоби розробки (проєктування).

3. Проєктування веб-дизайну. 4. Проєктування архітектури веб-застосунку.

5. Розробка веб-застосунку. 6. Тестування створеного веб-застосунку.

7. Економічний розрахунок. 8. Аспекти охорона праці та техніки безпеки

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Презентація Power Point – 12 слайдів

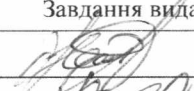
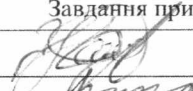
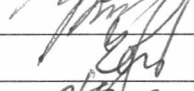

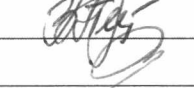



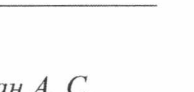

(Загальні відомості; Колірна гама веб-застосунку; Схема технологій веб-застосунку;

Схема мапи веб-застосунку; Схема БД веб-застосунку; Процес-розробки веб-застосунку;

Метод на отримання меню; Основна функціональність веб-застосунку;

Тестування веб-сторінки з меню)

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується


Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Жадан А. С.		
Економічний розділ	Іванченков В. С.		
Розділ охорони праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кривченко Ю. В.		

7. Дата видачі завдання

15.01.2024

Керівник

Жадан А. С.


(підпис)

Завдання прийняв до виконання

Курілець Н. А.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

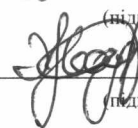
№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Формування вступу	29.04.24	виконано
2	Аналіз предметної області	10.05.24	виконано
3	Підбір технічної літератури	19.05.24	виконано
4	Вибір технологій та засобів розробки (проекткування)	20.05.24	виконано
5	Проекткування веб-дизайну веб-застосунку	22.05.24	виконано
6	Проекткування архітектури веб-застосунку	24.05.24	виконано
7	Реалізація веб-застосунку	27.05.24	виконано
8	Тестування створеного веб-застосунку	29.05.24	виконано
9	Оформлення пояснювальної записки	31.05.24	виконано
10	Оформлення графічної (презентаційної) частини	01.06.24	виконано
11	Економічний розрахунок	02.06.24	виконано
12	Опис охорони праці та техніки безпеки	09.06.24	виконано
13	Аналіз результатів проектування	13.06.24	виконано
14	Підготовка доповіді для захисту	16.06.24	виконано

Дипломник



(підпис)

Керівник



(підпис)

ЗМІСТ

ВСТУП	7
1 ОСНОВНИЙ РОЗДІЛ	9
1.1 Аналіз предметної області	9
1.1.1 Огляд існуючих рішень	9
1.1.2 Технології та засоби розробки	14
1.2 Проєктування застосунку	19
1.2.1 Технічне завдання на розробку	19
1.2.2 Проєктування дизайну застосунку	20
1.2.3 Проєктування бази даних	22
1.2.4 Проєктування архітектури застосунку	27
1.3 Реалізація застосунку	29
1.3.1 Налаштування бази даних	29
1.3.2 Налаштування роутингу	30
1.3.3 Створення моделей (Models)	30
1.3.4 Створення представлень (Views)	33
1.3.5 Створення контролерів (Controllers)	37
1.3.6 Створення системи авторизації	38
1.4 Тестування створеного веб-застосунку	41
1.4.1 Функціональний огляд головної веб-сторінки	42
1.4.2 Функціональний огляд інформаційних веб-сторінок	43
1.4.3 Функціональний огляд веб-сторінок зворотнього зв'язку	45
1.4.4 Функціональний огляд системи адміністрування	47
2 ЕКОНОМІЧНИЙ РОЗДІЛ	50
2.1 Резюме	50
2.2 Визначення трудомісткості розробки програмного забезпечення	50
2.3 Розрахунок ціни програмного продукту	54
3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	56
3.1 Вступ	56

3.2 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці програмного комплексу	56
3.3 Гігієнічні вимоги до виробничого середовища	57
3.3.1 Вимоги до приміщення	57
3.3.2 Освітлення	57
3.3.3 Шум	57
3.3.4 Мікроклімат	58
3.3.5 Електробезпека	58
3.3.6 Вимоги до організації робочого місця працівника	59
3.4 Пожежна безпека	60
ВИСНОВКИ	62
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	63
ДОДАТОК А. Програмний код основної логіки веб-застосунку	64
ДОДАТОК Б. Слайди мультимедійної презентації	73

ВСТУП

У сучасному світі кав'ярні стають не лише місцями для вживання кави, але й соціальними центрами, де люди збираються для роботи, зустрічей або просто відпочинку. Зростання популярності кавової культури створює попит на ефективні та інноваційні рішення для управління кав'ярним бізнесом. Застосування веб-орієнтованих систем управління стає критичною необхідністю для оптимізації процесів, забезпечуючи підприємствам зручність, швидкість та ефективність у діяльності.

Веб-системи для кав'ярень можуть об'єднувати в собі різноманітні компоненти, що включають у себе системи замовлення та доставки, управління запасами, контроль якості, аналіз витрат та інше. Ці системи дозволяють власникам кав'ярень ефективно керувати усіма аспектами бізнесу, підвищуючи продуктивність та забезпечуючи високий рівень обслуговування для клієнтів.

Завдяки автоматизації засобами веб-систем, кав'ярні можуть швидше реагувати на зміни в попиті, оптимізувати робочі процеси та прискорювати обробку замовлень. Це дозволяє підприємствам бути більш конкурентоспроможними на ринку та забезпечує їхню стабільність у динамічному середовищі бізнесу.

Системи управління кав'ярнею можуть включати в себе функціонал для замовлення столиків, керування меню, обліку фінансів та звітності перед контролюючими органами. Застосування різних ролей у системі дозволяє ефективно розподіляти відповідальність між персоналом, забезпечуючи доступ лише до необхідної інформації та функціоналу для кожного користувача.

Метою даної роботи є розробка веб-орієнтованої системи для ведення кав'ярного бізнесу, яка має застосовувати передові технології для оптимізації основних аспектів діяльності кав'ярні. Система повинна забезпечувати зручний та ефективний інструментарій для управління, замовлення та аналізу даних, що дозволить власникам кав'ярень підтримувати високий рівень сервісу та конкурентоспроможності

					<i>РП 07. 06 000. 00 ДП ПЗ</i>	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

При дослідженні основних проблем предметної області, аналізі аналогів та засобів розробки використано системний підхід. Для створення бази даних (БД) використано технологію розробки реляційних БД MySQL. Для розробки програмного продукту використано технологію об'єктно-орієнтованого програмування (ООП) засобами мов PHP та JavaScript.

Практичне значення отриманих результатів полягатиме в тому, що розроблена система буде корисною для власників кав'ярень, допомагаючи їм підтримувати ефективно управління бізнесом та забезпечувати високий рівень обслуговування для своїх клієнтів.

					<i>РП 07. 06 000. 00 ДП ПЗ</i>	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

1 ОСНОВНИЙ РОЗДІЛ

1.1 Аналіз предметної області

1.1.1 Огляд існуючих рішень

Розглянемо наступні аналоги веб-застосунків до веб-застосунку, що розроблюється – “AROMA Kava”, “Coffee Ocean”, “АЛЬТЕРНАТИВА”.

Розглянемо перший аналог веб-застосунку, що розробляється – «AROMA Kava».

“AROMA Kava” – це всеукраїнська мережа кав'ярень, що пропонує широкий вибір продукції за доступними цінами. Працюють у понад 45 містах України [1].

На рис. 1.1. представлено головну сторінку веб-застосунку аналогу “AROMA Kava”.

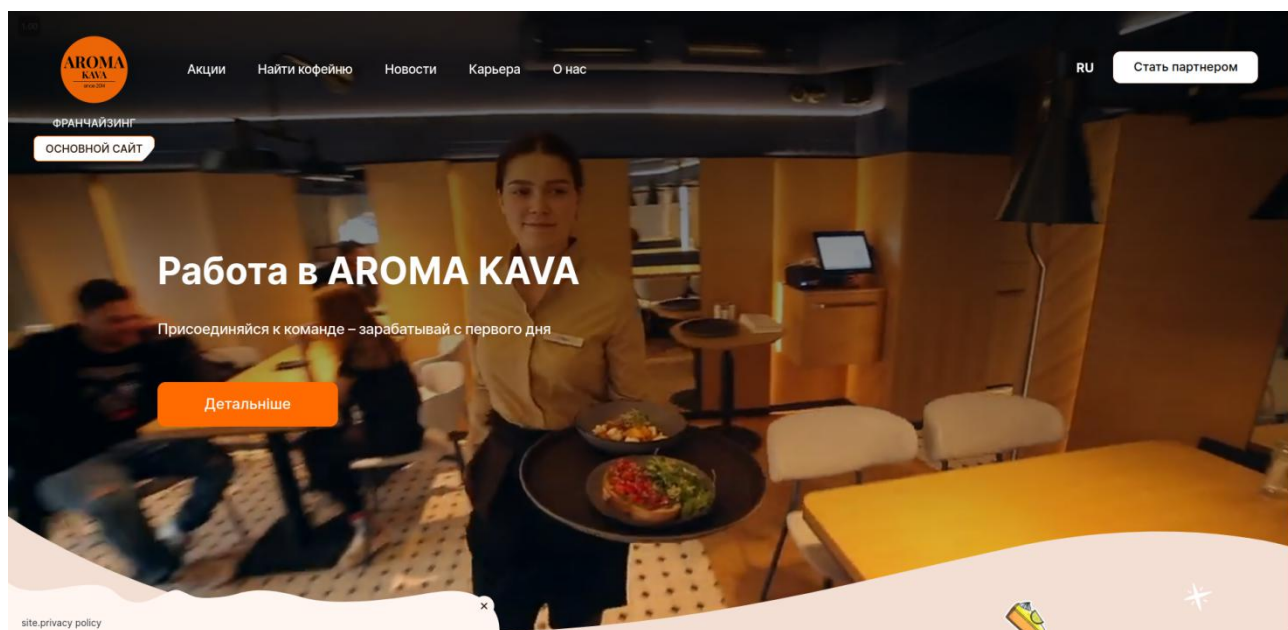


Рисунок 1.1. Головна сторінка веб-застосунку “AROMA Kava”

Розглянемо переваги веб-застосунку “AROMA Kava”:

1. Сучасний анімований графічний інтерфейс: Це додає сайту естетичний вигляд і може залучити увагу користувачів. Анімація може також покращити відчуття взаємодії з сайтом, зробивши його більш привабливим для відвідувачів.

					РП 07. 06 001. 00 ДП ПЗ	Арк.
						9
Ізм.	Лист	№ докум.	Підпис	Дата		

2. Наявність мобільної версії: Це важливий аспект у сучасному світі, оскільки все більше користувачів використовують мобільні пристрої для доступу до Інтернету. Мобільна версія може забезпечити зручний доступ до сайту навіть на різних пристроях.

3. Наявність великої кількості розділів: Це дозволяє користувачам з легкістю знаходити інформацію, яка їм потрібна. Розділи можуть бути структуровані за темами, що полегшує навігацію та зрозуміння контенту.

Розглянемо недоліки веб-застосунку “AROMA Kava”:

1. Перевантажений графічний інтерфейс: Це може призвести до погіршення досвіду користувача, особливо на мобільних пристроях або при повільному Інтернет-з'єднанні. Надмірна кількість графічних елементів може перенавантажити сторінку і зробити її важкою для завантаження та використання.

2. Низька швидкість завантаження: Це може відлякувати користувачів і призвести до втрати трафіку на сайті. Повільне завантаження може бути особливо проблематичним на мобільних пристроях або у випадку зі слабким Інтернет-з'єднанням.

3. Не повна локалізація: Це може створити проблеми для користувачів, які не розуміють мову, на якій представлений сайт. Відсутність повної локалізації може зменшити кількість користувачів і призвести до втрати можливих клієнтів, особливо у випадку, якщо цільова аудиторія має різні мовні переваги.

Розглянемо другий аналог веб-застосунку, що розробляється – “Coffee Ocean”.

“CoffeeOcean” – це компанія, яка надає можливість відкрити кав'ярню за їх франшизою у різних містах, включаючи Одесу, з повним супроводом та підтримкою [2].

На рис. 1.2. представлено головну сторінку веб-застосунку “Coffee Ocean”.

					РП 07. 06 001. 00 ДП ПЗ	Арк.
						10
Ізм.	Лист	№ докум.	Підпис	Дата		

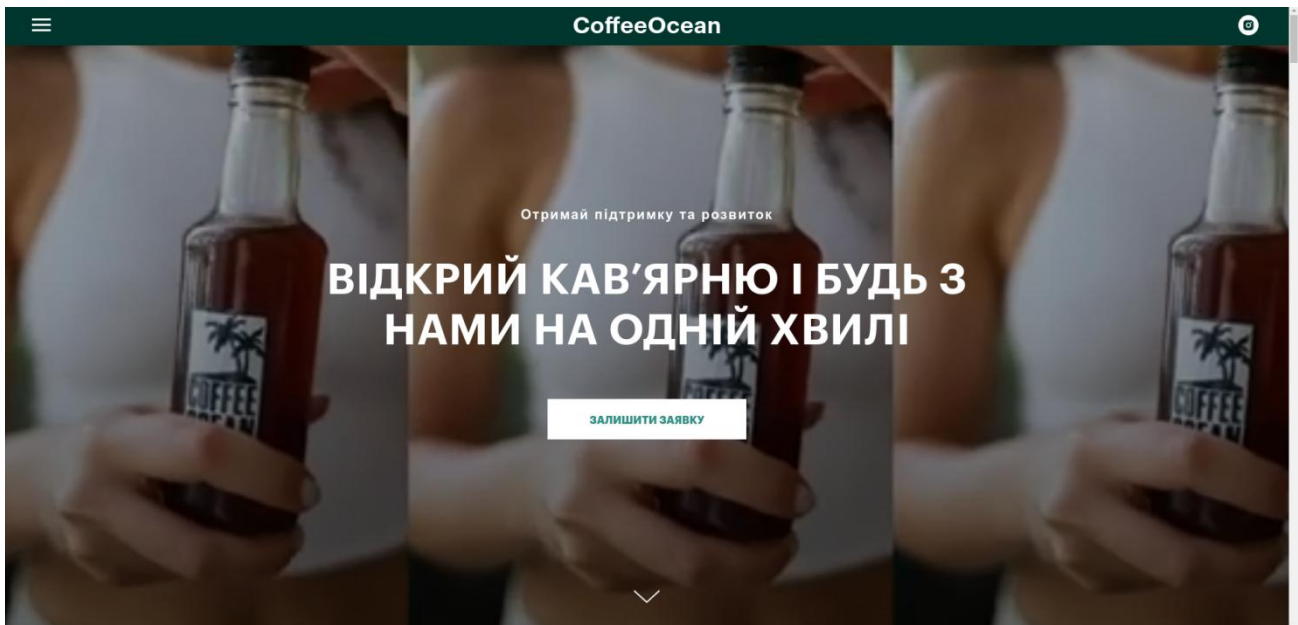


Рисунок 1.2. Головна сторінка веб-застосунку “Coffee Ocean”

Розглянемо переваги веб-застосунку “Coffee Ocean”:

1. Сучасний анімований графічний інтерфейс: Це приваблює користувачів і створює приємне враження від відвідування сайту. Анімація може зробити користувацький досвід більш привабливим та цікавим.
2. Наявність мобільної версії: Це важливо в сучасному світі, де більшість користувачів використовують мобільні пристрої для доступу до Інтернету. Мобільна версія дозволяє зручно переглядати сайт на різних пристроях і забезпечує більш широке охоплення аудиторії.
3. Легкість подання інформації: Якщо інформація легко знаходиться та сприймається користувачами, це полегшує їхню взаємодію з сайтом і позитивно впливає на їх задоволення від використання його сервісів.

Розглянемо недоліки веб-застосунку “Coffee Ocean”:

1. Проблеми з відображенням банеру на десктопній версії: Це може негативно впливати на естетичний вигляд сайту і зменшувати його професійний вигляд. Крім того, це може зменшити ефективність рекламних кампаній, які використовують банери.
2. Відсутність розділів на сайті: Це може ускладнювати навігацію користувачів і зробити пошук інформації менш зручним. Відсутність

					РП 07. 06 001. 00 ДП ПЗ	Арк.
						11
Ізм.	Лист	№ докум.	Підпис	Дата		

структурованості може призвести до того, що користувачі втратять цікаві для них розділи або не зможуть знайти потрібну інформацію.

3. Відсутність детальної інформації про меню: Це може робити сайт менш привабливим для користувачів, які шукають конкретні страви або напої. Недостатня інформація про меню може зменшити інтерес користувачів до відвідування кав'ярні, оскільки вони не зможуть заздалегідь з'ясувати, що саме пропонується.

Розглянемо третій аналог веб-застосунку, що розробляється – “АЛЬТЕРНАТИВА”.

“АЛЬТЕРНАТИВА” – це затишне місце, де можна насолодитися ароматною кавою та смачними солодощами. Розташована у двох зручних локаціях: на Черьомушках за адресою Академіка Філатова, 24, та на Таїрові за адресою Академіка Вільямса, 68/1; ідеальне місце для зустрічей з друзями або робочих зустрічей, адже вона працює щодня з 08:00 до 22:00, що дозволяє завітати сюди в будь-який час дня [3].

На рис. 1.3. представлено головну сторінку веб-застосунку “АЛЬТЕРНАТИВА”.

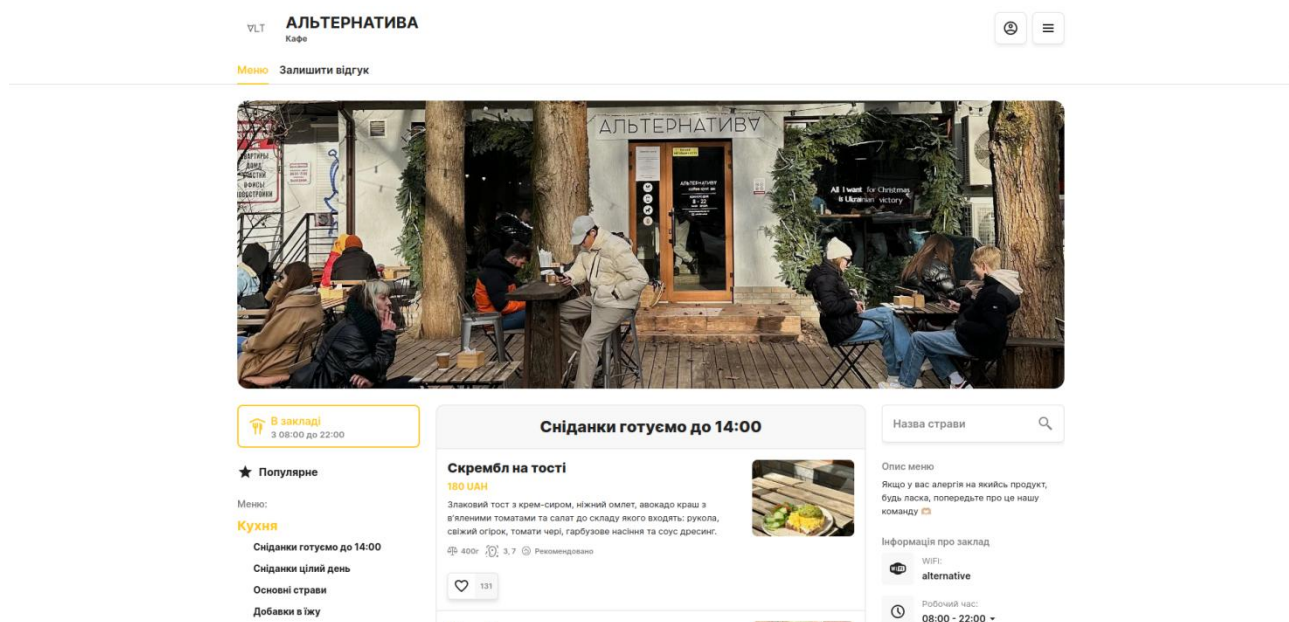


Рисунок 1.3. Головна сторінка веб-застосунку “АЛЬТЕРНАТИВА”

Розглянемо переваги веб-застосунку “АЛЬТЕРНАТИВА”:

					РП 07. 06 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		12

1. Мінімалістичний інтерфейс: Мінімалістичний дизайн може створити чисте та привабливе візуальне враження для відвідувачів. Він спрощує навігацію та робить сайт більш зрозумілим для користувачів.
2. Детальне меню та інформація: Наявність детального меню та інформації допомагає користувачам швидко знайти те, що їх цікавить, і зробити правильний вибір. Це може підвищити задоволення клієнтів та їх лояльність.

Розглянемо недоліку веб-застосунку “АЛЬТЕРНАТИВА”:

1. Використання платформи для створення сайтів: Використання платформи для створення сайтів може обмежити можливості розширення та налаштування. Це може призвести до того, що сайт виглядатиме стандартно та може бути менш унікальним у порівнянні з іншими сайтами. Крім того, певні функції або налаштування можуть бути обмежені або складні у реалізації через обмежені можливості платформи.

Розглянемо порівняльну таблицю аналогів – “AROMA Kava”, “Coffee Ocean” “АЛЬТЕРНАТИВА”.

У табл. 1.1. представлено порівняння аналогів.

Таблиця 1.1. Порівняльна характеристика аналогів

Характеристика	AROMA Kava	Coffee Ocean	АЛЬТЕРНАТИВА
Наявність реєстрації	ні	ні	так
Наявність форми зворотнього зв'язку	так	так	так
Наявність детальної інформації	так	ні	так
Наявність реклами	ні	ні	ні
Наявність української мови	так	так	так
Сайт використовує власний двигун	так	так	ні

Таким чином, після проведення аналізу, можна зробити висновок, що аналоги, що розглядаються, не є однозначними. Під час розробки цільового результуючого веб-застосунку будуть враховані всі недоліки вищезазначених веб-застосунків.

1.1.2 Технології та засоби розробки

Для розробки графічного інтерфейсу веб-застосунку (UI/UX) застосовуватимуться HTML, CSS та Bootstrap.

HTML створює структуру та зміст веб-сторінок [4].

CSS відповідає за зовнішній вигляд, зокрема розміщення, кольори та шрифти [5].

Bootstrap спрощує створення адаптивних та стильних веб-сайтів завдяки використанню HTML, CSS та JavaScript [6].

Для розробки клієнтської частини веб-застосунку (Front-End) застосовуватимуться JavaScript та бібліотека jQuery.

JavaScript додає інтерактивність на веб-сторінках, дозволяючи взаємодіяти з користувачем, маніпулювати DOM та виконувати різноманітні завдання [7].

jQuery спрощує маніпулювання HTML-елементами, обробку подій, анімацію та роботу з AJAX [8].

Для розробки серверної частини веб-застосунку (Back-End) застосовуватимуться PHP, SQL та СУБД MySQL.

PHP широко застосовується для створення динамічного контенту, обробки форм та роботи з базами даних [9].

SQL взаємодіє з реляційними базами даних, виконуючи операції додавання, видалення, модифікації та отримання даних [10].

MySQL зберігає дані у веб-застосунках та інших програмах, що використовують реляційні бази даних, і є однією з найпопулярніших відкритих СУБД [11].

Розглянемо обрані технології більш наглядно за допомогою структурної схеми.

На рис. 1.4 представлено структурну схему технологій веб-застосунку, що буде розроблено.

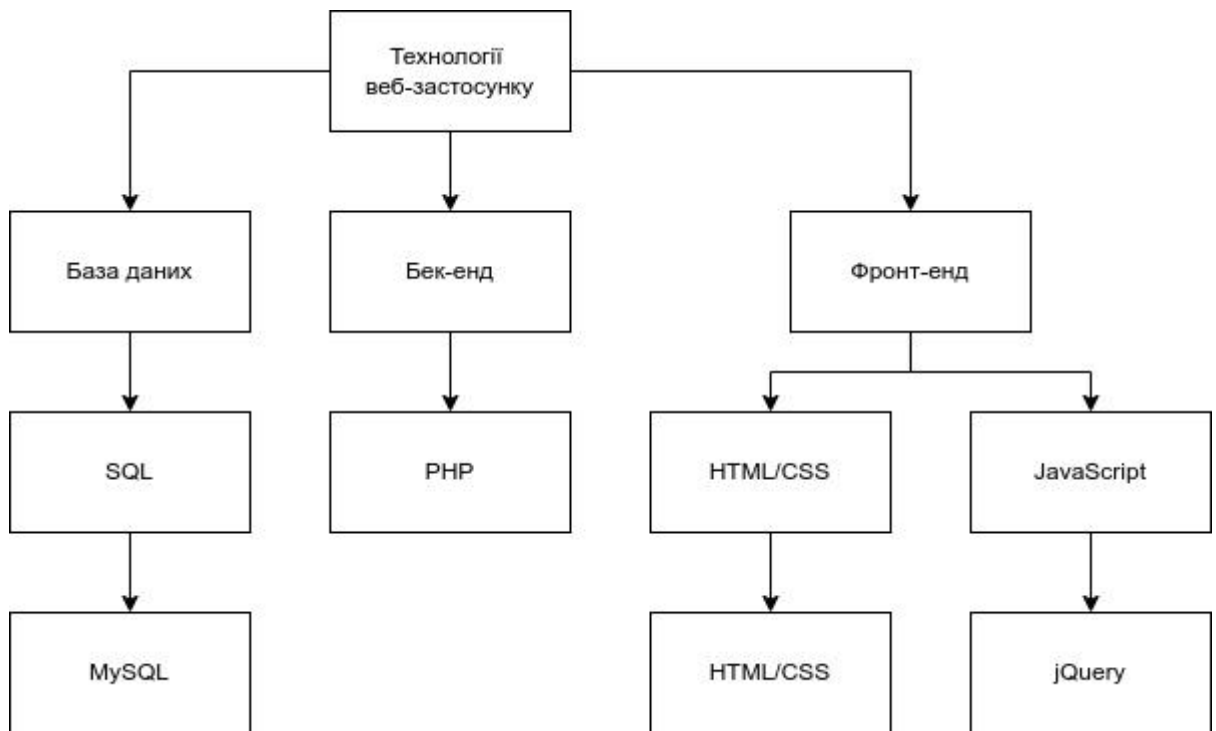


Рисунок 1.4. Структурна схема обраних технологій

Розглянемо більш детально обрану технологію для створення веб-системи – мови програмування PHP.

Переваги мови програмування PHP:

1. Простота вивчення та використання: PHP має простий синтаксис, що полегшує вивчення для початківців.
2. Широке застосування: PHP використовується для створення різноманітних веб-проектів, від простих веб-сайтів до складних веб-застосунків.
3. Велика спільнота: Існує велика спільнота PHP-розробників, яка надає підтримку, розв'язання проблем та безліч корисних бібліотек і фреймворків.
4. Швидкодія: PHP може бути досить швидкодіючим, особливо за умови оптимізації та використання кешування.
5. Інтеграція з базами даних: PHP легко інтегрується з багатьма системами управління базами даних, такими як MySQL, PostgreSQL, SQLite тощо.

Недоліки мови програмування PHP:

1. Спадкові проблеми: PHP почав свій шлях як скриптова мова, тому має деякі спадкові проблеми, такі як неявність суворої типізації.
2. Безпека: Історично PHP був відомий своїми проблемами з безпекою. Вразливості можуть виникати через недостатні перевірки даних, SQL-ін'єкції та інші причини.
3. Швидкодія: Хоча PHP може бути швидкодієним, в порівнянні з іншими мовами веб-програмування, такими як Node.js або Go, він може виявитися менш продуктивним в деяких випадках.
4. Масштабованість: В деяких випадках PHP може мати проблеми з масштабуванням для великих проєктів через відсутність стандартних рішень для обробки багатofункціональних застосунків.
5. Асинхронність: PHP не є найкращим вибором для асинхронних операцій, які можуть бути важливими для деяких сучасних веб-застосунків.

Не зважаючи на ці недоліки, PHP залишається дуже популярною мовою для веб-розробки завдяки своїй простоті, широкому спектру застосування і великій спільноті розробників.

Так як мови HTML, CSS та JavaScript є монополістами у своїх нішах, розглянемо порівняльну таблицю мови програмування PHP з однією з найпопулярніших мов програмування – C#.

У табл. 1.2 представлено порівняння аналогів.

Таблиця 1.2. Порівняльна характеристика мов програмування

Характеристика	PHP	C#
Простота вивчення	Легко вивчити для початківців	Складніша для початківців
Синтаксис	Простий, нестрогий	Сучасний, строгий

Веб-розробка	Основне призначення	Може використовуватись
Широке застосування	Дуже поширений у веб-розробці	Використовується у різних сферах
Спільнота	Велика та активна	Велика та активна
Безпека	Історично проблеми з безпекою	Високий рівень безпеки
Масштабованість	Може мати проблеми	Добре масштабований
Швидкодія	Може бути швидким	Висока продуктивність
Підтримка баз даних	Легка інтеграція з багатьма БД	Потужні інструменти для роботи з БД
Платформа	Переважно серверна (LAMP)	Кросплатформенний (Windows, Linux, macOS)
Розширюваність	Велика кількість бібліотек і фреймворків	Велика кількість бібліотек і фреймворків
Асинхронність	Слабка підтримка	Потужна підтримка асинхронності
Підтримка ООП	Основні концепції підтримуються	Повна підтримка ООП
Середовище розробки	Багато доступних IDE	Потужні IDE (Visual Studio)

Ізм.	Лист	№ докум.	Підпис	Дата

Підтримка корпорацій	Відкрите співтовариство	Підтримка Microsoft
Документація	Велика кількість ресурсів	Високоякісна документація

Загалом, даний вибір технологій зумовлений тим, що всі вищезазначені технології активно використовуються по всьому світу та займають високі позиції у топах інформаційних технологій.

Для цього, щоб в цьому переконатися, розглянемо статистику від сервісу «statista».

Statista надає онлайн-платформу для статистичних даних і маркетингових досліджень. Вона пропонує доступ до великої кількості статистичних даних з різних галузей і ринків, що можуть бути корисними для бізнесу, академічних досліджень та інших видів аналізу.

Розглянемо статистику найбільш використовуваних мов програмування серед розробників у всьому світі станом на 2023 рік.

На рис. 1.5. представлено статистику найбільш використовуваних мов програмування серед розробників у всьому світі станом на 2023 рік від онлайн-платформи «statista».

Technology & Telecommunications > Software

Most used programming languages among developers worldwide as of 2023

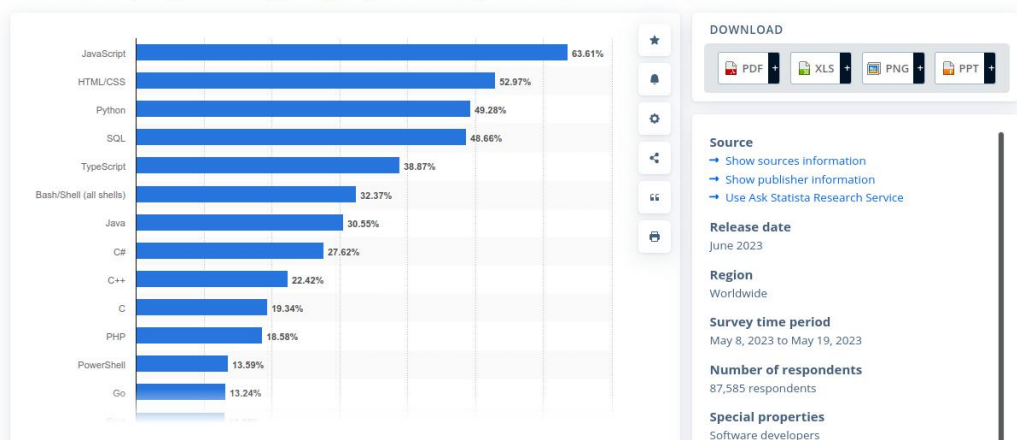


Рисунок 1.5. Статистика популярності мов програмування від сервісу «statista»

Таким чином, зважаючи на вищезазначену інформацію, можна стверджувати, що обрані технології входять до топ-10 і мають велику спільноту по всьому світу, тому їх вибір є доцільним.

1.2 Проєктування застосунку

1.2.1 Технічне завдання на розробку

До веб-застосунку, що розробляється, ставляться наступні вимоги:

1. Наявність мінімалістичного привабливого графічного інтерфейсу (GUI):
Вимога до мінімалістичного дизайну передбачає створення інтерфейсу, який зосереджується на основних функціях і мінімізує зайві деталі. Це забезпечує естетично приємний вигляд програми, не завантажуючи користувача зайвою інформацією. Оптимізація інтерфейсу для мінімалізму дозволяє користувачам швидко зорієнтуватися та ефективно використовувати програму.
2. Наявність простого інтуїтивно зрозумілого досвіду користувача (UX):
Забезпечення простоти та інтуїтивної зрозумілості в UX дозволяє користувачам легко взаємодіяти з програмою без необхідності додаткових пояснень. Це досягається шляхом розташування елементів у логічних місцях, використання зрозумілих піктограм та текстових пояснень, а також врахування типових сценаріїв взаємодії користувачів.

На рис. 1.6 представлено схему розділів навігації веб-застосунку, що буде розроблено.

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
						19
Ізм.	Лист	№ докум.	Підпис	Дата		

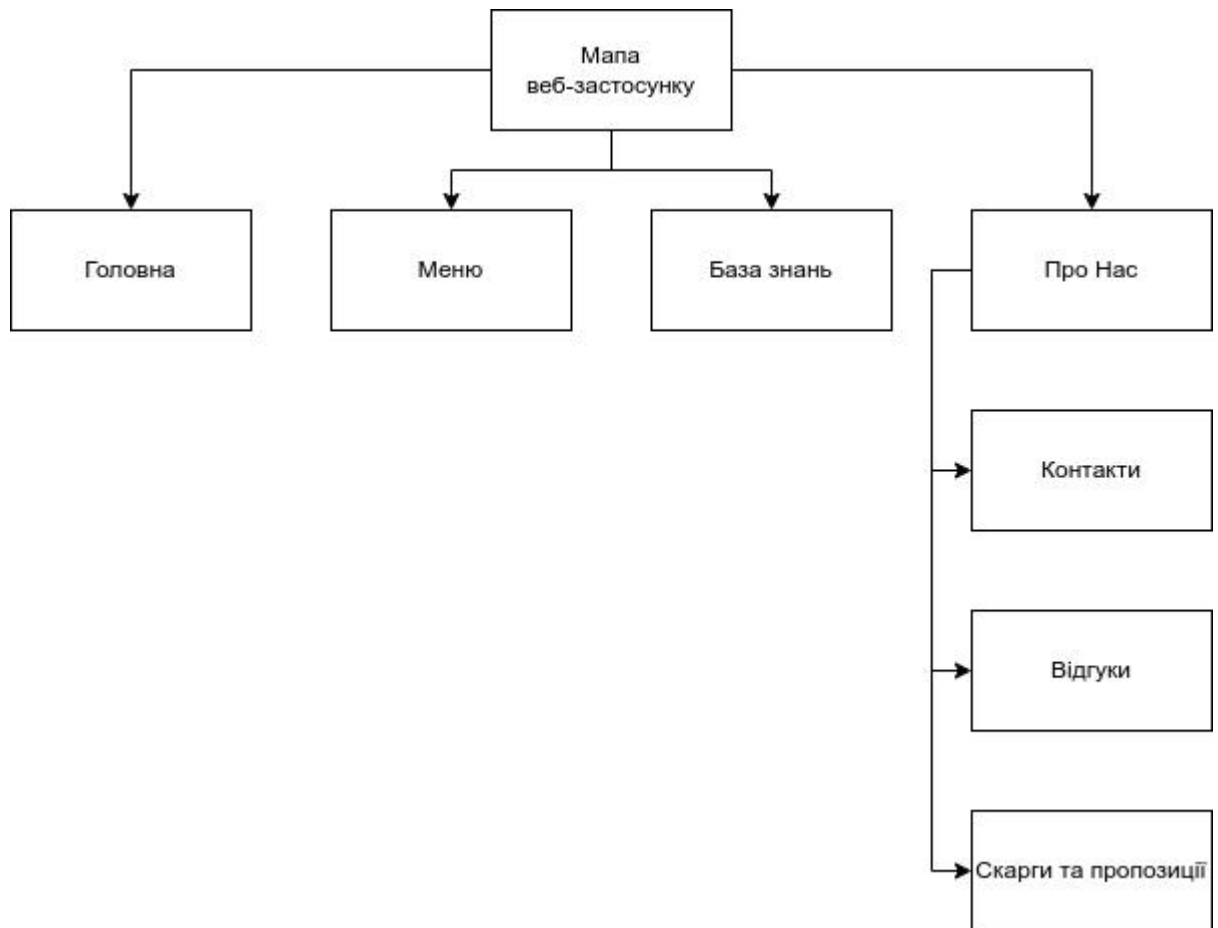


Рисунок 1.6. Схема розділів навігації веб-застосунку, що буде розроблено

3. Наявність системи керування контентом (CMS) для гнучкого управління вмістом: Використання CMS спрощує процес управління вмістом для адміністраторів, дозволяючи їм швидко та ефективно додавати, редагувати та видаляти контент без необхідності втручання програмістів. Ця функціональність забезпечує гнучкість та швидкість в управлінні вмістом застосунку, що є ключовим для його успішного використання.

1.2.2 Проектування дизайну застосунку

При замовленні веб-сайту замовник очікує представлення прототипів для подальшого узгодження. Ці прототипи відіграють ключову роль у визначенні дизайну та функціональності сайту перед його реалізацією. Під час створення прототипів потрібно враховувати технічне завдання та предметну область. Крім того, макети, як мінімум для основних елементів, таких як футер, навігаційне меню тощо, розробляються у програмі Adobe Photoshop. Ці макети потім

передаються фронт-енд розробникам або верстальникам для подальшої реалізації.

Розглянемо дизайн навігаційного меню веб-застосунку, що розробляється.

Навігаційне меню являє собою візуальний елемент, який містить посилання на основні розділи або сторінки сайт.

На рис. 1.7. представлено дизайн навігаційного меню веб-застосунку, що розробляється.



Рисунок 1.7. Навігаційна меню веб-застосунку, що розробляється

Розглянемо дизайн хедер веб-застосунку, що розробляється.

Хедер являє собою верхню частину веб-сторінки, яка може містити різноманітні елементи, що повторяються на кожній сторінці, в даному випадку він представлений у вигляді слайдеру.

Слайдер являє собою інтерактивний елемент на веб-сторінці, що містить карусель зображень або контенту, який автоматично або за допомогою користувача перемикається між різними елементами, презентуючи їх.

На рис. 1.8 представлено дизайн хедеру веб-застосунку, що розробляється.



Рисунок 1.8. Дизайн хедеру веб-застосунку, що розробляється

Розглянемо дизайн футеру веб-застосунку, що розробляється.

Футер являє собою нижню частину веб-сторінки, яка містить додаткову інформацію, таку як контактні дані, посилання на соціальні мережі, копірайт і інші додаткові елементи.

На рис. 1.9 представлено дизайн футеру веб-застосунку, що розробляється.

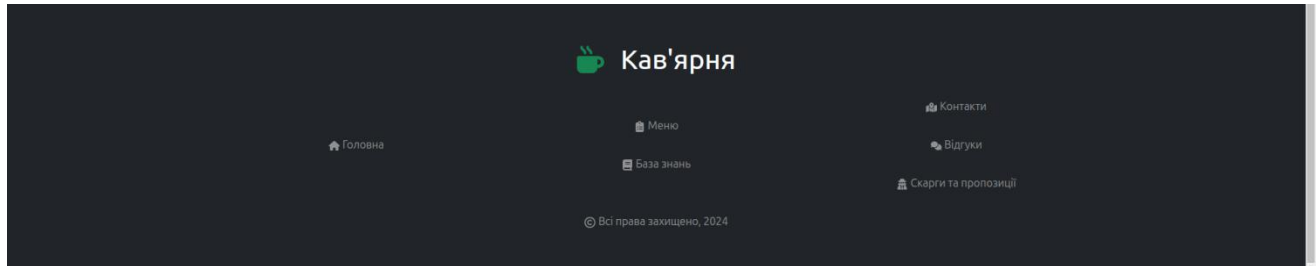


Рисунок 1.9. Дизайн футеру веб-застосунку, що розробляється

1.2.3 Проектування бази даних

Під час проектування бази даних було обрано технології реляційних баз даних. Реляційна база даних (РБД) являє собою сукупність даних, організованих у відповідності з моделлю даних, відомою як реляційна модель. У реляційній моделі дані представлені у вигляді таблиць, які складаються з рядків і стовпців. Кожен рядок таблиці представляє запис або кортеж даних, а кожний стовпець - атрибут або поле. Реляційні бази даних можуть мати відносини або зв'язки між таблицями, що дозволяє ефективно виражати складність даних та їх взаємозв'язки.

Розглянемо схему базу даних, що було створено в phpMyAdmin. phpMyAdmin являє собою веб-інструмент з відкритим кодом, призначений для адміністрування MySQL і MariaDB баз даних через веб-інтерфейс.

На рис. 1.10 представлено схему бази даних веб-застосунку, що розробляється.

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		22

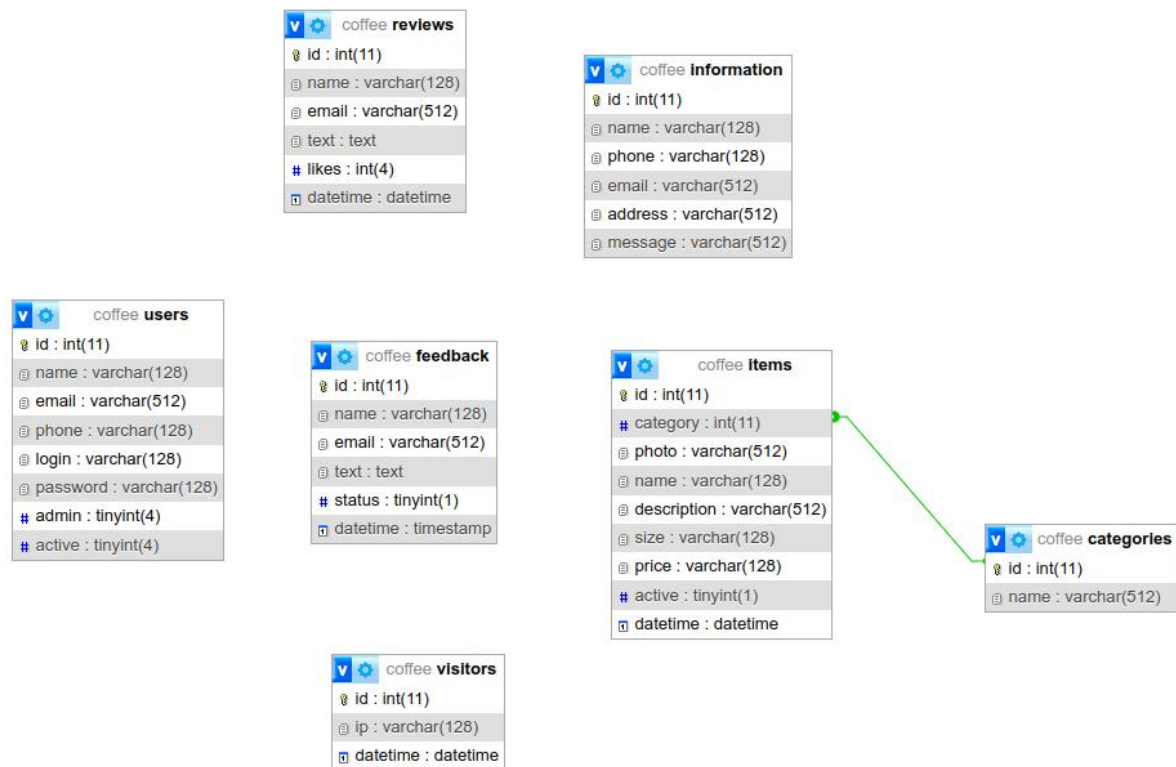


Рисунок 1.10. Схема бази даних, що розробляється

Розглянемо структуру бази даних веб-застосунку, що розробляється, більш детально.

Таблиця “information” містить інформацію про інформацію у веб-системі.

У табл. 1.3 представлено опис таблиці “information”.

Таблиця 1.3. Опис таблиці “information”

Поле	Тип	Коментар
id	INT	Ідентифікатор запису.
name	VARCHAR	Назва закладу.
phone	VARCHAR	Телефон закладу.
email	VARCHAR	Email закладу.
address	VARCHAR	Адреса закладу.
message	VARCHAR	Повідомлення від закладу.

Таблиця “users” містить інформацію про користувачів у веб-системі.

У табл. 1.4 представлено опис таблиці “users”.

Таблиця 1.4. Опис таблиці “users”

Поле	Тип	Коментар
id	INT	Ідентифікатор користувача.
name	VARCHAR	Ім'я користувача.
email	VARCHAR	Email користувача.
phone	VARCHAR	Телефон користувача.
login	VARCHAR	Логін користувача.
password	VARCHAR	Хеш пароля користувача.
admin	TINYINT	Прапорець адміністратора.
active	TINYINT	Статус активності облікового запису користувача.

Таблиця “categories” містить інформацію про категорії меню.

У табл. 1.5 представлено опис таблиці “categories”.

Таблиця 1.5. Опис таблиці “categories”

Поле	Тип	Коментар
id	INT	Ідентифікатор категорії.
name	VARCHAR	Назва категорії.

Таблиця “items” містить інформацію про меню.

У табл.1.6 представлено опис таблиці “items”.

Таблиця 1.6. Опис таблиці “items”

Поле	Тип	Коментар
id	INT	Ідентифікатор товару.
category	INT	Ідентифікатор категорії товару (зв'язок з категоріями).
photo	VARCHAR	Шлях до зображення товару.

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 06 001. 00 ДП ПЗ

Арк.

24

Продовження таблиці 1.6

name	VARCHAR	Назва товару.
description	VARCHAR	Опис товару.
size	VARCHAR	Розмір або об'єм товару.
price	VARCHAR	Ціна товару.
active	TINYINT	Статус активності товару.
datetime	DATETIME	Дата та час додавання товару.

Таблиця “reviews” містить інформацію про відгуки.

У табл. 1.7 представлено опис таблиці “reviews”.

Таблиця 1.7. Опис таблиці “reviews”

Поле	Тип	Коментар
id	INT	Ідентифікатор відгуку.
name	VARCHAR	Ім'я автора відгуку.
email	VARCHAR	Email автора відгуку.
text	TEXT	Текст відгуку.
likes	INT	Кількість лайків для відгуку.
datetime	DATETIME	Дата та час створення відгуку.

Таблиця “feedback” містить інформацію про зворотній зв'язок.

У табл. 1.8 представлено опис таблиці “feedback”.

Таблиця 1.8. Опис таблиці “feedback”

Поле	Тип	Коментар
id	INT	Ідентифікатор відгуку.
name	VARCHAR	Ім'я автора відгуку.
email	VARCHAR	Email автора відгуку.

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 06 001. 00 ДП ПЗ

Арк.

25

text	TEXT	Текст відгуку.
status	TINYINT	Статус відгуку.

Таблиця “visitors” містить інформацію про відвідувачів веб-системи.

У табл. 1.9 представлено опис таблиці “visitors”.

Таблиця 1.9 Опис таблиці “visitors”

Поле	Тип	Коментар
id	INT	Ідентифікатор відвідувача.
ip	VARCHAR	IP-адреса відвідувача.
datetime	DATETIME	Дата та час візиту.

Розглянемо список створених таблиць у phpMyAdmin. Таким чином маємо 7 таблиць.

На рис. 1.11 представлено список створених таблиць в phpMyAdmin.

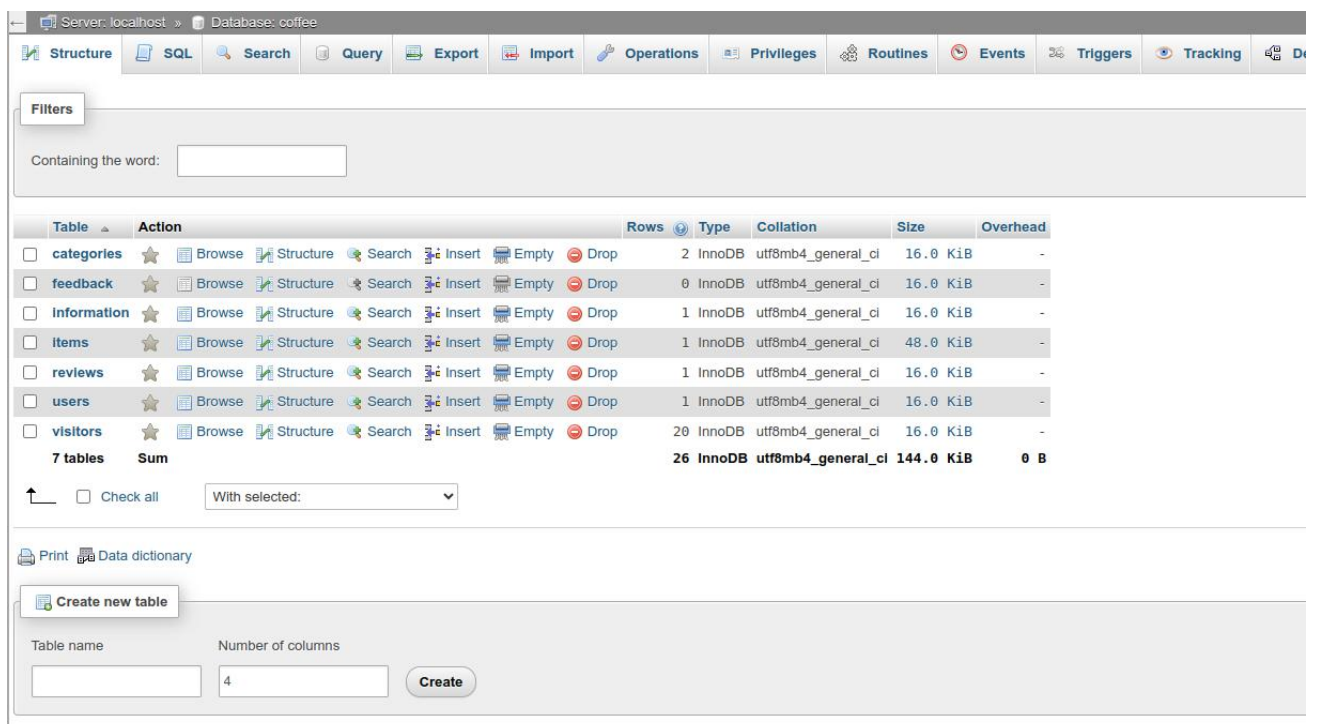


Рисунок 1.11. Список створених таблиць в phpMyAdmin

1.2.4 Проєктування архітектури застосунку

Даний веб-застосунок ілюструє монолітну архітектурну концепцію, що означає, що клієнтська та серверна складові не розділені та не функціонують як окремі сутності. Замість цього, вони взаємодіють як єдиний компонент.

Цей тип архітектури сприяє використанню об'єктно-орієнтованого підходу до програмування (ООП) та застосуванню шаблону проєктування "Модель-Представлення(Вид)-Контролер" (MVC).

У монолітній архітектурі один клас моделі відповідає за взаємодію з однією таблицею бази даних та управляє SQL запитам та операціями CRUD (створення, читання, оновлення, видалення).

Кожен файл представлення (вигляду) відображає макет графічного інтерфейсу, написаний мовами HTML/CSS/JS.

Один клас контролера встановлює зв'язок між даними та графічним інтерфейсом. Ця взаємодія реалізується через протокол HTTP та його методи GET та POST.

Оскільки це монолітна архітектура, веб-застосунок використовує мультисторінковий підхід, де кожен розділ на сайті представляє окрему сторінку, що призводить до перезавантаження сторінок при їх навігації.

Таким чином, загальну архітектуру веб-застосунку, що розробляється, можна представити наступним чином [12, 13].

На рис. 1.12 представлено загальну архітектуру веб-застосунку, що розробляється.

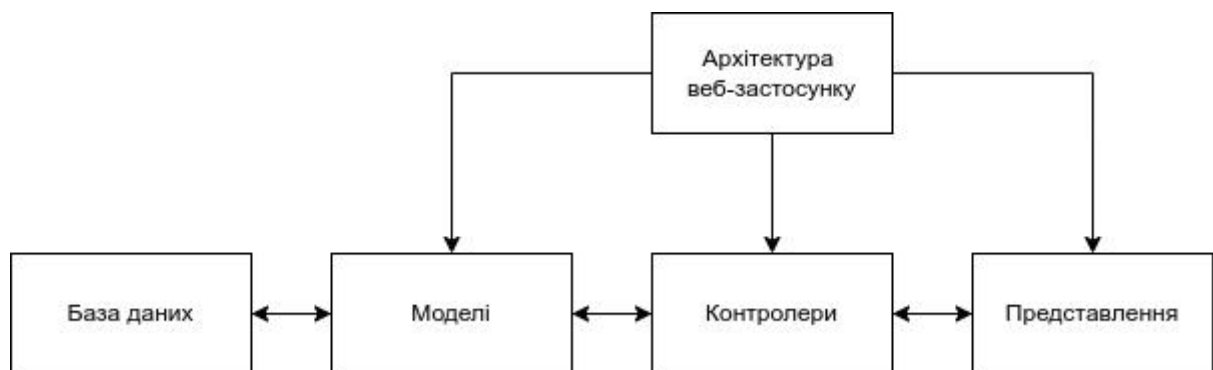


Рисунок 1.12. Загальна архітектура веб-застосунку, що розробляється

Ізм.	Лист	№ докум.	Підпис	Дата

Для реалізації веб-застосунку за даними параметрами використовується мікро-фреймворк PHP з наступною архітектурою.

На рис. 1.13 представлено файлову архітектуру мікро-фреймворку PHP.

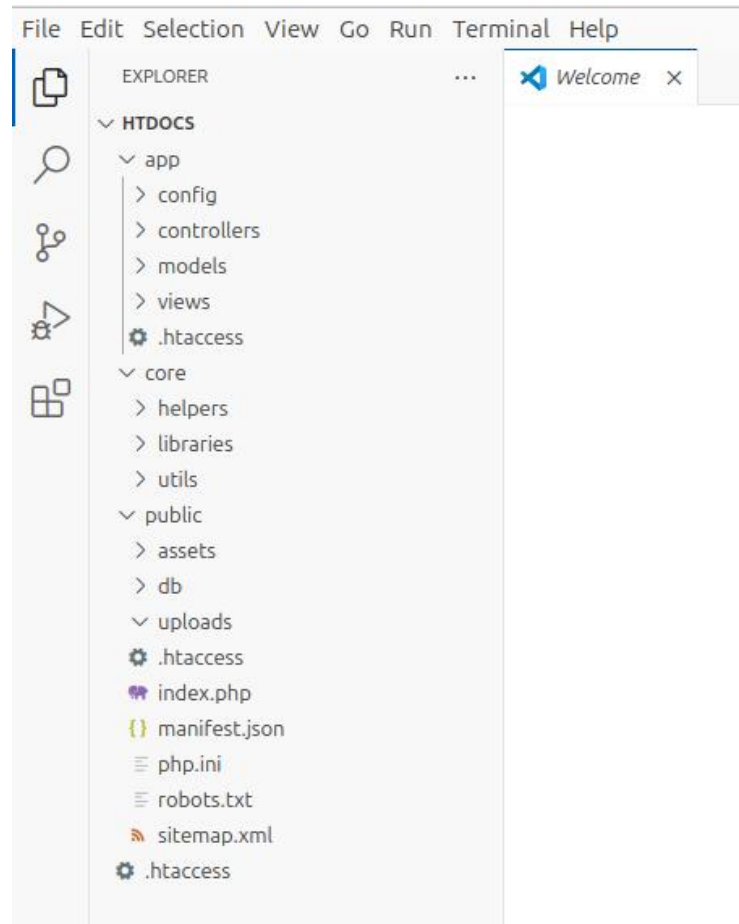


Рисунок 1.13. Файлова архітектура мікро-фреймворку PHP

Розглянемо файлову архітектуру мікро-фреймворку PHP більш детально.

./app: Ця папка містить основний застосунок. Вона містить логіку застосунку, включаючи моделі, контролери та представлення.

./app/views: У цій папці розташовані файли, які відповідають за представлення (View) в шаблонах MVC. Тут зазвичай знаходяться файли HTML, CSS та JavaScript, що відображають дані користувачам.

./app/models: Модель (Model) в MVC відповідає за взаємодію з базою даних та обробку даних. У цій папці зазвичай знаходяться файли, які представляють різні об'єкти даних та виконують логіку доступу до даних.

`./app/controllers`: Контролер (Controller) в MVC призначений для обробки запитів користувача та виклику відповідних методів моделі та представлення для відображення результатів. У цій папці зазвичай знаходяться файли, які містять класи контролерів.

`./app/config`: У цій папці зазвичай знаходяться файли конфігурації застосунку, такі як файли налаштувань маршрутизації, бази даних, тощо.

`./core`: Ця папка містить основний код фреймворка. Вона може включати в себе допоміжні класи, функції та бібліотеки, необхідні для роботи фреймворка.

`./core/helpers`: У цій папці зазвичай знаходяться допоміжні функції або класи, які можуть бути використані для зручності розробки.

`./core/utils`: Ця папка містить допоміжні утиліти або класи, які надають корисні функції, які можуть бути використані в різних частинах фреймворка.

`./core/libraries`: Тут зазвичай знаходяться сторонні бібліотеки або компоненти, які використовуються в застосунку.

`./public`: Ця папка містить ресурси, які доступні публічно через веб-сервер. Це може включати файли CSS, JavaScript, зображення тощо.

`./public/uploads`: У цій папці зазвичай зберігаються файли, які користувачі завантажують на сервер.

`./public/assets`: Тут зазвичай знаходяться статичні ресурси, такі як стилі, скрипти тощо, які використовуються в клієнтському коді.

`./public/db`: Ця папка може містити файли резервних копій бази даних або інші файли, пов'язані з базою даних.

1.3 Реалізація застосунку

1.3.1 Налаштування бази даних

У файлі `/app/config/config.php` встановимо з'єднання з СУБД MySQL. Нижче представлено код, що встановлює з'єднання з СУБД MySQL.

```
define('DB_HOST', '127.0.0.1');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'coffee');
```

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
						29
Ізм.	Лист	№ докум.	Підпис	Дата		

Цей файл є конфігураційним файлом для веб-застосунку на основі PHP. Він містить налаштування для підключення до бази даних, визначає шляхи до різних каталогів і статичних ресурсів, а також встановлює назву сайту і версію застосунку.

1.3.2 Налаштування роутингу

Роутінг являє собою механізм веб-застосунку, який визначає, який код має бути виконаний для кожного запиту користувача до сервера на підставі URL-адреси, що була відправлена.

У файлі `/app/config/routes.php` налаштуємо роутинг MVC веб-застосунку. Нижче представлено код, що описує роутинг MVC веб-застосунку.

```
<?php
// Routes array
return array(
    '' => 'home/index',
    'home' => 'home/index',
    'menu' => 'menu/index',
    'contacts' => 'contacts/index',
    'feedback' => 'feedback/index',
    'feedback/addFeedback' => 'feedback/addFeedback',
    'reviews' => 'reviews/index',
    'reviews/([0-9]+)' => 'reviews/index/$1',
    'reviews/addReview' => 'reviews/addReview',
    'reviews/updateReview/([0-9]+)' => 'reviews/updateReview/$1',
    'knowledge' => 'knowledge/index',
    'login' => 'login/index',
    'login/signUp' => 'login/signUp',
    'login/signIn' => 'login/signIn',
    'login/signOut' => 'login/signOut',
    'admin' => 'admin/index',
    'admin/updateInformation' => 'admin/updateInformation',
    'admin/createCategory' => 'admin/createCategory',
    'admin/updateCategory' => 'admin/updateCategory',
    'admin/deleteCategory' => 'admin/deleteCategory',
    'admin/createItem' => 'admin/createItem',
    'admin/updateItem' => 'admin/updateItem',
    'admin/deleteItem' => 'admin/deleteItem',
    'admin/deleteReview' => 'admin/deleteReview',
    'admin/updateFeedback' => 'admin/updateFeedback',
    'admin/deleteFeedback' => 'admin/deleteFeedback',
);
```

Цей код визначає маршрути, за якими буде переадресовано користувача при різних запитах URL-адрес. Він допомагає системі зрозуміти, який код потрібно виконати для кожного запиту користувача.

1.3.3 Створення моделей (Models)

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
						30
Ізм.	Лист	№ докум.	Підпис	Дата		

Створимо класи моделей для сутностей бази даних.

Модель (Model) являє собою клас, що відповідає за представлення даних та бізнес-логіку програми. Вона відокремлює логіку застосунку від користувацького інтерфейсу та забезпечує управління даними.

На рис. 1.14 представлено створені моделі для веб-застосунку.

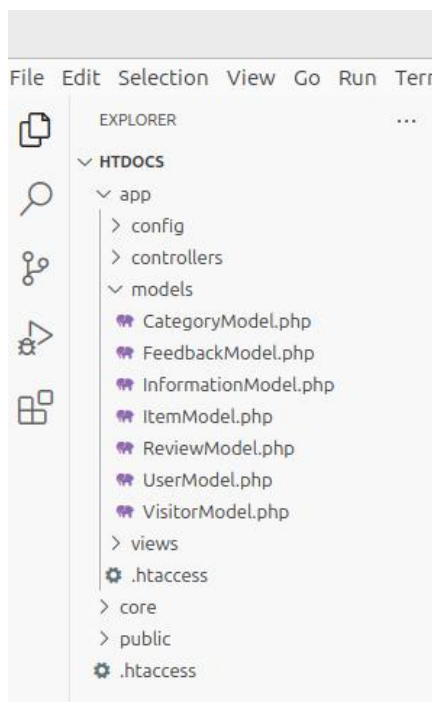


Рисунок 1.14. Створені моделі в MVC проєкті

Розглянемо клас типової моделі, наприклад для таблиці items, що відповідає за меню.

Нижче представлено код клас моделі для основної таблиці.

```
<?php
class ItemModel
{
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function selectAll()
    {
        // Query
        $this->db->query('SELECT * FROM items');
        // Execute
        $results = $this->db->resultSet();
        // Return result
        return $results;
    }
}
```

```

}

public function selectByCategory($category)
{
    // Query
    $this->db->query('SELECT * FROM items WHERE category = :category');
    // Bind value
    $this->db->bind(':category', $category);
    // Execute
    $results = $this->db->resultSet();
    // Return result
    return $results;
}

public function insert($data)
{
    // Query
    $this->db->query('INSERT INTO items (category, photo, name, description,
size,
                                price,
                                active)
VALUES(:category, :photo, :name, :description, :size, :price, :active)');
    // Bind values
    $this->db->bind(':category', $data['category']);
    $this->db->bind(':photo', $data['photo']);
    $this->db->bind(':name', $data['name']);
    $this->db->bind(':description', $data['description']);
    $this->db->bind(':size', $data['size']);
    $this->db->bind(':price', $data['price']);
    $this->db->bind(':active', $data['active']);
    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}

public function update($data)
{
    // Query
    $this->db->query('UPDATE items SET category = :category, photo = :photo,
name = :name, description = :description, size = :size, price = :price, active = :active
WHERE id = :id');
    // Bind values
    $this->db->bind(':id', $data['id']);
    $this->db->bind(':category', $data['category']);
    $this->db->bind(':photo', $data['photo']);
    $this->db->bind(':name', $data['name']);
    $this->db->bind(':description', $data['description']);
    $this->db->bind(':size', $data['size']);
    $this->db->bind(':price', $data['price']);
    $this->db->bind(':active', $data['active']);
    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {

```

					РП 07. 06 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		32

```

        return false;
    }
}

public function delete($data)
{
    // Query
    $this->db->query('DELETE FROM items WHERE id = :id');
    // Bind values
    $this->db->bind(':id', $data['id']);
    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}
}
?>

```

Цей PHP-код визначає клас ItemModel, який взаємодіє з базою даних для управління записами у таблиці items.

Конструктор класу ініціалізує об'єкт Database для роботи з базою даних.

Метод selectAll отримує всі записи з таблиці items.

Метод selectByCategory отримує записи з таблиці items, які відповідають певній категорії, використовуючи параметр category.

Метод insert додає новий запис до таблиці items, прив'язуючи відповідні значення з масиву data.

Метод update оновлює існуючий запис у таблиці items на основі ідентифікатора id, тоді як метод delete видаляє запис з таблиці items за ідентифікатором id.

1.3.4 Створення представлень (Views)

Створимо файли представлень.

Представлення (View) являє собою файл, що представляє представляє інтерфейс користувача у вигляді HTML-сторінок.

Представлення включають в себе шаблонізатор.

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
						33
Ізм.	Лист	№ докум.	Підпис	Дата		

Шаблонізатор являє собою механізм у веб-розробці, який дозволяє розділити веб-сторінку на частини, такі як заголовок, навбар, хедер, футер та управляти цими частинами окремо.

На рис. 1.15 представлено створений шаблонізатор з зіставними частинами.

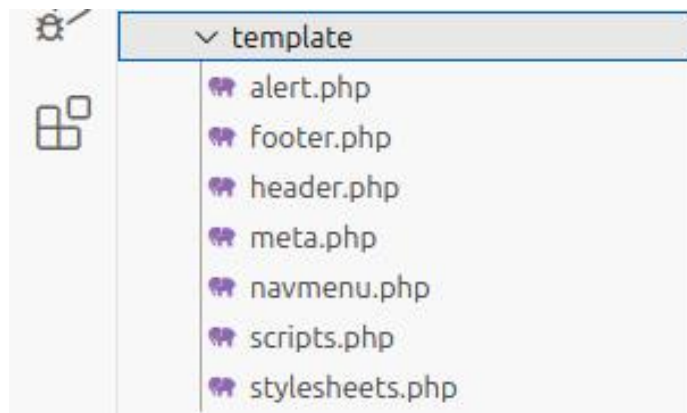


Рисунок 1.15. Шаблонізатор з зіставними частинами

Опишемо детально кожен файл.

Файл alert.php може створювати різноманітні повідомлення або сповіщення для відображення користувачу.

Файл footer.php відповідає за створення нижньої частини веб-сторінки, яка містить загальну інформацію або посилання.

Файл header.php відповідає за створення верхньої частини веб-сторінки, яка містить слайдер.

Файл meta.php може генерувати HTML-код для метатегів, таких як заголовок сторінки, мета-опис та ключові слова.

Файл navmenu.php створює навігаційне меню, яке дозволяє користувачам навігуватися по сайту або взаємодіяти з основними розділами.

Файл scripts.php може містити JavaScript-код, який використовується для покращення функціональності веб-сторінки, такий як валідація форм або анімація.

Файл stylesheets.php може містити CSS-стили, які застосовуються до веб-сторінки, щоб задати її зовнішній вигляд і форматування.

Наприклад, розглянемо код файл header.php, що містить слайдер.

Нижче представлено код слайдеру.

```

<div class="simple-slider">
  <div class="swiper-container">
    <div class="swiper-wrapper">
      <div class="swiper-slide" style="background: url(<?php echo
URLROOT; ?>/assets/img/bgs/tyler-nix-yKalliZTaQU-unsplash.jpg) center center / cover no-
repeat;"></div>
      <div class="swiper-slide" style="background: url(<?php echo
URLROOT; ?>/assets/img/bgs/nathan-dumlao-71u2f0ofI-U-unsplash.jpg) center center / cover
no-repeat;"></div>
      <div class="swiper-slide" style="background: url(<?php echo
URLROOT; ?>/assets/img/bgs/jarek-ceborski-IhqDpFz7I8Q-unsplash.jpg) center center /
cover no-repeat;"></div>
    </div>
    <div data-bss-hover-animate="pulse" class="swiper-pagination"></div>
    <div data-bss-hover-animate="pulse" class="swiper-button-prev" style="color:
var(--bs-form-valid-color);"></div>
    <div data-bss-hover-animate="pulse" class="swiper-button-next" style="color:
var(--bs-form-valid-color);"></div>
  </div>
</div>

```

Цей код створює простий слайдер, який використовує бібліотеку Swiper для створення каруселі зображень. Кожен слайд має задній фон, який встановлюється за допомогою CSS inline стилів і використовує зображення, вказані в URLROOT, що дозволяє динамічно завантажувати зображення з визначеного місця. Також використовуються елементи управління Swiper, такі як пагінація та кнопки "Попередній" та "Наступний", для навігації між слайдами.

На рис. 1.16 представлено створені представлення MVC веб-застосунку.

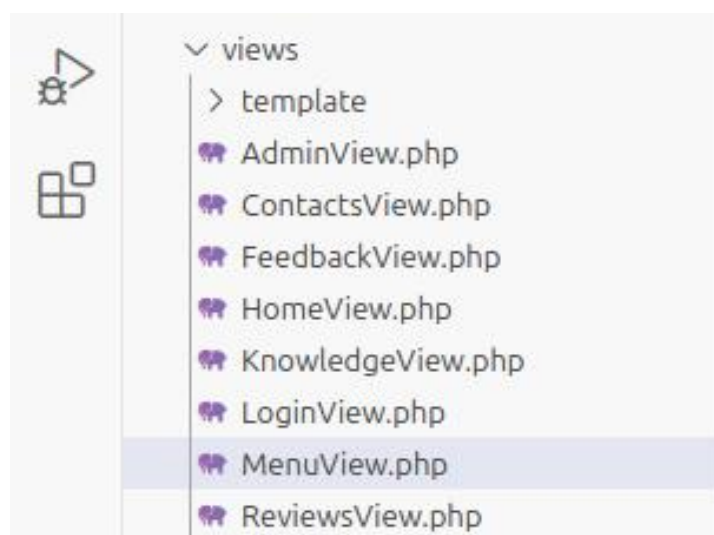


Рисунок 1.16. Представлення MVC веб-застосунку

Розглянемо типове представлення, наприклад для виводу меню.

Нижче представлено код представлення для виводу меню.

```
<!DOCTYPE html>
```

```

<html data-bs-theme="light" lang="en">

<head>
  <?php require_once APPROOT . '/views/template/meta.php'; ?>
  <?php require_once APPROOT . '/views/template/stylesheets.php'; ?>
</head>

<body class="bg-white">
  <?php require_once APPROOT . '/views/template/scripts.php'; ?>
  <?php require_once APPROOT . '/views/template/navmenu.php'; ?>
  <?php require_once APPROOT . '/views/template/header.php'; ?>

  <section class="p-5">
    <div class="container">
      <?php foreach ($data['itemsByCategory'] as $categoryName => $items) : ?>
        <h2 >>/h2>
        <h1 class="text-success text-center m-3"><?php echo
$categoryName; ?></h1>
        <div class="row row-cols-3 justify-content-center align-items-center">
          <?php if (!empty($items)) : ?>
            <?php foreach ($items as $item) : ?>
              <div class="col">
                <div class="card" data-bss-hover-animate="pulse">
                  
                  <div class="card-body">
                    <h1 class="card-title"><?php echo $item-
>name;?></h1>
                    <p class="card-text"><?php echo $item-
>description;?></p>
                    <p class="card-text"><?php echo $item-
>size;?></p>
                    <h4 class="card-title text-success"><?php echo
$item->price;?></h4>
                  </div>
                </div>
              </div>
            </div>
            <?php endforeach; ?>
          <?php else : ?>
            <?php require APPROOT . '/views/template/alert.php'; ?>
          <?php endif; ?>
        </div>
      <?php endforeach; ?>
    </div>
  </section>

  <?php require_once APPROOT . '/views/template/footer.php'; ?>
</body>

</html>

```

Даний HTML-код представляє сторінку, що відображається користувачам і містить категорії товарів з їх описами та зображеннями. У рядку 8-12 включаються файли метаданих, стилів та скриптів, які необхідні для правильної роботи сторінки. На рядку 16-30 маємо цикл, який проходиться по кожній категорії товарів і відображає їх назви та карточки товарів, включаючи їх

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		36

зображення, опис, розмір і ціну. Умова на рядку 20 перевіряє, чи існують товари у поточній категорії, в разі відсутності викликається шаблон "alert.php". Завершується сторінка файлом footer.php, який містить завершення тегу body та html.

1.3.5 Створення контролерів (Controllers)

Створимо класи-контролери, що зв'яжуть класи моделей (дані з БД) з файлами представлень (HTML-шаблонами графічного інтерфейсу користувача).

Контроллер являє собою компонент MVC архітектури, що відповідає за обробку вхідних даних, управління виконанням операцій та взаємодію з моделлю даних і представленням за допомогою роутингу.

На рис. 1.17 представлено створені контролери MVC веб-застосунку.

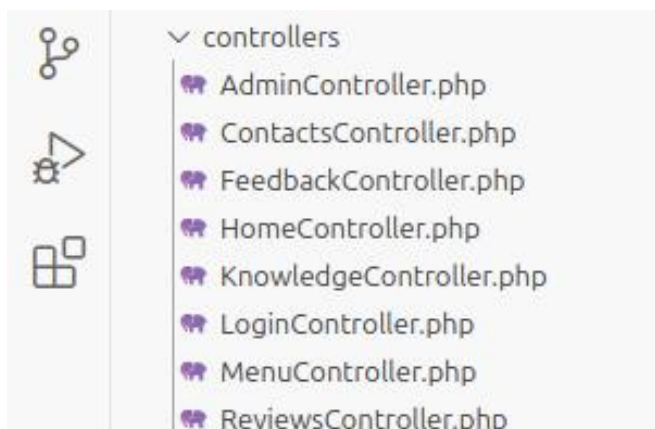


Рисунок 1.17. Контролери MVC веб-застосунку

Розглянемо типовий контролер, наприклад для виведення меню.

Нижче представлено код контролеру для виведення меню.

```
<?php
class MenuController extends Controller {
    private $informationModel;
    private $itemModel;
    private $categoryModel;

    public function __construct() {
        $this->informationModel = $this->model('Information');
        $this->itemModel = $this->model('Item');
        $this->categoryModel = $this->model('Category');
    }

    public function index()
    {
        $information = $this->informationModel->selectById(1);
```

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 06 001. 00 ДП ПЗ

Арк.

37

```

    $categories = $this->categoryModel->selectAll();
    $itemsByCategory = array();

    // Fetch items for each category
    foreach ($categories as $category)
    {
        $items = $this->itemModel->selectByCategory($category->id);
        $itemsByCategory[$category->name] = $items;
    }

    $data = [
        'information' => $information,
        'itemsByCategory' => $itemsByCategory
    ];

    $this->view('Menu', $data);

    flash('message');
}
?>

```

Цей контролер відповідає за керування відображенням меню на сторінці. При створенні екземпляра контролера він ініціалізує моделі для отримання інформації про інформацію, пункти меню та категорії. У методі `index()`, він здійснює вибірку інформації про заклад за ідентифікатором 1 та всіх доступних категорій. Потім, для кожної категорії, він отримує відповідні пункти меню та групує їх за категоріями. Нарешті, він передає отримані дані у вигляді асоціативного масиву в шаблон відображення "Menu". Після цього він викликає функцію `flash()`, яка має на меті відобразити користувачеві повідомлення.

1.3.6 Створення системи авторизації

У контексті архітектури MVC (Model-View-Controller) авторизація інтегрується таким чином, щоб контролювати доступ до контролерів і моделей, використовуючи сесії для збереження стану користувача.

Сесія являє собою механізм збереження даних на стороні сервера, який дозволяє веб-застосунку підтримувати стан між різними запитами від одного і того ж користувача. Коли користувач вперше відвідує сайт, сервер створює унікальний ідентифікатор сесії, який надсилається користувачу у вигляді cookie або додається до URL. Усі наступні запити від цього користувача будуть містити цей ідентифікатор, дозволяючи серверу отримувати доступ до відповідних даних сесії.

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
						38
Ізм.	Лист	№ докум.	Підпис	Дата		

Алгоритм роботи системи авторизації включає кілька кроків. Спочатку користувач надає свої облікові дані, такі як ім'я користувача та пароль, які контролер отримує і передає моделі для перевірки. Модель виконує запит до бази даних, щоб перевірити, чи існує користувач із такими обліковими даними і чи пароль правильний. Якщо дані коректні, модель повертає відповідь, яку контролер використовує для створення сесії. Сесія зберігає інформацію про авторизованого користувача, таку як ідентифікатор користувача або інші відповідні дані. При наступних запитах контролер перевіряє, чи існує активна сесія і чи містить вона інформацію про авторизацію. Якщо сесія активна і користувач авторизований, контролер надає доступ до запитуваного ресурсу або виконує відповідну дію. Якщо сесії немає або користувач не авторизований, контролер може перенаправити користувача на сторінку входу або видати повідомлення про відсутність доступу.

Розглянемо контролер авторизації та метод збереження сесії.

Нижче представлено код методу для збереження сесії.

```
public function createUserSession($user) {  
    // Save data to ssession  
    $_SESSION['id'] = $user->id;  
    $_SESSION['name'] = $user->name;  
    $_SESSION['login'] = $user->login;  
    $_SESSION['admin'] = $user->admin;  
}
```

Даний код представляє метод у класі, який створює сесію користувача. Він отримує об'єкт користувача як параметр і зберігає інформацію про нього у сесії. Кожен параметр, такий як ідентифікатор користувача, ім'я, логін та статус адміністратора, зберігається у глобальній змінній `$_SESSION`. Це дозволяє зберігати інформацію про авторизованого користувача на стороні сервера протягом сесії. Такий підхід забезпечує безпеку, оскільки дані сесії зберігаються на сервері і не відправляються користувачеві, а також забезпечує доступ до цих даних на протязі усієї сесії.

Розглянемо представлення авторизації, а саме форму авторизації.

Нижче представлено HTML-код розмітки форми авторизації.

```
<form method="POST">
```

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
						39
Ізм.	Лист	№ докум.	Підпис	Дата		

```

        <input class="form-control mt-3 mb-3" type="text" data-bss-
hover-animate="pulse" name="login" placeholder="Логін">
        <input class="form-control mt-3 mb-3" type="password" data-bss-
hover-animate="pulse" name="password" placeholder="Пароль">
        <button class="btn btn-outline-dark mt-3 mb-3 w-100" data-bss-
hover-animate="pulse" type="submit" formaction="<?php echo
URLROOT; ?>/login/signIn">Авторизуватись</button>
    </form>

```

Даний HTML-код представляє собою форму авторизації. У формі є два текстових поля для введення логіну та паролю, які визначаються через атрибути name. Після введення користувачем даних і натискання кнопки "Авторизуватись", дані форми надсилаються методом POST за допомогою HTTP-запиту на сервер. Атрибут formaction вказує URL, до якого буде відправлено цей запит, а сам URL генерується з константи URLROOT та шляху до контролера для авторизації. При відправці форми викликається відповідний контролер, який обробляє дані запиту і виконує процес авторизації користувача. Метод POST у протоколі HTTP передає дані в тілі запиту, що робить їх менш видимими для користувача та підвищує безпеку, що особливо важливо для передачі конфіденційної інформації, такої як пароль. На відміну від методу POST, метод GET передає дані через URL, що робить їх видимими в адресній стрічці браузера та призначеними для передачі нечутливих даних або отримання даних з сервера.

Розглянемо контролер системи авторизації та безпосередньо методу авторизації, що виконується під час натискання кнопки.

Нижче представлено код методу авторизації.

```

public function signIn() {
    // Check for POST
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        // Process form

        // Sanitize POST data
        $_POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_FULL_SPECIAL_CHARS);

        // Init data
        $data = [
            'login' => trim($_POST['login']),
            'password' => trim($_POST['password'])
        ];

        $loggedInUser = $this->userModel->select($data['login'], $data['password'],
1);

```

					РП 07. 06 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		40

```

// Validate user
if($loggedInUser) {
    $this->createUserSession($loggedInUser);
    redirect('admin');
}
else
{
    flash('message', 'Операція неуспішна! Можливо, не вірні дані. Спробуйте
інші дані.');
```

Цей код представляє собою метод для входу користувача в систему. Він перевіряє, чи надійшов запит методом POST, обробляє форму, очищує вхідні дані від потенційно небезпечних символів, витягує логін та пароль з POST-запиту, ініціалізує змінні для зберігання цих даних. Потім він викликає метод моделі для перевірки логіну та паролю в базі даних. Якщо користувача знайдено, він створює сеанс користувача і перенаправляє його на сторінку адміністратора. У протилежному випадку, він відображає повідомлення про невдалий вхід і перенаправляє користувача на сторінку входу.

1.4 Тестування створеного веб-застосунку

Виконаємо мануальне тестування створеного веб-застосунку, протестувавши основну функціональність основних веб-сторінок реалізованого веб-застосунку.

Мануальне тестування являє собою процес перевірки програмного забезпечення вручну для виявлення дефектів без використання автоматичних інструментів.

Мануальне тестування є важливим етапом життєвого циклу майже будь-якого програмного продукту через те, що воно дозволяє виявляти помилки, які можуть бути пропущені автоматичними тестами, крім того, воно дозволяє виявити помилки зі сторони користувацького досвіду.

Нижче представлено підрозділи з мануальним тестуванням створеного веб-застосунку у вигляді функціонального огляду кожної веб-сторінки.

1.4.1 Функціональний огляд головної веб-сторінки

Розглянемо головну веб-сторінку веб-застосунку.

Головна веб-сторінка є обличчям веб-застосунку і надає користувачеві наступні основні функціональні можливості: навігація та отримання інформації про екстрені повідомлення.

На рис. 1.18 представлено головну веб-сторінку веб-застосунку.

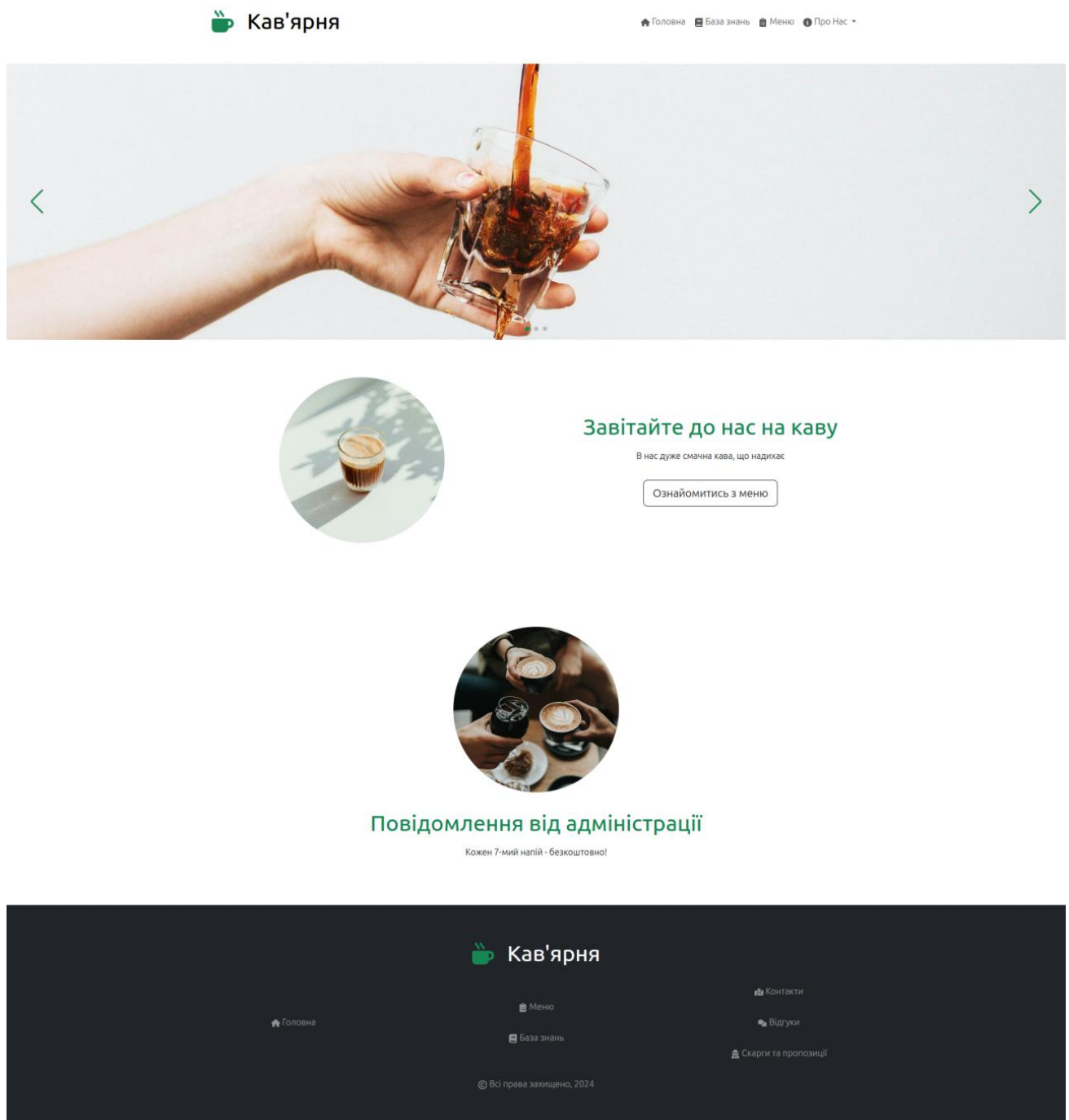


Рисунок 1.18. Головна сторінка веб-застосунку

					РП 07. 06 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		42

1.4.2 Функціональний огляд інформаційних веб-сторінок

Розглянемо наступну веб-сторінку веб-застосунку – “База знань”, “Меню” та “Контакти”.

Веб-сторінка “База знань” надає користувачеві наступну основну функціональну можливість: отримання теоретичної інформації про кавові напої з вільної енциклопедії “Wikipedia”.

На рис. 1.19 представлено веб-сторінку веб-застосунку – “База знань”.

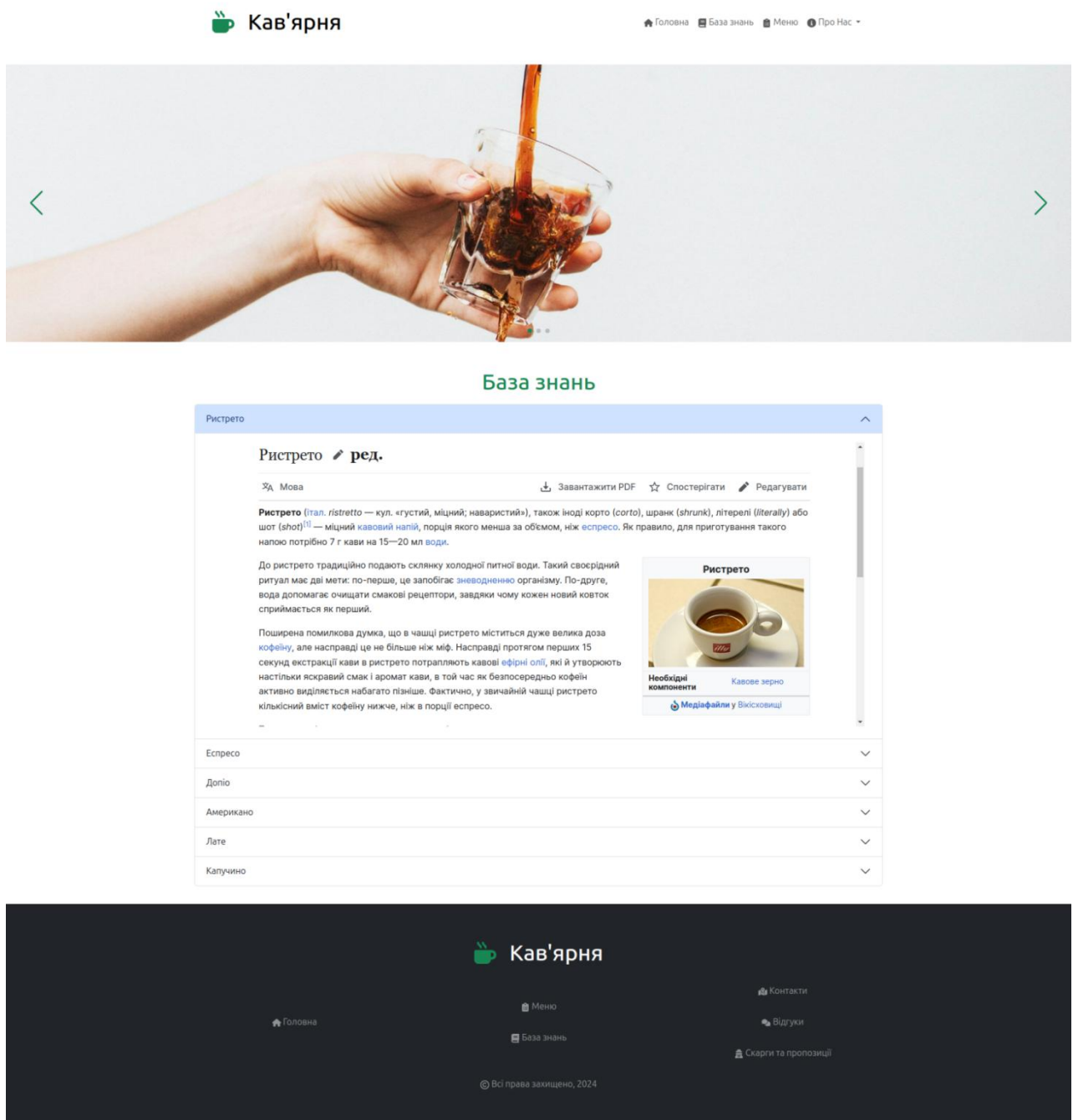


Рисунок 1.19. Веб-сторінка веб-застосунку – “База знань”

					РП 07. 06 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		43

Веб-сторінка “Меню” надає користувачеві наступну основну функціональну можливість: ознайомлення з асортиментом кав’ярні по категоріям.

На рис. 1.20 представлено веб-сторінку веб-застосунку – “Меню”.

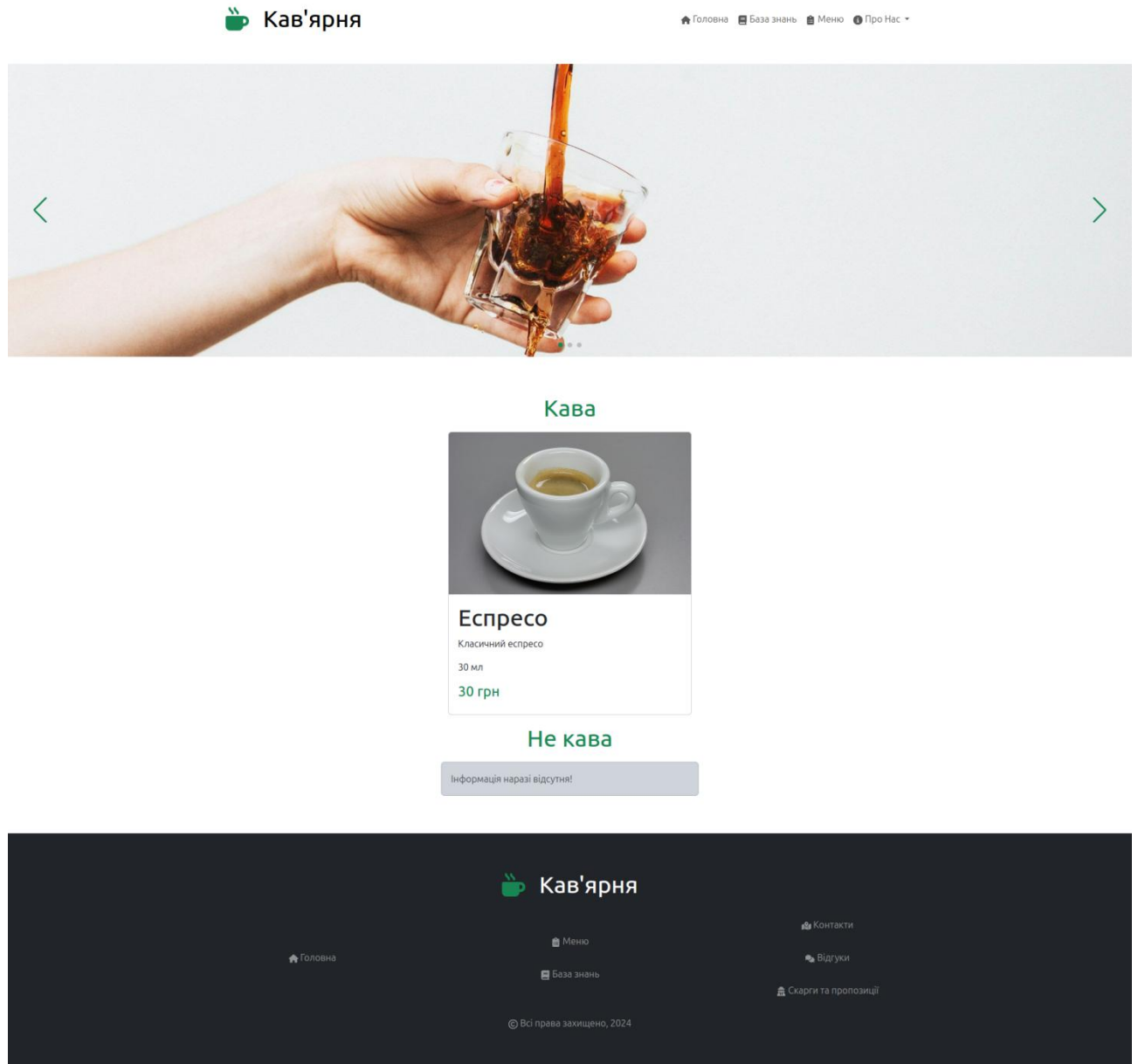





Рисунок 1.20. Веб-сторінка веб-застосунку – “Меню”

Веб-сторінка “Контакти” надає користувачеві наступну основну функціональну можливість: отримання інформації про фізичну адресу, номер телефону та електронну пошту кав’ярні.

На рис. 1.21 представлено веб-сторінку веб-застосунку – “Контакти”.

					<i>РП 07. 06 001. 00 ДП ПЗ</i>	Арк.
						44
Ізм.	Лист	№ докум.	Підпис	Дата		



 Адреса м. Одеса, вул. Приморська 1 <input type="button" value="Відкрити карту"/>	 Телефон +3800000000000 <input type="button" value="Подзвонити"/>	 E-Mail name@domain.com <input type="button" value="Написати лист"/>
--	--	---

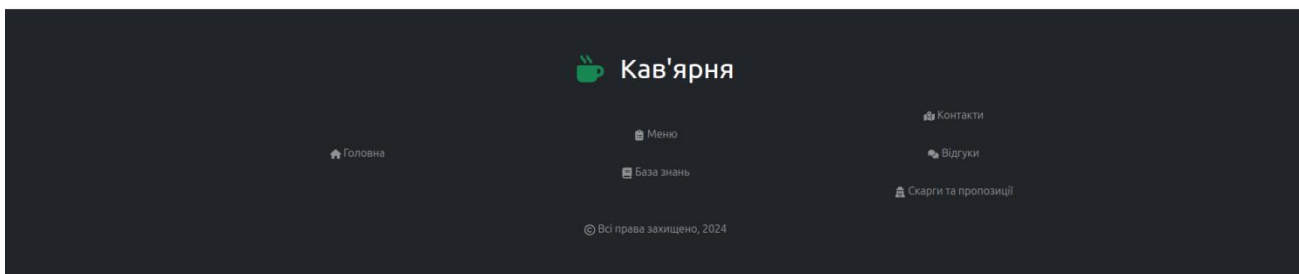


Рисунок 1.21. Веб-сторінка веб-застосунку – “Контакти”

1.4.3 Функціональний огляд веб-сторінок зворотнього зв’язку

Розглянемо наступні веб-сторінки веб-застосунку – “Відгуки” та “Зворотній зв’язок”.

Веб-сторінка “Відгуки” надає користувачеві наступну основну функціональну можливість: можливість залишити відгук про кав’ярню та оцінювати вже існуючі відгуки засобами вподобайок. Користувачі можуть детально описати свій досвід, поділившись враженнями про якість обслуговування, атмосферу та смак напоїв. Система вподобайок дозволяє іншим відвідувачам бачити, які відгуки вважаються корисними чи актуальними. Це допомагає створити більш об’єктивну картину про кав’ярню та покращує взаємодію між користувачами, сприяючи активній участі в обговореннях.

На рис. 1.22 представлено веб-сторінку веб-застосунку – “Відгуки”.



Залишити відгук

Ім'я

E-Mail

Відгук

Дивитися відгуки

Тест

Тест

2024-04-03 18:02:42

♥ 1

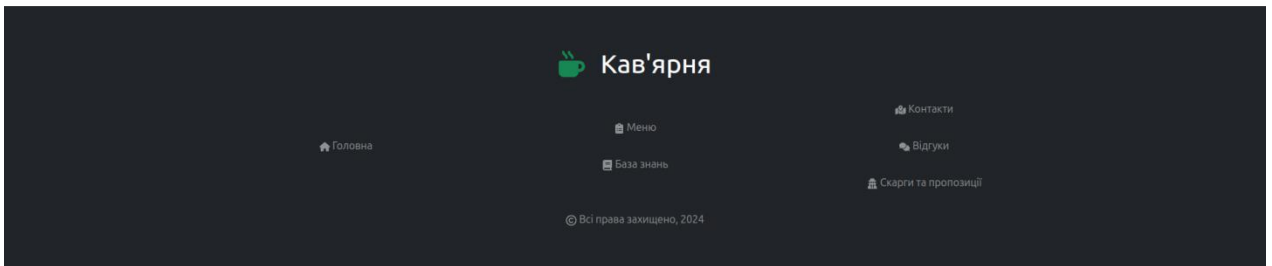


Рисунок 1.22 Веб-сторінка веб-застосунку – “Меню”

Веб-сторінка “Скарги та пропозиції” надає користувачеві наступну основну функціональну можливість: можливість залишити анонімну скаргу про сервіс кав’ярні або пропозицію на тему співпраці або покращення сервісу. Користувачі можуть вільно висловлювати свої зауваження щодо якості обслуговування або будь-яких інших аспектів роботи кав’ярні, не розкриваючи своєї особистості. Це забезпечує відвертість і чесність у зворотному зв’язку, що допомагає керівництву кав’ярні швидше ідентифікувати та вирішувати проблеми. Крім того, сторінка надає платформу для висловлення конструктивних пропозицій, які можуть сприяти покращенню сервісу та розвитку нових ідей для співпраці. Веб-сторінка стимулює активну участь користувачів у покращенні якості обслуговування та підвищенні загального рівня задоволеності клієнтів.

Ізм.	Лист	№ докум.	Підпис	Дата

На рис. 1.23 представлено веб-сторінку веб-застосунку – “Скарги та пропозиції”.

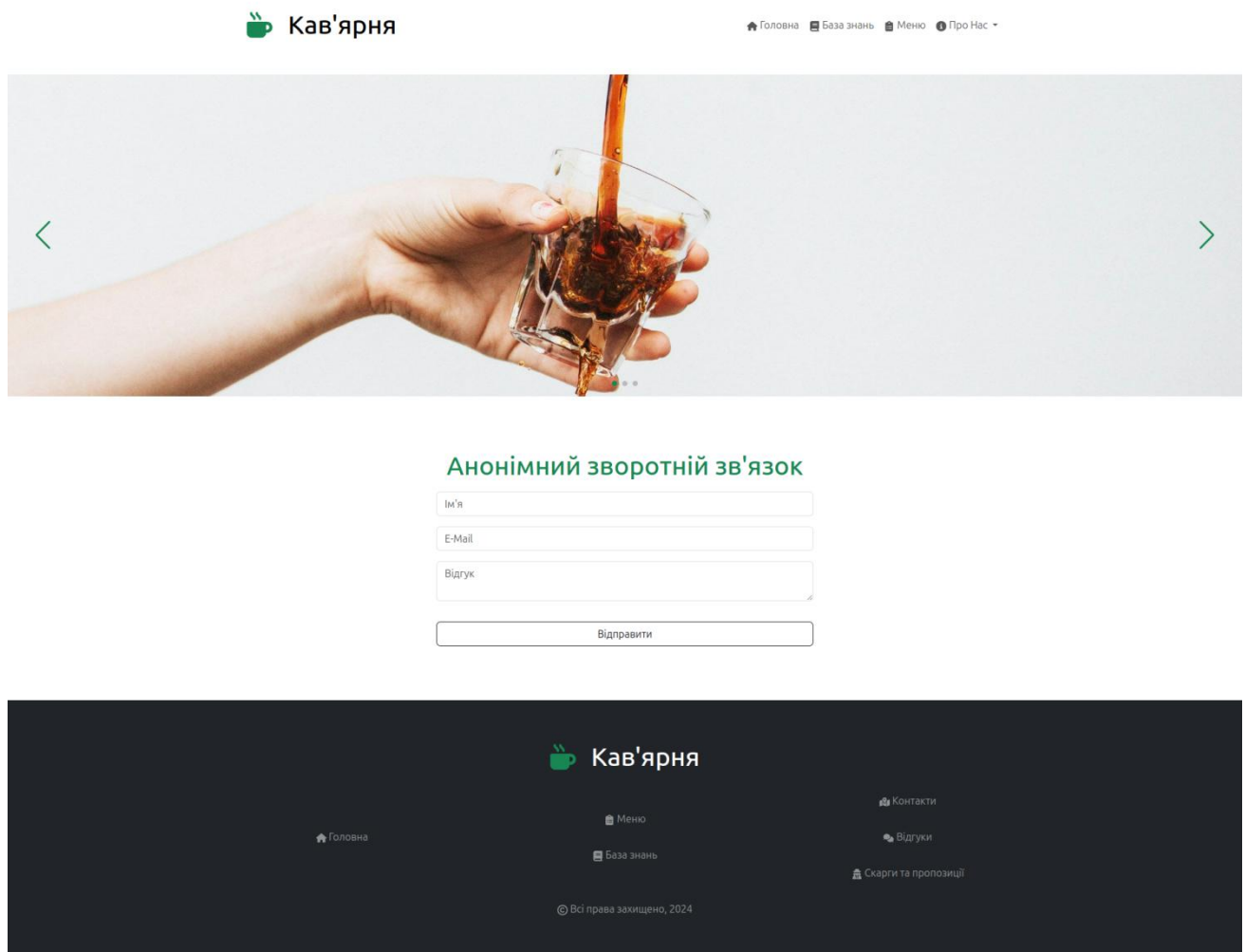


Рисунок 1.23. Веб-сторінка веб-застосунку – “Скарги та пропозиції”

1.4.4 Функціональний огляд системи адміністрування

Розглянемо наступні веб-сторінки веб-застосунку – “Авторизація” та “Менеджемент веб-системи”.

Веб-сторінка “Авторизація” надає користувачеві наступну основну функціональну можливість: авторизація адміністратора у веб-систему. Після успішної авторизації адміністратор отримує доступ до спеціальних функцій та може керувати адміністративною частиною веб-застосунку. Така система авторизації забезпечує безпеку даних, оскільки заснована на використанні сесій,

					РП 07. 06 001. 00 ДП ПЗ	Арк.
						47
Ізм.	Лист	№ докум.	Підпис	Дата		

що дозволяє відслідковувати та контролювати доступ до конкретних ресурсів у веб-застосунку.

На рис. 1.24 представлено веб-сторінку веб-застосунку – “Авторизація”.

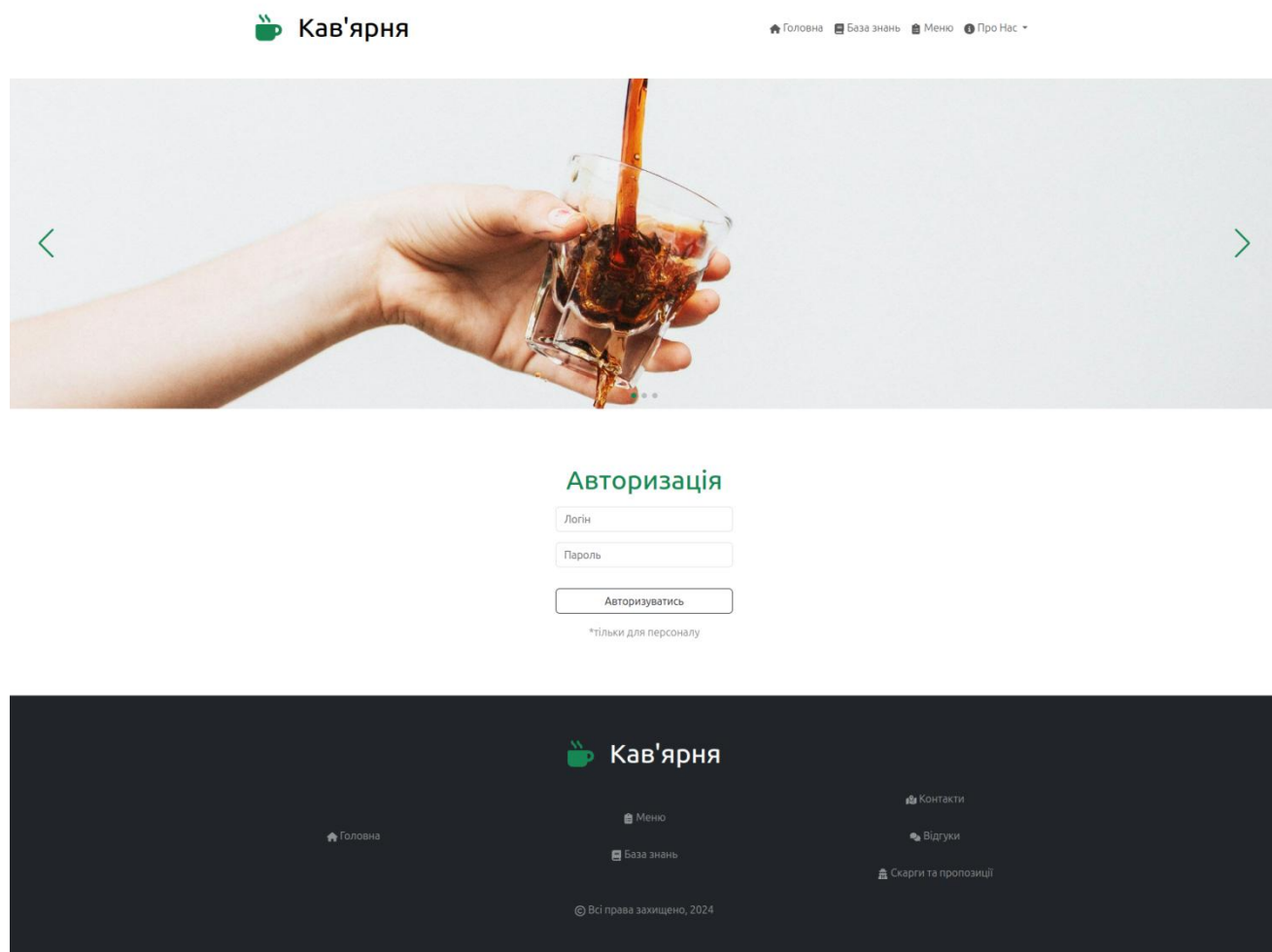


Рисунок 1.24. Веб-сторінка веб-застосунку – “Авторизація”

Веб-сторінка “Менеджмент веб-системи” дозволяє користувачеві здійснювати широкий спектр операцій з основною інформацією системи. Крім перегляду, створення, редагування та видалення інформації, користувачі можуть впорядковувати її за розділами, що сприяє зручності в навігації та організації даних. Такий функціонал робить процес управління веб-системою більш ефективним та продуктивним для користувачів усіх рівнів навичок.

На рис. 1.25 представлено веб-сторінку веб-застосунку – “Менеджмент веб-системи”.

					РП 07. 06 001. 00 ДП ПЗ	Арк.
						48
Ізм.	Лист	№ докум.	Підпис	Дата		



Налаштування

Привіт, Admin

Вийти з панелі адміністратор

Статистика

Загальна кількість відвідувачів: 1

Менеджмент

Назва	Категорія	Фото	Опис	Розмір	Ціна	Активний	Дії
<input type="text"/>	Кава	Choose File No...en	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	Додати
Еспресо	Кава	Choose File No...en	Класичний еспр	30 мл	30 грн	<input checked="" type="checkbox"/>	Редагувати Видалити

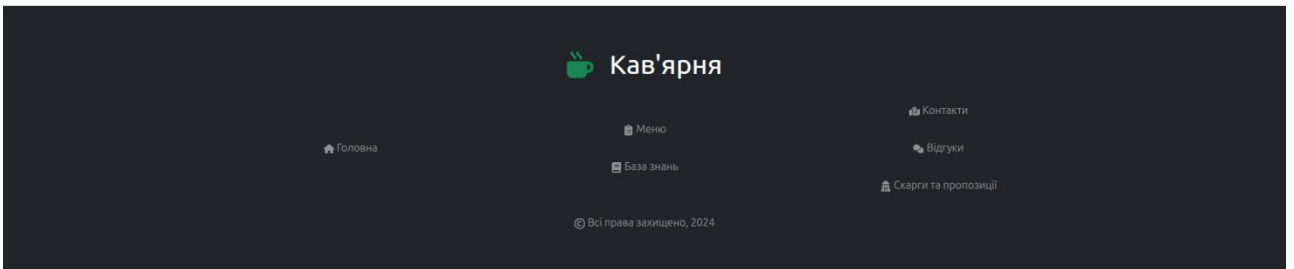


Рисунок 1.25. Веб-сторінка веб-застосунку – “Менджмент веб-системи”

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 06 001. 00 ДП ПЗ

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

В даному дипломному проєкті розроблено веб-орієнтовану систему для управління кав'ярним бізнесом, яка базується на передових технологіях в галузі програмування та баз даних та відповідає вимогам та потребам сучасного ринку. Проведений аналіз показав, що використання такої системи є критично важливим для оптимізації процесів у кав'ярнях, забезпечуючи їм конкурентоспроможність та ефективне управління. Застосування веб-систем дозволяє кав'ярням швидше реагувати на зміни в попиті, оптимізувати робочі процеси та забезпечувати високий рівень обслуговування для клієнтів.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення. Оцінка якості програмного продукту включає визначення трудомісткості та вартості його створення.

2.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_0 , усл. машинних командах
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП введення інформації	1060 – 5750

У табл. 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл. 2.2.

Таблиця 2.2. Обсяг ПП

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
5.00	306
6.00	330
7.00	357
8.00	385
9.00	414
10.00	445

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k = 0,7 \div 0,8$): $T_{ар} = 306 \times 0,8 = 244,8$ (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{ТЗ} = T^a p \leftarrow \mathcal{L}_1 \leftarrow \mathcal{K}_H \quad (2.1)$$

$$T_{ПП} = T^a p \leftarrow \mathcal{L}_2 \leftarrow \mathcal{K}_H \quad (2.2)$$

$$T_{РП} = T^a p \leftarrow \mathcal{L}_3 \leftarrow \mathcal{K}_H \leftarrow \mathcal{K}_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.3);

K_n – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.4);

K_t – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.5.).

Таблиця 2.3. Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_n
А	Принципово нові ПО	1,75 – 1,2
Б	ПО – розвиток визначеног параметричного ряду	1,0 – 0,8
В	ПО маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПО типовими програмами, %	Значення K_t
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ} = T_a * L_1 * K_n = 244,8 * 0,12 * 0,8 = 23,50 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість розробки технічного проєкту

$$T_{ТП} = T_a * L_2 * K_n = 244,8 * 0,15 * 0,8 = 29,37 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки робочого проєкту

$$T_{РП} = T_a * L_3 * K_n * K_r = 244,8 * 0,58 * 0,8 * 0,7 = 79,51 \text{ (люд/годин)} \quad (2.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{ТЗ}=2$ (стор), розробка ТП $N_{ТП}=43$ (стор), розробка робочого проєкту $N_{РП}=9$ (стор), пояснювальна записка відповідно $N_{ПЗ}=15$ (стор). Розрахунок зведений у таблицю 2.6.

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин		
1.ТЗ	$T_{РТЗ}=23,50$	$T_{КК}=0,7*N_{ТЗ}= 0,7*2=1,4$	$T_{НК}=0,15*N_{ТЗ}=0,15*2=0,30$
2.Розробка ТП	$T_{РТП}=29,37$	$T_{КК}=0,7*N_{ТП}=0,7*43=30,1$	$T_{НК}=0,15*N_{ТП}=0,15*43=6,45$
3.Розробка РП	$T_{РРП}= 79,51$	$T_{КК}=0,7*N_{РП}=0,7*9=6,3$	$T_{НК}=0,15*N_{РП}=0,15*9=1,35$
4.Розробка ПЗ	$T_{ПЗ}=1,5*N_{ПЗ}=1,5*15=22,5$	$T_{КК}=0,7*N_{ТЗ}=0,7*15=10,5$	$T_{НК}=0,15*N_{ПЗ}=0,15*15=2,25$
Усього, в т.ч.:	213,53		
- на розробку	$T_p=154,88$		
- контроль керівника		$T_{КК}=48,3$	
- нормоконтроль			$T_{НК}=10,35$

2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, загальні витрати на розробку ПП. Розрахунок основної заробітної плати виконавців приведений у табл. 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2024» встановлено мінімальну заробітну плату у місячному розмірі з 1 квітня 2024 року - 8000 гривень; мінімальну погодинну тарифну ставку – 46,00 грн.

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	154,88	55,35	8573,48
2.Контроль керівника	48,3	120,28	5810,00
3.Нормоконтроль	10,35	120,28	1245,00
Усього	-	-	$30= 15628,48$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.8.

Таблиця 2.8. Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	70	4.0	280,0
Разом	-	-	-	$V_{mi}=280,0$
Транспортно – заготівельні витрати (10%)				$V_{тр_з} = 0,1 \cdot V_{m1} = 0,1 \cdot 280 = 28,00$
Усього				$V_m = V_{mi} + V_{тр_з} = 308,00$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.9.

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	308,00	V_m (див. табл. 2.8)
2. Основна заробітна плата	15628,48	Z_o (див. табл. 2.7)
3. Додаткова заробітна плата	1562,84	$Z_d = 0,1 \leftarrow Z_o = 15628,48 * 0,1$
4. Відрахування до єдиного фонду соціального внеску	3864,37	$V_{e.c.v.} = 0,22 \leftarrow (Z_o + Z_d) = 0,22 * (15628,48 + 1562,84)$
5. Накладні витрати	6251,39	$V_{нак.} = 0,4 \leftarrow Z_o = 0,4 * 15628,48$
6. Повна собівартість	27681,08	$C_{пов} = V_m + Z_o + Z_d + V_{e.c.v.} + V_{нак.} = 308,00 + 15628,48 + 1562,84 + 3864,37 + 6251,39$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$P = (C_p * R) / 100 = (27681,08 * 10) / 100 = 2768,11 \text{ грн.} \quad (2.4)$$

Де p – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$C_o = C_p + P = 27681,08 + 2768,11 = 30449,19 \text{ грн.} \quad (2.5)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$C_p = C_o + ПДВ = 30449,19 + 30449,19 * 0.2 = 36539,03 \text{ грн.} \quad (2.6)$$

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Вступ

Безпечні умови праці – не тільки запорука комфортного існування працівників у межах підприємства, а в першу чергу – їх здоров'я та працездатності, а відтак і прибутковості підприємства. Безпека праці на підприємстві може бути на належному рівні тільки тоді, коли всебічно виконуються вимоги трудового законодавства, державних стандартів України, норм і правил, розроблених для збереження здоров'я працюючих.

В розділі охорона праці дипломного проекту розглядаються питання умов праці програміста (оператора ПК), які повинні бути забезпечені на підприємстві для безпечної роботи працівника при розробки «Веб-орієнтованої системи для ведення кав'ярного бізнесу»

3.2 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці програмного комплексу

Небезпечним називається фактор, вплив якого на працюючу людину в певних умовах може привести до виробничої травми або іншому раптовому різкому погіршенню здоров'я. Якщо ж виробничий чинник приведе до захворювання або зниження працездатності, то його вважають шкідливим. Залежно від рівня й тривалості впливу, шкідливий чинник може стати небезпечним.

В процесі роботи на користувачів ПК можуть мати вплив наступні небезпечні та шкідливі фактори:

- Невідповідність параметрів мікроклімату нормам;
- Недостатній рівень освітленості;
- Ураження електрострумом;
- Статична електрика;
- Порушення організації робочого місця тощо.

3.3 Гігієнічні вимоги до виробничого середовища

У відповідності з Правилами охорони праці під час експлуатації ЕОМ на робочому місці користувача ПК повинні бути створенні умови для високопродуктивної праці. Розглянемо ці умови.

3.3.1 Вимоги до приміщення

Для приміщень, які призначені для роботи з ВДТ, доцільно обрати орієнтацію вікон на північ або на північний схід. На вікнах повинні бути жалюзі, що регулюються, або штори, що дають можливість їх повністю закривати. Приміщення відповідно до ДБН В.2.5-28-2018 «Природне і штучне освітлення» повинні мати природне та штучне освітлення. З приміщеннях ВДТ мають бути обладнані побутові приміщення для відпочинку, психологічного розвантаження тощо.

Площа на одне робоче місце для користувачів повинна складати не менше 6 кв.м, а об'єм – не менше 20,0 куб.м. Стіни пофарбовані матовою фарбою, у відповідності з санітарними вимогами.

3.3.2 Освітлення

Для освітлення приміщення, у якому працює користувач ПК, використовується змішане освітлення, тобто сполучення природного й штучного освітлення. Для загального освітлення приміщення використовуються газорозрядні лампи типу ЛД. Норма для необхідної освітленості робочого місця становить 300-500 лк.

3.3.3 Шум

При розумовій праці, яка вимагає зосередженості припустимий рівень шуму становить 50дБ. Для зменшення шуму й вібрації в приміщенні устаткування, апарати й прилади встановлюють на спеціальні прокладки, що амортизують. Якщо стіни в приміщенні є джерелами шумоутворення, вони повинні бути облицьовані звуковбирним матеріалом.

					РП 07. 06 003. 00 ДП ПЗ	Арк.
						57
Ізм.	Лист	№ докум.	Підпис	Дата		

3.3.4 Мікроклімат

Порушення відповідності ц параметрів мікроклімату впливають на працездатність працівників, їх реакцій, збільшення кількості помилок. Тому в приміщенні повинні бути установлені оптимальні параметри мікроклімату: температура повітря 22-25 0С, вологість повітря – 40-60%, швидкість пуху повітря – 0,1-0,2 м/с. Для цього приміщення має бути оснащено системами опалення й кондиціонування, що забезпечують постійне й рівномірне нагрівання, циркуляцію й очищення повітря від пилу й шкідливих речовин.

3.3.5 Електробезпека

Проходячи через організм людини електричний струм робить термічну, електролітичну і біологічну дію.

Для попередження поразок електричним струмом необхідно:

У повному обсязі виконувати правила провадження робіт і правил технічної експлуатації;

Виключати можливість доступу працівника до частин устаткування, що працює під небезпечною напругою, неізольованим частинам, призначеним для роботи при малій напрузі й не підключеним до захисного заземлення;

Застосовувати ізоляцію, що служить для захисту від поразки електричним струмом.

Для попередження поразок електричним струмом необхідно:

У повному обсязі виконувати правила провадження робіт і правил технічної експлуатації;

Виключати можливість доступу працівника до частин устаткування, що працює під небезпечною напругою, неізольованим частинам, призначеним для роботи при малій напрузі й не підключеним до захисного заземлення;

Застосовувати ізоляцію, що служить для захисту від поразки електричним струмом.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із

заземленим відкритим екраном) мають бути надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння працівника під напругу.

3.3.6 Вимоги до організації робочого місця працівника

Робочі місця повинні бути розташовані так, щоб у поле зору працюючого не попадали поверхні, що мають властивість віддзеркалювання, вікна освітлювальні прилади. Відеотермінали повинні встановлюватися під кутом 90-100 градусів від вікон, так, щоб світло падало з боку. Робочі місця з ВДТ доцільно розміщати в глибині приміщення. Розташування відео терміналу, при якому працюючий звернений обличчям або спиною до вікон, неприпустимо при будь-якому способі реалізації загального висвітлення, як прямим, так і відбитим світлом.

Робочий стіл повинен регулюватися по висоті в границях 680-800 мм, а ширина – забезпечувати можливість виконання операцій в зоні досяжності моторного поля. Рекомендовані розміри столу: висота 725 мм, ширина 600-1400 мм, глибина 800-1000 мм. Робочий стілець повинен бути оснащений підйомно-поворотним пристроєм для регулювання висоти сидіння і спинки, а також кута її нахилу. Регулювання кожного параметра повинне вироблятися легко, бути незалежним і надійно фіксуватися.

Розташування екрана ВДТ має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30 до нормальної лінії погляду працюючого.

Клавіатуру слід розташовувати на поверхні столу на відстані 100...300 мм від краю, звернутого до працюючого.

Організація робочого місця користувача комп'ютера повинна забезпечувати відповідність усіх елементів робочого місця та їх взаємного розташування ергономічним вимогам.

На рис. 3.1 представлено робоче місце і робоча поза користувача комп'ютера.

					<i>РП 07. 06 003. 00 ДП ПЗ</i>	Арк.
						59
Ізм.	Лист	№ докум.	Підпис	Дата		

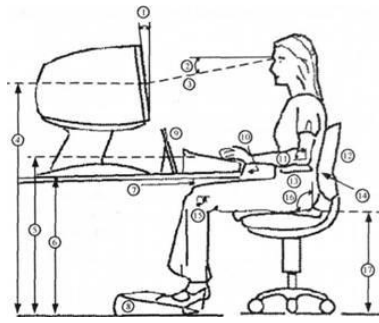


Рисунок 3.1. Робоче місце і робоча поза користувача комп'ютера

Робоче місце і робоча поза користувача комп'ютера включає:

1 — кут екрана; 2 — кут огляду (зору); 3 — відстань огляду; 4 — висота середини екрана; 5 — висота клавіатури; 6 — висота столу; 7 — відстань колін від столу; 8 — підставка для ніг; 9 — підставка для документів; 10 — положення рук; 11 — кут ліктів; 12 — спинка крісла; 13 — підлокітник; 14 — опора для попереку; 15 — кут колін; 16 — кут спинки крісла; 17 — висота сидіння

3.4 Пожежна безпека

Пожежна безпека приміщень, що мають електричні мережі, регламентується ГОСТ 12.1.033-81, ГОСТ 12.1.004-85. Робота оператора ЕОМ повинна вестися в приміщенні, що відповідає категорії Д пожежної безпеки (негорючі речовини й матеріали в холодному стані).

Пожежна безпека забезпечується:

- системою запобігання пожежі;
- системою протипожежного захисту;
- організаційно-технічними заходами.

Протипожежний захист приміщення забезпечується застосуванням установки автоматичної пожежної сигналізації, наявністю засобів пожежогасіння, організацією своєчасної евакуації людей.

Для ліквідації невеликих осередків пожеж, а також для гасіння пожеж у початковій стадії їх розвитку силами персоналу об'єктів, застосовуються первинні засоби пожежогасіння. Це вогнегасники (вуглекислотні та порошкові), пожежний інвентар (покривала з негорючого полотна, ящики з піском, бочки з водою), пожежний інвентар.

На рис. 3.2 представлено первинні засоби пожежогасіння.



Рисунок 3.2. Первинні засоби пожежогасіння

ВИСНОВКИ

У дипломному проєкті було розроблено веб-орієнтовану систему для управління кав'ярним бізнесом, яка базується на передових технологіях в галузі програмування та баз даних та відповідає вимогам та потребам сучасного ринку. Проведений аналіз показав, що використання такої системи є критично важливим для оптимізації процесів у кав'ярнях, забезпечуючи їм конкурентоспроможність та ефективне управління. Застосування веб-систем дозволяє кав'ярням швидше реагувати на зміни в попиті, оптимізувати робочі процеси та забезпечувати високий рівень обслуговування для клієнтів.

Для розробки програмного продукту було задіяно різноманітний інструментарій, включаючи технологію розробки реляційних баз даних MySQL та об'єктно-орієнтоване програмування з використанням мов PHP та JavaScript. Цей інструментарій надав можливість ефективно створити програмний продукт, який відповідає високим стандартам якості та продуктивності.

У створеному програмному продукті реалізовано широкий функціонал, який включає у себе системи управління меню, зворотнім зв'язком та інші важливі аспекти управління кав'ярним бізнесом.

В пояснювальній записці розглянуті всі питання передбачені технічним завданням на дипломне проєктування, проведено аналіз предметної області; детально описано технології та засоби, які використовувалися при створенні проєкту; проведено розрахунок економічної ефективності створення та впровадження створеного проєкту; розглянуті питання охорони праці та наведений перелік використаних джерел.

					<i>РП 07. 06 000. 00 ДП ПЗ</i>	Арк.
						62
Ізм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Aroma Kava. [Веб-сайт]. URL: <https://aromakava.ua/en>.
2. CoffeeOcean. [Веб-сайт]. URL: <https://coffeeocean.od.ua/>.
3. АЛЬТЕРНАТИВА. [Веб-сайт]. URL: <https://alt.choiceqr.com/>.
4. HTML Docs. [Веб-сайт]. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
5. CSS Docs. [Веб-сайт]. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
6. Bootstrap Docs. [Веб-сайт]. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>.
7. JavaScript Docs. [Веб-сайт]. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/>.
8. jQuery Docs. [Веб-сайт]. URL: <https://api.jquery.com/>.
9. PHP Docs. [Веб-сайт]. URL: <https://www.php.net/docs.php/>.
10. SQL Tutorial. [Веб-сайт]. URL: https://www.w3schools.com/sql/sql_intro.asp.
11. MySQL Docs. [Веб-сайт]. URL: <https://dev.mysql.com/doc/>.
12. К. В. Двірничук, Д. О. Вацек. Веб-програмування та веб-дизайн: Навчальний посібник. – «Чернівецький національний університет імені Юрія Федьковича», 2022.
13. Дворецький, І. М. Журавська, С. В. Дворецька, С. Ю. Боровльова. Розробка гібридних застосунків для цифрової трансформації бізнесу: Навчальний посібник. – «ЧНУ ім. Петра Могили», 2022.
14. statista. [Веб-сайт]. URL: <https://www.statista.com/>.

					РП 07. 06 000. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		63

ДОДАТОК А. Програмний код основної логіки веб-застосунку

```
// ItemModel.php

<?php
class ItemModel
{
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function selectAll()
    {
        // Query
        $this->db->query('SELECT * FROM items');

        // Execute
        $results = $this->db->resultSet();

        // Return result
        return $results;
    }

    public function selectByCategory($category)
    {
        // Query
        $this->db->query('SELECT * FROM items WHERE category = :category');

        // Bind value
        $this->db->bind(':category', $category);

        // Execute
        $results = $this->db->resultSet();

        // Return result
        return $results;
    }

    public function selectByPage($page, $perPage)
    {
        // Calculate the offset based on the current page and items per page
        $offset = ($page - 1) * $perPage;

        // Query with pagination
        $this->db->query('SELECT * FROM items LIMIT :perPage OFFSET :offset');

        // Bind values
        $this->db->bind(':perPage', $perPage);
        $this->db->bind(':offset', $offset);

        // Execute
        $results = $this->db->resultSet();

        // Return result
    }
}
```

```

    return $results;
}

public function insert($data)
{
    // Query
    $this->db->query('INSERT INTO items (category, photo, name, description, size,
price, active) VALUES(:category, :photo, :name, :description, :size, :price, :active)');

    // Bind values
    $this->db->bind(':category', $data['category']);
    $this->db->bind(':photo', $data['photo']);
    $this->db->bind(':name', $data['name']);
    $this->db->bind(':description', $data['description']);
    $this->db->bind(':size', $data['size']);
    $this->db->bind(':price', $data['price']);
    $this->db->bind(':active', $data['active']);

    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}

public function update($data)
{
    // Query
    $this->db->query('UPDATE items SET category = :category, photo = :photo, name
= :name, description = :description, size = :size, price = :price, active = :active
WHERE id = :id');

    // Bind values
    $this->db->bind(':id', $data['id']);
    $this->db->bind(':category', $data['category']);
    $this->db->bind(':photo', $data['photo']);
    $this->db->bind(':name', $data['name']);
    $this->db->bind(':description', $data['description']);
    $this->db->bind(':size', $data['size']);
    $this->db->bind(':price', $data['price']);
    $this->db->bind(':active', $data['active']);

    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}

public function delete($data)
{
    // Query
    $this->db->query('DELETE FROM items WHERE id = :id');
}

```

```

        // Bind values
        $this->db->bind(':id', $data['id']);

        // Execute
        if($this->db->execute())
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
?>

// MenuView.php

<!DOCTYPE html>
<html data-bs-theme="light" lang="en">

<head>
    <?php require_once APPROOT . '/views/template/meta.php'; ?>
    <?php require_once APPROOT . '/views/template/stylesheets.php'; ?>
</head>

<body class="bg-white">
    <?php require_once APPROOT . '/views/template/scripts.php'; ?>
    <?php require_once APPROOT . '/views/template/navmenu.php'; ?>
    <?php require_once APPROOT . '/views/template/header.php'; ?>

    <section class="p-5">
        <div class="container">
            <?php foreach ($data['itemsByCategory'] as $categoryName => $items) : ?>
                <h2 ></h2>
                <h1 class="text-success text-center m-3"><?php echo $categoryName; ?></h1>
                <div class="row row-cols-3 justify-content-center align-items-center">
                    <?php if (!empty($items)) : ?>
                        <?php foreach ($items as $item) : ?>
                            <div class="col">
                                <div class="card" data-bss-hover-animate="pulse">
                                    
                                    <div class="card-body">
                                        <h1 class="card-title"><?php echo $item-
>name;?></h1>
                                        <p class="card-text"><?php echo $item-
>description;?></p>
                                        <p class="card-text"><?php echo $item->size;?></p>
                                        <h4 class="card-title text-success"><?php echo
$item->price;?></h4>
                                    </div>
                                </div>
                            </div>
                        </div>
                    <?php endforeach; ?>
                <?php else : ?>
                    <?php require APPROOT . '/views/template/alert.php'; ?>
                <?php endif; ?>
            </div>
        <?php endforeach; ?>
    </div>
</section>

```

```
<?php require_once APPROOT . '/views/template/footer.php'; ?>
</body>
```

```
</html>
```

```
// MenuController.php
```

```
<?php
class MenuController extends Controller {
    private $informationModel;
    private $itemModel;
    private $categoryModel;

    public function __construct() {
        $this->informationModel = $this->model('Information');
        $this->itemModel = $this->model('Item');
        $this->categoryModel = $this->model('Category');
    }

    public function index()
    {
        $information = $this->informationModel->selectById(1);
        $categories = $this->categoryModel->selectAll();
        $itemsByCategory = array();

        // Fetch items for each category
        foreach ($categories as $category)
        {
            $items = $this->itemModel->selectByCategory($category->id);
            $itemsByCategory[$category->name] = $items;
        }

        $data = [
            'information' => $information,
            'itemsByCategory' => $itemsByCategory
        ];

        $this->view('Menu', $data);

        flash('message');
    }
}
?>
```

```
// AdminController.php
```

```
<?php
class AdminController extends Controller {
    private $visitorModel;
    private $informationModel;
    private $categoryModel;
    private $itemModel;
    private $reviewModel;
    private $feedbackModel;

    public function __construct() {
        if(!isLoggedIn()) {
            redirect('login');
        }
        else if ($_SESSION['admin'] == false) {
            redirect('user');
        }
    }
}
```

```

    }
    $this->visitorModel = $this->model('Visitor');
    $this->informationModel = $this->model('Information');
    $this->categoryModel = $this->model('Category');
    $this->itemModel = $this->model('Item');
    $this->reviewModel = $this->model('Review');
    $this->feedbackModel = $this->model('Feedback');
}

public function index() {
    $visitors = $this->visitorModel->total();
    $information = $this->informationModel->selectById(1);
    $categories = $this->categoryModel->selectAll();
    $items = $this->itemModel->selectAll();
    $reviews = $this->reviewModel->selectAll();
    $feedback = $this->feedbackModel->selectAll();

    $data = [
        'visitors' => $visitors,
        'information' => $information,
        'categories' => $categories,
        'items' => $items,
        'reviews' => $reviews,
        'feedback' => $feedback
    ];

    $this->view('Admin', $data);

    flash('message');
}

public function updateInformation() {
    try {
        if ($_SESSION['admin'] == true) {
            if($_SERVER['REQUEST_METHOD'] == 'POST') {
                $POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

                $data = [
                    'id' => trim($_POST['id']),
                    'name' => trim($_POST['name']),
                    'phone' => trim($_POST['phone']),
                    'email' => trim($_POST['email']),
                    'address' => trim($_POST['address']),
                    'message' => trim($_POST['message']),
                ];

                if ($this->informationModel->update($data)) {
                    flash('message', 'Операція успішна! ');
                    redirect('admin');
                }
            }
        }
    }
    catch (Exception $e) {
        flash('message', 'Операція не успішна! ');
        redirect('admin');
    }
}

public function createCategory() {
    try {

```

```

        if ($_SESSION['admin'] == true) {
            if($_SERVER['REQUEST_METHOD'] == 'POST') {
                $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

                $data = [
                    'name' => trim($_POST['name'])
                ];

                if ($this->categoryModel->insert($data)) {
                    flash('message', 'Операція успішна! ');
                    redirect('admin');
                }
            }
        }
    }
    catch (Exception $e) {
        flash('message', 'Операція не успішна! ');
        redirect('admin');
    }
}

public function updateCategory() {
    try {
        if ($_SESSION['admin'] == true) {
            if($_SERVER['REQUEST_METHOD'] == 'POST') {
                $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

                $data = [
                    'id' => trim($_POST['id']),
                    'name' => trim($_POST['name'])
                ];

                if ($this->categoryModel->update($data)) {
                    flash('message', 'Операція успішна! ');
                    redirect('admin');
                }
            }
        }
    }
    catch (Exception $e) {
        flash('message', 'Операція не успішна! ');
        redirect('admin');
    }
}

public function deleteCategory() {
    try {
        if ($_SESSION['admin'] == true) {
            if($_SERVER['REQUEST_METHOD'] == 'POST') {
                $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

                $data = [
                    'id' => trim($_POST['id']),
                ];

                if ($this->categoryModel->delete($data)) {
                    flash('message', 'Операція успішна! ');
                    redirect('admin');
                }
            }
        }
    }
}

```



```

        flash('message', 'Операція не успішна! ');
        redirect('admin');
    }
}

public function updateFeedback() {
    try {
        if ($_SESSION['admin'] == true) {
            if($_SERVER['REQUEST_METHOD'] == 'POST') {
                $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

                $data = [
                    'id' => trim($_POST['id']),
                    'name' => trim($_POST['name']),
                    'email' => trim($_POST['email']),
                    'text' => trim($_POST['text']),
                    'status' => trim($_POST['status']),
                ];

                if ($this->feedbackModel->update($data)) {
                    flash('message', 'Операція успішна! ');
                    redirect('admin');
                }
            }
        }
        catch (Exception $e) {
            flash('message', 'Операція не успішна! ');
            redirect('admin');
        }
    }

    public function deleteFeedback() {
        try {
            if ($_SESSION['admin'] == true) {
                if($_SERVER['REQUEST_METHOD'] == 'POST') {
                    $_POST = filter_input_array(INPUT_POST,
FILTER_SANITIZE_FULL_SPECIAL_CHARS);

                    $data = [
                        'id' => trim($_POST['id']),
                    ];

                    if ($this->feedbackModel->delete($data)) {
                        flash('message', 'Операція успішна! ');
                        redirect('admin');
                    }
                }
            }
        }
        catch (Exception $e) {
            flash('message', 'Операція не успішна! ');
            redirect('admin');
        }
    }
}
}

```

ДОДАТОК Б. Слайди мультимедійної презентації



ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
ОНТУ»

Розробка веб-орієнтованої системи для ведення кав'ярного бізнесу

Курілець Наталія Андріївна

Загальні відомості

У сучасному світі кав'ярні стають соціальними центрами для роботи, зустрічей і відпочинку. Зростання популярності кавової культури створює попит на ефективні веб-орієнтовані системи управління. Вони оптимізують процеси, забезпечують швидкість та зручність у діяльності, дозволяють швидше реагувати на зміни в попиті та прискорюють обробку замовлень. Це підвищує конкурентоспроможність і стабільність бізнесу. Розроблена система буде корисною для власників кав'ярень, допомагаючи підтримувати ефективне управління і високий рівень обслуговування клієнтів.

Колірна гама веб-застосунку



Зелені кольори:
КАВОВІ ЛИСТЯ

Темні кольори:
обсмажені зерна

Схема технологій веб-застосунку

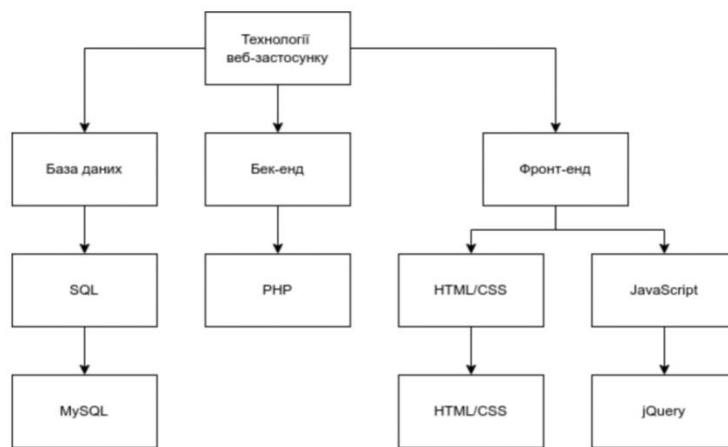


Схема мапи веб-застосунку

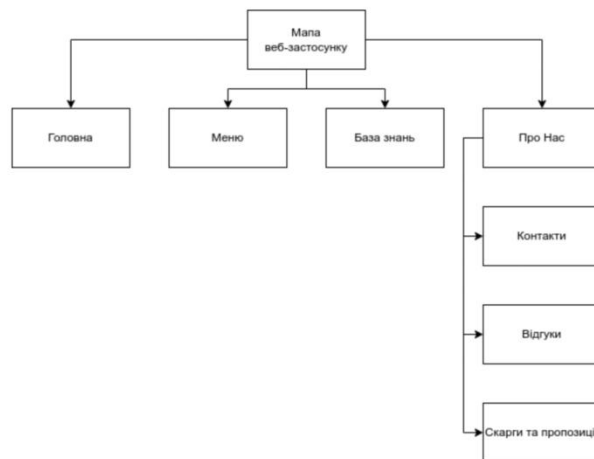


Схема бази даних веб-застосунку

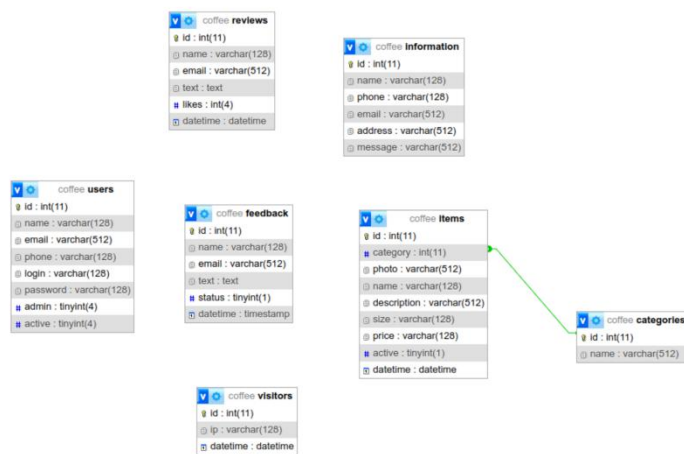
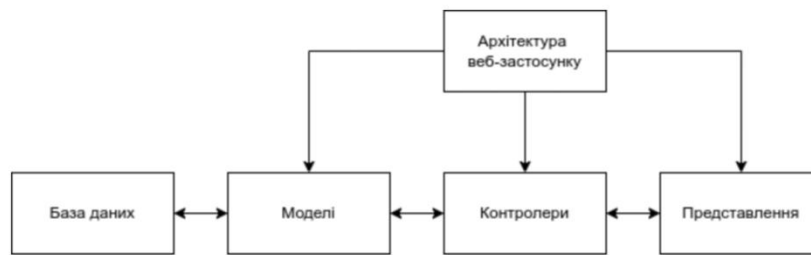


Схема архітектури веб-застосунку



```
File Edit Selection View Go Run Terminal Help
EXPLORER
APP
  app
  config
  controllers
    AdminController.php
    ContactsController.php
    FeedbackController.php
    HomeController.php
    KnowledgeController.php
    LoginController.php
    MenuController.php
    ReviewsController.php
  models
    CategoryModel.php
    FeedbackModel.php
    InformationModel.php
    ItemModel.php
    ReviewModel.php
    UserModel.php
    VisitorModel.php
  views
    template
      AdminView.php
      ContactsView.php
      FeedbackView.php
      HomeView.php
      KnowledgeView.php
      LoginView.php
      MenuView.php
      ReviewsView.php
    .htaccess
  core
  helpers
  libraries
  utils
  public
  assets
  db
  uploads
  .htaccess
  index.php
  manifest.json
OUTLINE
TIMELINE
MenuView.php X
app > views > MenuView.php
1 <!DOCTYPE html>
2 <html data-bs-theme="light" lang="en">
3
4 <head>
5   <?php require_once APPROOT . '/views/template/meta.php'; ?>
6   <?php require_once APPROOT . '/views/template/stylesheets.php'; ?>
7 </head>
8
9 <body class="bg-white">
10 <?php require_once APPROOT . '/views/template/scripts.php'; ?>
11 <?php require_once APPROOT . '/views/template/navmenu.php'; ?>
12 <?php require_once APPROOT . '/views/template/header.php'; ?>
13
14 <section class="p-5">
15   <div class="container">
16     <?php foreach ($data['itemsByCategory'] as $categoryName => $items) : ?>
17       <h2 ></h2>
18       <h1 class="text-success text-center m-3"><?php echo $categoryName; ?></h1>
19       <div class="row row-cols-3 justify-content-center align-items-center">
20         <?php if (empty($items)) : ?>
21           <?php foreach ($items as $item) : ?>
22             <div class="col">
23               <div class="card" data-bss-hover-animate="pulse">
24                 name; ?>" />
25                 <div class="card-body">
26                   <h1 class="card-title"><?php echo $item->name; ?></h1>
27                   <p class="card-text"><?php echo $item->description; ?></p>
28                   <p class="card-text"><?php echo $item->size; ?></p>
29                   <h4 class="card-title text-success"><?php echo $item->price; ?></h4>
30                 </div>
31               </div>
32             </div>
33           <?php endforeach; ?>
34         <?php else : ?>
35           <?php require APPROOT . '/views/template/alert.php'; ?>
36         <?php endif; ?>
37       </div>
38     <?php endforeach; ?>
39   </div>
40 </section>
41
42 <?php require_once APPROOT . '/views/template/footer.php'; ?>
43 </body>
44 </html>
```

Процес-розробки веб-застосунку

Створення веб-сторінки "Меню", використовуючи Microsoft Visual Studio Code

Метод на отримання позицій меню з БД (SQL)

```
public function selectByCategory($category)
{
    // Query
    $this->db->query('SELECT * FROM items WHERE category = :category');

    // Bind value
    $this->db->bind(':category', $category);

    // Execute
    $results = $this->db->resultSet();

    // Return result
    return $results;
}
```

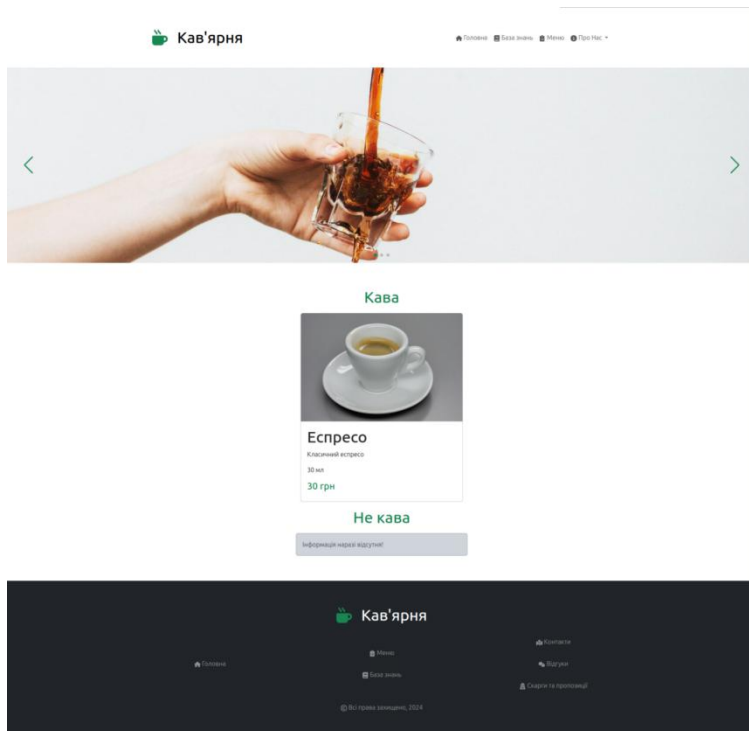
Основна функціональність веб-застосунку

Адміністратор

- Зміна інформації про заклад (в т.ч. логотипу для франшиз)
- Додавання меню
- Видалення відгуків

Користувач

- Перегляд інформації
- Додавання відгуків та вподобайок на них



Тестування основної веб-сторінки "Меню"

Відповідає за відображення асортименту кав'ярні

Дякую за увагу

Готова вислухати запитання від шановної комісії

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Курілець Наталії Андріївни

(прізвище, ім'я та по батькові)

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: Розробка веб-орієнтованої системи для ведення кав'ярного бізнесу

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 78 сторінки. У пояснювальній записці описано етапи розробки веб-орієнтованої системи для ведення кав'ярного бізнесу засобами PHP та MySQL. Графічна частина складається з окремих слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувачка освіти Курілець Наталія поступово та послідовно виконувала всі етапи, проявляла ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувачка освіти виконувала самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувачка освіти Курілець Наталія під час роботи над дипломним проектом вивчила достатньо багато літературних та інтернет-джерел за даною тематикою.

Вважаю, що теоретична підготовка дипломниці достатня і вона готова до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувачка освіти Курілець Наталія показала вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування та математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як MS VS Code, PHP, MySQL, MS PowerPoint, MS Visio та ін.

Оцінка розрахункової частини Відмінно
Оцінка графічної частини Добре
Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту Жадан Артур Сергійович

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний фаховий коледж ОНТУ», викладач спецдисциплін циклової комісії комп'ютерної техніки та програмної інженерії

Підпис 

«10» 06 2024 р.

РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Курілець Наталії Андріївни

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Жадан Артур Сергійович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка веб-орієнтованої системи для ведення кав'ярного бізнесу

Обсяг розрахунково-пояснювальної записки 78 сторінок

Обсяг графічної (презентаційної) частини 12 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі управління кав'ярним бізнесом та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту _____

Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 12 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скриншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки відмінна, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту Реалізовано веб-орієнтовану систему для ведення кав'ярного бізнесу, що дозволяє франшизам кав'ярень управляти своїм бізнесом, виконуючи менеджмент меню та відгуків.

Стилізація графічного інтерфейсу відповідає тематиці обраної предметної області.

Розроблена веб-система добре підходить для малого бізнесу.

д) основні недоліки дипломного проекту _____

1. У роботі не представлено блок-схем алгоритмів для створених скриптів;

2. При впровадженні веб-системи у сферу середнього бізнесу, можуть виникнути певні складнощі у зв'язку із обмеженнями веб-системи;

3. Наявні деякі помилки оформлення тексту пояснювальної записки

Оцінка розрахункової частини _____ Відмінно

Оцінка графічної частини _____ Добре

Загальна оцінка _____ Відмінно

Прізвище, ім'я, по батькові рецензента _____ Васіліу Євген Вікторович

Місце роботи і посада рецензента _____ Державний університет інтелектуальних технологій і зв'язку, д.т.н., проф. кафедри КБ та ТЗІ



Підпис _____

06 _____ 2024 р.

Ім'я користувача:
Катерина Григоріївна Краснокутська

ID перевірки:
1016335982

Дата перевірки:
08.06.2024 18:55:38 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
08.06.2024 18:58:22 EEST

ID користувача:
100011688

Назва документа: 4РП-07_Курілець_Н

Кількість сторінок: 47 Кількість слів: 6626 Кількість символів: 51534 Розмір файлу: 10.02 MB ID файлу: 1016136706

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

4.94%
Схожість

Найбільша схожість: 0.71% з Інтернет-джерелом (<https://gitlab.fdmci.hva.nl/oordi002/paymentplatform/-/commit/a1c0...>)

4.94% Джерела з Інтернету 384

Сторінка 49

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи 3

Підозріле форматування 12 сторінок

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Курілець Наталія Андріївна,
здобувачка освіти гр. 4РП-07, та

Жадан Артур Сергійович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

**«Розробка веб-орієнтованої системи для ведення кав'ярного бізнесу»
(авторка роботи – Курілець Н.А., керівник роботи – Жадан А.С.)**

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

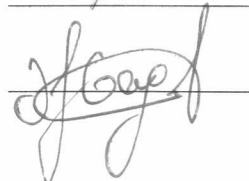
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Курілець Н.А. /

Керівник



/ Жадан А.С. /

«10» червня 2024 р.