

ЗМІСТ

АНОТАЦІЯ	7
ВСТУП	8
1 ЗАГАЛЬНИЙ РОЗДІЛ	9
1.1 Аналітичний огляд існуючих рішень	9
1.2 Технічне завдання.....	12
1.2.1 Найменування та область застосування.....	12
1.2.2 Призначення розробки.....	13
1.2.3 Вимоги до функціоналу web-сайту	13
1.2.4 Вимоги до програмної документації	15
1.2.5 Техніко-економічні показники	16
1.2.6 Стадії та етапи розробки.....	16
1.2.7 Порядок контролю та прийому.....	17
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ	19
2.1 Розробка структури сайту і web-сторінок	19
2.2 Створення та верстка сторінок сайту.....	22
2.3 Розробка структури базт даних сайту	23
2.4 Програмування сайту.....	25
2.4.1 Написання клієнтської частини	28
2.4.2 Розробка панелі адміністрування сайту	30
2.5 Тестування web-сайту.....	31
3 СПЕЦІАЛЬНИЙ РОЗДІЛ	35
3.1 Інструкція з розміщення сайту в Інтернеті.....	35
3.2 Інструкція з обслуговування та наповнення сайту.....	37
3.3 Інструкція з поширення та підтримки сайту	37

					2024.ДП.122.421.15.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка вебсайту "Кулінарний асистент"		
Розроб.		Линда В. І.					
Перевір.		Марціаш Г. Я.					
Реценз.							
Н. Контр.		Приймак В.А.					
Затверд.					Лім. Арк. Аркушів		
					5		
					ВСП «ТФК ТНТУ» КН-421 Тернопіль - 2024		

4 ЕКОНОМІЧНИЙ РОЗДІЛ.....	40
4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт	40
4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи	42
4.3 Розрахунок матеріальних витрат.....	44
4.4 Розрахунок витрат на електроенергію	44
4.5 Розрахунок суми амортизаційних відрахувань вебсайту «Кулінарний асистент»	45
4.6 Обчислення накладних витрат.....	46
4.7 Складання кошторису витрат та визначення собівартості веб-застосунку “Кулінарний Асистент”	46
4.8 Розрахунок ціни робіт.....	47
4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень	47
5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ..	50
5.1. Заходи щодо забезпеченню безпечної роботи при ремонті технологічного обладнання.....	50
5.2 . Фізичні основи електробезпеки.	52
ВИСНОВКИ.....	55
ПЕРЕЛІК ПОСИЛАНЬ.....	56
ДОДАТКИ.....	57
Додаток А – лістинг файлу “APP.js”	57
Додаток Б – лістинг файлу “Home.jsx”	59
Додаток В – лістинг файлу “RecipeItem.jsx ”	60
Додаток Г – лістинг файлу “AuthContext.jsx”	64
Додаток Е – лістинг файлу “authorised.js”	66
Додаток Є – лістинг файлу “UserAccount.jsx”	68
Додаток Ж – лістинг файлу “DishController.java”	71

АНОТАЦІЯ

Тема дипломного проекту: Розробка вебсайту «Кулінарний асистент»
Мета проекту – розробка вебсайту підбору кулінарних рецептів, який буде сприяти просуванню кулінарної культури України.

Пояснювальна записка складається з 5 розділів.

У першому розділі виконано аналітичний огляд існуючих рішень.
Описано призначення даної розробки, вимоги до програмного, апаратного забезпечення та ведення документації та технічне завдання.

У другому розділі дається загальний опис задачі та її специфічні особливості. Описано методи побудови сайту з використанням вибраних мов програмування, баз даних та систем керування контентом. Здійснюється розробка проекту та модуля часу роботи для розроблюваного сайту.

В спеціальному розділі описано процедури по розміщенню та наповненню і вебсайту в мережі Інтернет. Також описано створення та використання тестів і тестових програм та інструкцію з експлуатації вебсайту.

В економічному розділі описується детальний опис обрахування технічно-економічних показників на етапі розробки проекту.

В розділі охорони праці описується техніка безпеки та принципи її дотримання, а також екологічні вимоги до проекту.

Обсяг пояснювальної записки 73 сторінки.

До складу дипломного проекту входить графічна частина, яка складається з структурної схеми програми, блок-схеми модуля, техніко-економічних показників, тексту модуля, що виконані на окремих аркушах формату А1.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Комп'ютерні технології відіграють важливу роль у сучасному суспільстві, докорінно змінюючи всі його аспекти та проникаючи в усі сфери людської діяльності - від науки та медицини до роботи, освіти та розваг. Ці технології революціонізують спосіб, у який ми спілкуємося, працюємо, навчаємося, досліджуємо та розважаємося, надаючи нові можливості для розвитку та вдосконалення. Розвиток Інтернету та мобільних технологій робить інформацію доступною будь-де і будь-коли та полегшує доступ до інформації та ресурсів. Соціальні мережі, електронна пошта, месенджери та інші засоби комунікації сприяють глобалізації та соціальній згуртованості, дозволяючи людям спілкуватися через географічні кордони. Це особливо важливо для бізнесу, створюючи нові можливості для співпраці, відкриваючи міжнародні ринки та підвищуючи ефективність управління.

Інформаційні технології є рушійною силою для інновацій у ряді секторів, включаючи бізнес. Дипломний проект надає унікальну можливість застосувати теоретичні знання на практиці та вирішити сучасні проблеми, пов'язані з використанням комп'ютерних технологій у бізнесі. Це дозволяє працювати над реальними проектами у сферах розробки програмного забезпечення, інформаційних систем та аналізу даних, вивчити передові методики та інструменти, освоїти нові методи та підходи, отримати цінний досвід роботи в команді та вирішувати проблеми на практиці.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

При виборі платформи для веб-додатків, яка сприяє наданню продовольчої допомоги, необхідно ретельно розглядати різні аспекти, що включають технічні вимоги, потреби користувачів та ефективність інтеграції з іншими системами.

Додаток повинен працювати безперебійно та без помилок. Це особливо важливо для платформи, що надає продовольчу допомогу, оскільки користувачі можуть залежати від нього у критичних ситуаціях. Надійність забезпечується вибором стабільної та добре підтримуваної платформи, яка вже зарекомендувала себе на ринку.

Однією з ключових вимог є простота використання для адміністраторів та кінцевих користувачів. Інтуїтивно зрозумілий інтерфейс, легко доступні функції та можливість швидкого навчання нових користувачів є важливими аспектами для успіху додатку.

Платформа повинна мати можливість масштабуватися з ростом обсягів роботи та інтегруватися з іншими сервісами. Це забезпечує гнучкість та можливість розширення функціональності за рахунок нових технологій та рішень.

На ринку існує безліч технологій, які можуть бути використані для створення веб-додатків, зокрема для платформи надання продовольчої допомоги.

React є однією з найпопулярніших бібліотек для створення інтерфейсів користувача. Вона відома своєю ефективністю у розробці компонентів із високим повторним використанням. Однак для початківців вона може вимагати певного часу на вивчення. Завдяки використанню віртуального DOM, React забезпечує високу продуктивність та швидкий рендеринг. Крім того, екосистема React постійно розширюється, включаючи такі інструменти,

як React Router для маршрутизації та Redux для управління станом. React також підтримує серверний рендеринг, що покращує SEO та зменшує час завантаження сторінок. Враховуючи ці переваги, React залишається вибором багатьох розробників для створення складних і масштабованих веб-додатків.

Axios - це бібліотека для виконання HTTP-запитів зручним інтерфейсом. Вона дозволяє ефективно управляти асинхронними запитам, але вимагає базових знань для налагодження. Axios підтримує обіцянки (promises), що спрощує роботу з асинхронним кодом, забезпечуючи легку інтеграцію з функціями async/await. Вона автоматично обробляє JSON-дані, роблячи процес обміну даними між клієнтом і сервером простішим.

Axios має інтуїтивно зрозумілий синтаксис для виконання HTTP-запитів, таких як GET, POST, PUT, DELETE тощо, і дозволяє легко налаштовувати заголовки, параметри запиту, тайм-аути та інші опції. Бібліотека підтримує інтерсептори запитів та відповідей, що дозволяє виконувати дії перед відправкою запиту або після отримання відповіді, наприклад, обробляти помилки або додавати токени аутентифікації. Axios забезпечує просту обробку помилок, дозволяючи легко визначати статуси помилок та реагувати на них.

Крім того, Axios дозволяє скасовувати запити за допомогою токенів скасування, що корисно для запобігання гонок запитів або непотрібних запитів у великих додатках. Вона підтримує роботу з файлами, дозволяючи завантажувати та відправляти файли на сервер, а також підтримує крос-доменні запити і автоматично обробляє політики CORS.

Завдяки своїй гнучкості та простоті використання, Axios залишається популярним вибором серед розробників для роботи з HTTP-запитами у веб-додатках.

React Router - це бібліотека для управління маршрутизацією у веб-додатках, створених на основі React. Вона забезпечує зручний спосіб організації навігації між різними компонентами та сторінками додатку. React

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Router підтримує динамічну маршрутизацію, що дозволяє створювати маршрути на основі стану або властивостей додатку.

React Router використовує декларативний підхід до визначення маршрутів, що полегшує читання та підтримку коду. Бібліотека надає різноманітні компоненти для роботи з маршрутизацією, такі як `<BrowserRouter>`, `<Route>`, `<Link>` та `<Switch>`, які дозволяють легко створювати та управляти маршрутами. Вона також підтримує вкладені маршрути, що забезпечує гнучкість у побудові складних ієрархій навігації.

Крім того, React Router надає можливість обробки редиректів та захисту маршрутів, що корисно для реалізації логіки аутентифікації та авторизації у додатках. Бібліотека забезпечує підтримку хеш- та історичної маршрутизації, що дозволяє вибирати відповідний метод маршрутизації залежно від потреб додатку.

Завдяки своїй гнучкості, розширюваності та простоті використання, React Router є одним з найбільш популярних рішень для управління маршрутизацією у веб-додатках на базі React.

Java та Spring відомі своєю масштабованістю і продуктивністю у створенні високопродуктивних додатків. Однак вони вимагають значного рівня знань і можуть бути складними для початківців у розробці та підтримці. Spring надає широкий спектр модулів і інструментів, таких як Spring Boot для швидкого створення додатків, Spring Security для забезпечення безпеки, та Spring Data для роботи з базами даних. Ці інструменти дозволяють спростити розробку складних корпоративних додатків, але вимагають глибокого розуміння концепцій та архітектури Spring. Незважаючи на криву навчання, переваги у вигляді надійності, масштабованості та підтримки великої спільноти розробників роблять Java та Spring популярним вибором для побудови багатьох додатків.

Вибір платформи для веб-додатку для надання продовольчої допомоги залежить від конкретних потреб проекту, рівня досвіду команди розробників та наявних ресурсів. Важливо враховувати всі аспекти, включаючи технічні

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

вимоги, вимоги до інтерфейсу та потреби у масштабованості та інтеграції з існуючими системами.

Усе розглянуте підкреслює важливість правильного вибору технологій для досягнення успіху веб-застосунку для надання продовольчої допомоги. Тільки після ретельного аналізу і оцінки всіх аспектів можна зробити обґрунтоване рішення щодо платформи, яка найкращим чином задовольнить потреби проекту та його користувачів.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Мета розробки web-сайту "Кулінарний асистент", це надання користувачам широкого вибору рецептів, основними особливостями та перевагами є:

- Широкий вибір рецептів, від простих до складних, від традиційних до екзотичних. Колекція налічуватиме тисячі рецептів, які враховують усі смакові палітри та дієтичні уподобання користувачів.
- Завдяки розширеним фільтрам користувач зможе швидко знаходити рецепти за інгредієнтами, типом кухні, часом приготування та рівнем складності. Вдосконалений пошук дозволить точно підібрати страву для будь-якого особливого заходу.
- Деталізовані інструкції та фотографії, до кожного рецепту буде надано чіткі кулінарні інструкції та яскраві фотографії, що допоможуть користувачу крок за кроком створити смачну страву. Незалежно від досвіду користувача в готуванні, він з легкістю зможе створити майстер-п'єс кулінарії.
- Функція збереження улюблених рецептів, дозволить створити особисту колекцію улюблених страв, щоб мати миттєвий доступ до них в будь-який час. Додаток забезпечує зручне керування обраними та зберігає їх у безпечному місці.

1.2.2 Призначення розробки

Мета в веб-сайті "Кулінарний асистент" полягає в створенні зручного та ефективного онлайн-простору для надання високоякісних та смачних рецептів. Суть "Кулінарного асистента" в тому щоб забезпечити клієнтів докладною інформацією про страви, простим процесом готування та бездоганним інструктажем. Завдяки високому рівню сервісу та різноманітному вибору якісних рецептів, "Кулінарний асистент" має за мету завоювати довіру клієнтів та стати провідним лідером на ринку онлайн-послуг.

1.2.3 Вимоги до функціоналу web-сайту

В веб-сайті "Кулінарний асистент" буде реалізовано велику кількість функцій, правильний опис функціоналу допоможе з написанням контролерів серверної частини та правильній навігації web-інтерфейсу. Функції веб-сайту "Кулінарний асистент" будуть реалізовані між 2 типами користувачів "Користувач" та "Адміністратор" адміністратор успадкує всі функції користувача і розширить їх додатковим функціоналом.

Функції Адміністратора:

- Створення нових рецептів.
- Редагування наявних рецептів.
- Видалення рецептів.
- Створення нових кухонь
- Редагування наявних кухонь.
- Видалення кухонь.

Функції Користувача:

- Вивід всіх страв
- Вивід однієї страви
- Вивід всіх кухонь

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

- Вивід однієї кухні
- Додавлення страви до списку обраних
- Вивід всіх обраних страв
- Реєстрація нового користувача
- Авторизація наявного користувача

Вимоги до функціоналу веб-сайту "Кулінарний асистент" чітко визначають набір можливостей для двох типів користувачів — "Користувач" і "Адміністратор". Реалізація цих функцій є ключовим етапом у процесі розробки, оскільки вони визначають як серверну частину, так і web-інтерфейс сайту. Адміністратор, успадковуючи функціонал користувача, розширює його можливості зі створення, редагування та видалення рецептів і кухонь. Користувач, з свого боку, матиме зручний доступ до різноманітного контенту, зокрема страв і кухонь, можливість управління обраними стравами, а також функції реєстрації та авторизації. Чітко визначений функціонал допоможе забезпечити якісну і зручну роботу веб-сайту для всіх його користувачів.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

1.2.4 Вимоги до програмної документації

Документація API повинна включати детальний опис кожної кінцевої точки, методів (GET, POST, PUT, DELETE) та їх параметрів, включаючи обов'язкові та необов'язкові поля. Надаватимуться приклади JSON запитів та відповідей для кожної кінцевої точки API. Swagger буде використовуватись для автоматичного генерування та підтримки документації API.

Інструкція з налаштування зв'язку між фронтом на React JS і бекендом на Java Spring включає покрокове керівництво. Налаштування Java Spring для обробки запитів від фронту. Використання Postman для тестування API, включаючи створення колекцій запитів, налаштування середовища та збереження прикладів відповідей.

Документація також має містити реальні приклади коду на React JS та Java Spring, що демонструють використання API, а також описання типових сценаріїв, такі як отримання даних про рецепти, створення нового рецепта, оновлення інформації про рецепти, видалення рецепта тощо.

Документація по користувацькому інтерфейсу повинна містити опис компонентів інтерфейсу на React JS, включаючи їхню структуру та стиль. Показані приклади реалізації, такі як головна сторінка, сторінка користувача елементи рецептів тощо. Рекомендації щодо кращих практик користувацького досвіду та інтерфейсу для інтеграції з API.

Описуються також процедури звернень для подання запитів на підтримку, включаючи кроки для відтворення проблеми, надання логів та іншої релевантної інформації.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

1.2.5 Техніко-економічні показники

У даному розділі будуть розглянуті техніко-економічні аспекти створення та експлуатації веб-сайту "Кулінарний асистент". Оцінка вартості розробки детально описана в четвертому розділі пояснювальної записки.

Передбачувані витрати на розробку веб-сайту:

Кількість годин - 162 години.

Витрати на оплату праці - 17 008 грн.

Єдиний соціальний внесок – 3741 грн.

Матеріальні витрати – 480 грн.

Витрати на електроенергію – 261 грн.

Амортизація відрахувань – 1200 грн.

Орієнтовані витрати на програмування, дизайн та тестування функціоналу сайту складають 22690 грн.

1.2.6 Стадії та етапи розробки

Розробка вебсайту “Кулінарний асистент” проводилася поетапно, щоб забезпечити його високу якість та відповідність потребам цільових користувачів. Процес складався з чітко визначених етапів, кожен з яких мав свої конкретні цілі та завдання.

Перший етап - аналіз та визначення вимог. Цей етап включає дослідження для розуміння потреб і очікувань цільових користувачів, а також вивчення конкурентів, щоб оцінити подібні інтернет-магазини та їхнє місце на ринку. На цьому етапі також формуються функціональні вимоги до веб-додатку для забезпечення його зручності та ефективності.

Другий етап - проектування та розробка. На цьому етапі визначається структура веб-застосунку, організація інформації та доступність різних сторінок. Створюються макети для візуалізації інтерфейсу та візуальний

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

стиль, який відповідав бренду "Кулінарний асистент" і приваблював користувачів.

Третій етап - функціональна розробка. На цьому етапі реалізуються основні функції веб-додатку, такі як каталог продуктів, кошик, система замовлень, різні способи оплати та система управління.

Четвертий етап - тестування та виправлення помилок. Всі функції веб-додатку ретельно проходять тестування виявляються помилки та виправляються для забезпечення найвищої якості користувацького досвіду. Тестування проводиться на різних пристроях для переконання в коректній роботі на комп'ютерах, планшетах та мобільних телефонах.

П'ятий етап - запуск та експлуатація веб-застосунку. Це включало підготовку до запуску, оптимізацію для пошукових систем, наповнення контентом та створення маркетингових матеріалів. Після запуску веб-додаток став доступним для широкого кола користувачів.

Останній етап - підтримка та постійне вдосконалення. Надавалася підтримка користувачам, регулярно виправлялися помилки та додавалися нові функції на основі отриманих відгуків та аналізу даних.

1.2.7 Порядок контролю та прийому

Для забезпечення готовності веб-проекту до публікації і відповідності всім вимогам, необхідно провести детальну перевірку. Цей процес відомий як процедура тестування та приймання і включає кілька основних етапів.

Спочатку виконується перевірка конфігурації, щоб переконатись, що браузер, веб-сервер, база даних і сам веб-проект на стороні сервера налаштовані належним чином. Це забезпечує стабільну роботу онлайн-каталогу та сумісність з браузером користувача.

Після цього важливо перевірити функціональні можливості веб-додатка. Це включає тестування всіх основних функцій, передбачених специфікацією проекту. Кожна взаємодія користувача з додатком повинна бути перевірена на

коректність і зручність використання. Також варто перевірити, що всі зміни даних, введені користувачем, правильно відображаються і зберігаються на сервері.

Останнім етапом є тестування безпеки. Важливо переконатись, що веб-додаток захищений від різних видів атак, таких як SQL-ін'єкції, XSS атаки та інші вразливості. Забезпечення безпеки даних і персональної інформації користувачів є критично важливим аспектом у процесі тестування та приймання веб-проекту.

Цей послідовний процес допомагає забезпечити якість та надійність веб-додатка перед його випуском в продакшн.

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури сайту і web-сторінок

Розробка вебсайту “Кулінарний асистент” для управління компонентами веб-сайту включає наступні функціональні елементи.

Меню навігації цей елемент дозволяє користувачам легко переміщатися між різними сторінками додатку. Меню навігації містить посилання на головну сторінку, сторінки рецептів, сторінки користувачів і інші важливі розділи.

Блок сортування та фільтрації компонентів цей елемент надає користувачам можливості сортування компонентів за різними критеріями, такими як ціна, виробник, рейтинг тощо. Фільтри дозволяють звужити результати пошуку, щоб знайти компонент, який підходить для конкретних потреб.

Компонент рецептів важливий елемент для управління рецептами, що дозволяє користувачам переглядати та вибирати рецепти.

Сторінка користувача це централізована платформа для доступу до інформації про обліковий запис користувача та його рецепти. Вона надає можливості перегляду профілю і виконання інших дій залежно від рівня доступу, включаючи адміністраторські функції для управління рецептами.

При розробці інтерфейсу важливо звернути увагу на зрозумілість та простоту використання кожного функціонального елемента. Детальні описи та візуалізації допоможуть зрозуміти дизайн цих елементів та їх взаємодію з користувачем. (див. рис. 2.1)

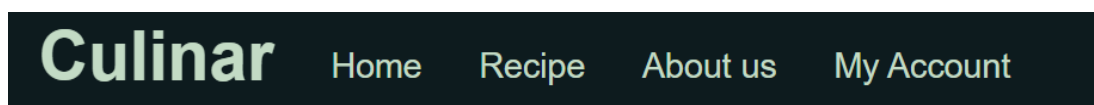


Рисунок 2.1 – Навігаційне меню вебсайту «Кулінарний асистент»

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

Блоки сортування дозволяють користувачам веб-додатків для управління компонентами сортувати компоненти за різними критеріями, такими як країна походження, тип і кулінарна складність. Це необхідний функціональний елемент, який повинен бути присутнім на будь-якому веб-ресурсі, щоб забезпечити зручність використання та швидкий доступ до необхідних даних. (див. рис. 2.2) ,

Відбір за країною

☐Українська кухня

☐Італійська кухня

☐Французька кухня

☐Китайська кухня

☐Японська кухня

☐Індійська кухня

Відбір за типом

☐Перші страви

☐Холодні страви

☐Закуси

☐Салати

Відбір за складністю

☐Легка

☐Середня

☐Важка

Пошук

Рисунок 2.2 – Блок сортування вебсайту «Кулінарний асистент»

Компонент рецептів відіграє важливу роль у веб-додатку, оскільки він дозволяє візуалізувати інформацію про рецепти та керувати нею. Цей компонент містить дані, які адміністратор вносить у додаток. Він слугує зручним інтерфейсом для перегляду, редагування та видалення інформації про продукт і робить роботу з додатком максимально приємною. Таким чином, компонент продукту виступає сполучною ланкою між користувачем і

системою, забезпечуючи зручну взаємодію та ефективне управління даними.
(див. рис. 2.3).

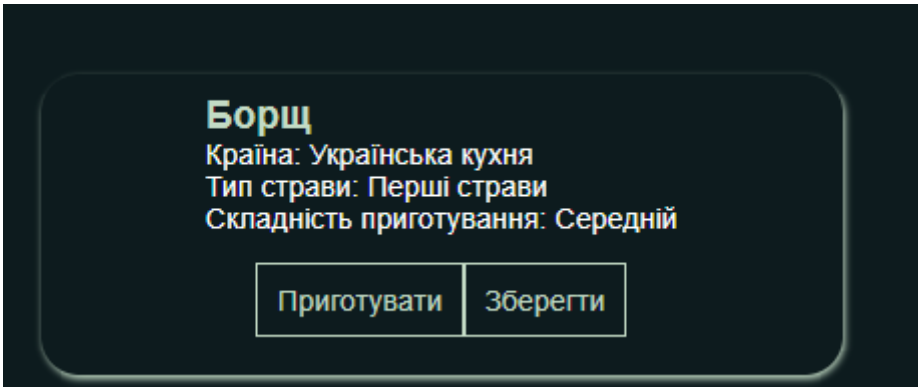


Рисунок 2.3 – Компонент рецепту вебсайту «Кулінарний асистент»

"Кулінарний асистент" пропонує зручну персональну сторінку, яка поєднує в собі профілі користувачів та управління рецептами. Платформа дозволяє користувачам переглядати свої персональні дані та виконувати різні дії відповідно до своїх прав. Адміністратори можуть додавати функції, пов'язані з рецептами, такі як додавання або редагування рецептів.
(див. рис. 2.4).

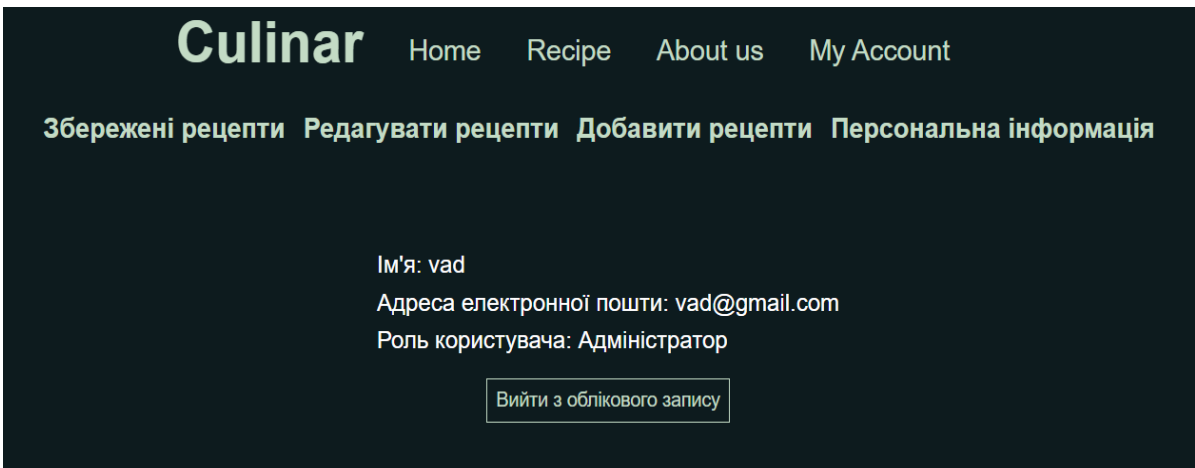


Рисунок 2.4 – Сторінка користувача вебсайту «Кулінарний асистент»

2.2 Створення та верстка сторінок сайту

Розробка сторінок веб-додатків - це складний процес, що базується на заздалегідь визначеній структурі та затвердженому макеті дизайну. Ефективне кодування вимагає використання відповідних інструментів, таких як Visual Studio Code, який надає необхідну функціональність для редагування веб-додатків.

Використання Visual Studio Code

Visual Studio Code є основним редактором для кодування веб-сторінок завдяки своїй потужності та універсальності. Visual Studio Code надає зручне середовище для написання, редагування та налагодження коду.

Розробка сторінок на основі дизайн-макету

Кожна сторінка веб-додатку розробляється відповідно до затвердженого дизайн-макету та керівництва по стилю. Це включає створення структури HTML, стилів CSS і, за необхідності, JavaScript для інтерактивності.

Структура HTML

Структура сторінок складається з різних елементів, таких як `

`, ``, `` і ``, які формують навігаційне меню та інші компоненти інтерфейсу. Наприклад, блок `

` використовується для групування елементів навігаційного меню, таких як кнопки та поля пошуку.

Стилізація за допомогою CSS

Стилізація використовує CSS з класами, що відповідають за зовнішній вигляд і поведінку кожного елемента. Наприклад, класи `searchzonewrapper`, `blue-button`, `red-button` і `sitename` використовуються для встановлення стилю (кольору, фону, шрифту тощо) відповідного елемента.

Модальні вікна

Модальні вікна, які використовуються для відображення додаткової інформації або підтвердження дій, також реалізуються за допомогою HTML і CSS. Клас модального вікна визначає стиль вікна, включаючи його розмір, колір фону та інші властивості. Елементи внутрішнього вмісту модального

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

вікна (наприклад, `modal-content`, `modal-header`, `modal-title`, `close`, `in-cart` тощо) також стилізуються за допомогою відповідних класів для досягнення бажаного вигляду та поведінки.

Розробка сторінок веб-додатків на основі заздалегідь визначеної структури та затвердженого дизайн-макету вимагає системного підходу, а використання сучасного інструментарію Visual Studio Code дозволяє розробникам ефективно створювати стильні та функціональні веб-інтерфейси.

2.3 Розробка структури базт даних сайту

На веб-застосунку “Кулінарний асистент” покращено та розширено функціонал системи управління базами даних за допомогою pgAdmin. Інструмент дозволяє ефективно керувати базами даних на основі PostgreSQL.

Переваги використання PostgreSQL з pgAdmin

Безкоштовне програмне забезпечення з відкритим вихідним кодом: pgAdmin є безкоштовним програмним забезпеченням з відкритим вихідним кодом і може використовуватися будь-яким користувачем. Це робить його привабливим варіантом для компаній та організацій з обмеженим бюджетом.

pgAdmin доступний для різних операційних систем, включаючи Windows, macOS і Linux. Це робить його корисним інструментом для користувачів, які працюють у різних середовищах.

pgAdmin має простий та інтуїтивно зрозумілий графічний інтерфейс, що робить його легким у вивченні та використанні.

pgAdmin надає широкий спектр функцій для управління базами даних PostgreSQL, включаючи створення і редагування таблиць, перегляд і редагування даних і виконання SQL запитів.

pgAdmin також має інтерфейс командного рядка, який дозволяє користувачам виконувати завдання адміністрування баз даних за допомогою команд.

PostgreSQL - це безкоштовне програмне забезпечення з відкритим вихідним кодом, доступне для всіх користувачів. Це гарантує прозорість та надійність розробки, а також дозволяє співпрацювати та модифікувати код.

PostgreSQL відома своєю надійністю та відмовостійкістю. Вона використовується в багатьох критично важливих системах, де прості є неприпустимими.

PostgreSQL може масштабуватися для обробки високих навантажень і великих обсягів даних. Тому вона підходить для використання у великих проєктах і організаціях.

PostgreSQL пропонує широкий спектр функцій безпеки для захисту даних від несанкціонованого доступу.

PostgreSQL підтримує стандартну мову запитів SQL, що робить її легкою у вивченні та використанні для користувачів, знайомих з іншими базами даних SQL.

Переваги спільного використання pgAdmin і PostgreSQL:

Ефективне адміністрування: pgAdmin спрощує адміністрування баз даних PostgreSQL, роблячи цей процес більш зручним і продуктивним.

PostgreSQL надає потужну та гнучку платформу для зберігання та обробки даних. pgAdmin надає зручний інтерфейс для управління цими функціями.

як pgAdmin, так і PostgreSQL є безкоштовними, з відкритим вихідним кодом і доступними для кожного користувача.

обидва проєкти підтримуються активною спільнотою розробників та користувачів, які надають підтримку та ресурси.

Загалом, pgAdmin і PostgreSQL є потужними і гнучкими рішеннями для управління базами даних і пропонують багато переваг для користувачів усіх рівнів.

Система зберігання рецептів використовує таблицю "страви", в якій дані про кожну страву зберігаються в структурованому вигляді.

Структура таблиці "страви"

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

cuisine_id ідентифікатор країни страви.

Id унікальний ідентифікатор продукту.

Description опис рецепта страви.

Difficulty складність рецепта.

Name назва рецепта.

Type тип страви.

youtube_video_url посилання на відео з рецептом. Початкове завантаження даних про рецепт в базу даних відбувається за допомогою файлу MyApplicationRunner. Цей файл містить значення для таблиць `dish` і `cuisine`. Цей файл можна використовувати для завантаження інформації про різні аспекти рецепта.

Файл MyApplicationRunner забезпечує гнучкість і зручність при завантаженні даних. Це дозволяє легко додавати нову інформацію або оновлювати існуючу без необхідності вносити зміни в код програми.

Використання баз даних при створенні веб-сайтів:

Коли веб-сайт завантажується, система автоматично отримує і обробляє інформацію про продукт з бази даних. Ці дані використовуються для динамічної генерації веб-сторінок.

Перехід від використання LocalStorage до використання серверної бази даних з PGAdmin значно покращує збереження даних і доступ до них у веб-додатку.

2.4 Програмування сайту

Після завантаження веб-сайту система автоматично отримує та обробляє інформацію про рецепти з бази даних. Ці дані використовуються для динамічного генерування веб-сторінок, на яких користувачі можуть: (див. таблицю 2.1).

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 – Структура сценаріїв та їх призначення

Назва сценарію	Призначення
Register.jsx Login.jsx	Сторінки реєстрації/авторизації користувачів
Home.jsx	Головна сторінка сайту
Recipes.jsx	Сторінка рецептів, пошук рецептів за категоріями
UserAccount.jsx	Сторінка користувача, вихід з акаунта перегляд збережених страв
RecipeItem.jsx	Компонент рецепта, взаємодія з рецептом

Так, структура вебсайту "Кулінарний асистент" включає декілька файлів, які розділяють код на функціональні частини. Давайте розглянемо кожну функцію з цих файлів більш детально:

Файли Register.jsx та Login.jsx містять теги та логіку для реєстрації та автентифікації користувачів - візьмемо для прикладу Register.jsx.

Основною функцією, що містить логіку та макет сторінки, є Register(). Ця функція використовується для створення сторінки реєстрації та передачі даних на сервер.

Функція включає перевірку даних, яка перевіряє, чи відповідають адреса електронної пошти, ім'я та пароль заданим критеріям(див. рис. 2.5).

Файл Home.jsx містить компонент Home.jsx, який відповідає за роботу з головною сторінкою сайту.

Register

Email

Name

Password

Repeat Password

[Registered already?](#)

Рисунок 2.5 - Демонстрація реєстрації нового користувача

Основна функція компонента Home.jsx полягає у створенні та наповненні основної структури головної сторінки. Він отримує дані з боку сервера, динамічно заповнює сторінку контентом і повертає шаблон HTML-коду для відображення користувачеві (див. рис. 2.6).

Файл userAccount.jsx містить інформацію про користувачів і їх збережені рецепти, а також основну логіку управління рецептами. Цей файл додає, редагує та видаляє нові рецепти для адміністратора веб-додатку.

У ньому зберігається вся необхідна інформація про особистий кабінет користувача, включаючи персональні дані користувача і список збережених рецептів. Крім того, файл містить функції, які дозволяють системному адміністратору ефективно керувати цими рецептами, дозволяючи додавати нові рецепти, змінювати або видаляти існуючі.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

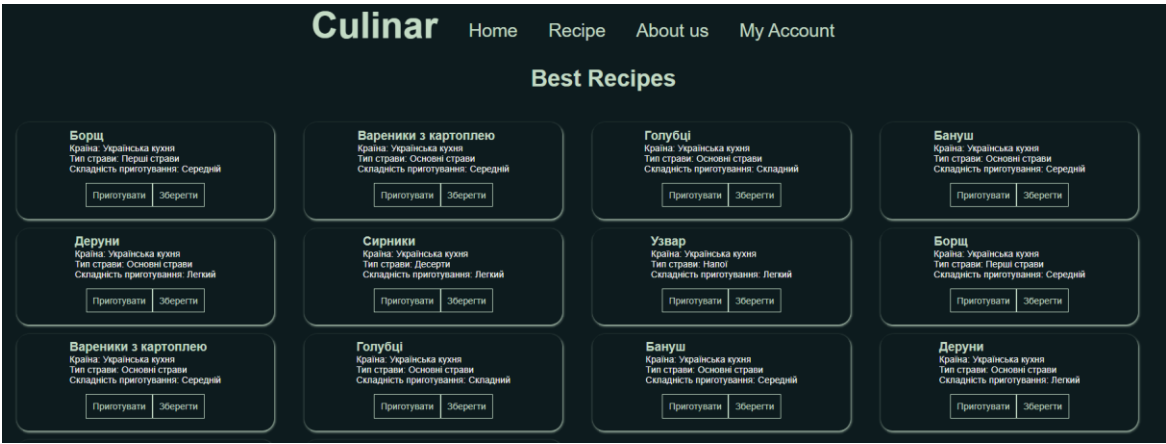


Рисунок 2.6 - Вид головної сторінки

Такий підхід не тільки спрощує управління рецептами, але й гарантує, що рецепти є безпечними та доступними для користувачів з відповідними правами.(див. рис. 2.7)

```
useEffect(() => {
  if (isAuthorized) {
    const role = checkUserRole(storedRole);
    setUserRole(role);
  }
}, [isAuthorized, storedRole]);
const handleNavigateToUserAccInfo = () => {
  setSelectedPage('userAccInfo');
};
const handleNavigateToFavouriteDishes = () => {
  setSelectedPage('userFavouriteDishes');
};
const handleNavigateToEditRecipes = () => {
  setSelectedPage('editRecipes');
};
return (
  <>
    <Header isAuthorized={isAuthorized} />
    <div className="main-content-top">
      <ul>
        <li><a onClick={handleNavigateToFavouriteDishes}><h4>Збережені рецепти</h4></a></li>
        <li><a onClick={handleNavigateToEditRecipes}><h4>Редагувати рецепти</h4></a></li>
        <li><a><h4>Додати рецепти</h4></a></li>
        <li><a onClick={handleNavigateToUserAccInfo}><h4>Персональна інформація</h4></a></li>
      </ul>
    </div>
    <div>
      {selectedPage === 'userAccInfo' && <UserAccInfo role={storedRole} email={storedEmail} sub={storedSub} />}
      {selectedPage === 'userFavouriteDishes' && <UserFavoriteDishes />}
      {selectedPage === 'editRecipes' && <EditRecipes />}
      {selectedPage === 'UpdateItem' && <UpdateItem />}
    </div>
  </>
);
```

Рисунок 2.7 - Частина коду кабінета користувача

2.4.1 Написання клієнтської частини

Клієнтська частина веб-додатку “Кулінарний асистент” є інтерфейсом, через який відвідувачі взаємодіють з додатком. Вона включає різноманітні

сторінки та функції, які спрощують процес покупки товару. Основні компоненти та функції клієнтської частини представлені нижче.

1. Головне меню

Головне меню: головна сторінка містить посилання на основні розділи
Банери: відображають нові та популярні продукти.

Рекомендовані продукти: показують блок товарів, рекомендованих покупцям менеджером сайту.

2. Сторінка/каталог рецептів

Список рецептів: відображає список усіх рецептів.

Фільтри: інструмент для фільтрації продуктів за різними параметрами, такими як країна походження, тип, складність рецепта тощо.

3. Сторінки рецептів

Відео-інструкція, що пояснює, як готувати рецепт.

Інгредієнти: список інгредієнтів, необхідних для приготування рецепту.

Як приготувати: покрокові інструкції з приготування рецепту з короткими вказівками.

4. Форма реєстрації нового облікового запису

Поля для введення персональних даних користувача, наприклад, імені, прізвища, електронної адреси тощо.

Поле для введення пароля. Містить вимоги до довжини та складності пароля.

Підтвердження пароля для перевірки правильності введеного пароля.

Кнопка реєстрації для відправки реєстраційної форми.

5. Форма авторизації для існуючих облікових записів

Поле для введення логіну або ідентифікатора користувача.

Поле для введення пароля для входу в систему.

Кнопка "Увійти" для підтвердження введених даних та авторизації.

6. Сторінка користувача

Відображення особистої інформації: ім'я користувача, адреса електронної пошти.

Перегляд ролі користувача: інформація про статус користувача (користувач або адміністратор).

Перегляд збережених рецептів: доступ до списку рецептів, які користувач зберіг у своєму обліковому записі.

Керування рецептами: дозволяє користувачам створювати нові рецепти, редагувати або видаляти існуючі рецепти.

2.4.2 Розробка панелі адміністрування сайту

Адміністрування вебсайту "Кулінарний асистент" включає в себе набір базових функцій, спрямованих на зручне та ефективне управління ресурсами та контентом. Модуль управління користувачами забезпечує безпеку і надійність реєстрації, аутентифікації та авторизації нових користувачів на сайті. Модуль містить форми для створення нових облікових записів і входу в систему існуючих користувачів, а також механізми зберігання і захисту конфіденційних даних.

Управління користувачами - реєстрація та аутентифікація користувачів. Реалізація форм для реєстрації нових користувачів та входу існуючих користувачів. Сюди входить зберігання даних користувачів, перевірка введених даних і забезпечення безпеки паролів за допомогою хешування.

Управління рецептами - додавання, редагування та видалення рецептів. Інтерфейс для адміністраторів для додавання нових рецептів та редагування або видалення існуючих рецептів за потреби. Це можна переглянути з адміністративної панелі або спеціального інтерфейсу користувача з додатковими правами доступу.

Управління категоріями і тегами - категоризація рецептів. Категорії і теги можна додавати і редагувати, щоб класифікувати рецепти і полегшити їх пошук користувачем.

Фільтрація рецептів - фільтрація за ключовими словами, інгредієнтами, назвою рецепту тощо. Функція фільтрації для точного відбору рецептів за

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

різними параметрами, такими як складність, тип кухні, час приготування тощо.

Безпека та захист даних - механізми аутентифікації та авторизації використовуються для забезпечення безпеки, захисту доступу до адміністративних функцій та персональних даних користувачів, а також для перевірки даних, що вводяться на стороні клієнта та сервера.

Веб-застосунок "Кулінарний асистент" надає комплексну функціональність для ефективного управління контентом і ресурсами. Він охоплює всі аспекти, необхідні для створення безпечного, зручного та функціонального веб-застосунку.

2.5 Тестування web-сайту

План тестування для перевірки роботи додатку:

- Протестувати фільтрацію рецептів за різними категоріями.
- Перевірити можливість редагування елемента та збереження змін.
- Протестувати видалення елемента і переконатись, що він успішно видаляється.
- Перевірити можливість додавання нових елементів і переконатись, що вони коректно відображаються на веб-сторінці.
- Перевірити вигляд веб-сторінки на екранах з низькою роздільною здатністю, щоб переконатися, що всі елементи відображаються коректно і не накладаються один на одного.

Виконується перевірка сортування, щоб переконатися, що список елементів відсортовано коректно. (див. рис. 2.8)

Фільтрація за критеріями пройшла успішно (див. рис. 2.9)

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

Відбір за країною
☒ Українська кухня
☐ Італійська кухня
☐ Французька кухня
☐ Китайська кухня
☐ Японська кухня
☐ Індійська кухня

Відбір за типом
☒ Перші страви
☐ Холодні страви
☐ Закуси
☐ Салати

Відбір за складністю
☐ Легка
☒ Середня
☐ Важка

Пошук

Рисунок 2.8 – Блок фільтрації

Борщ

Країна: Українська кухня

Тип страви: Перші страви

Складність приготування: Середній

Приготувати

Зберегти

Рисунок 2.9 – Відсортований рецепт

Наступним тестується можливість редагування рецепта на основі рецепта борщу. Необхідно увійти в адміністративну панель, обрати потрібний рецепт і внести зміни; після натискання кнопки "Редагувати" можна вносити

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

практично будь-які зміни в дані. Для прикладу змінити складність рецепта з “середньої” на “складну” (див. рис. 2.10)

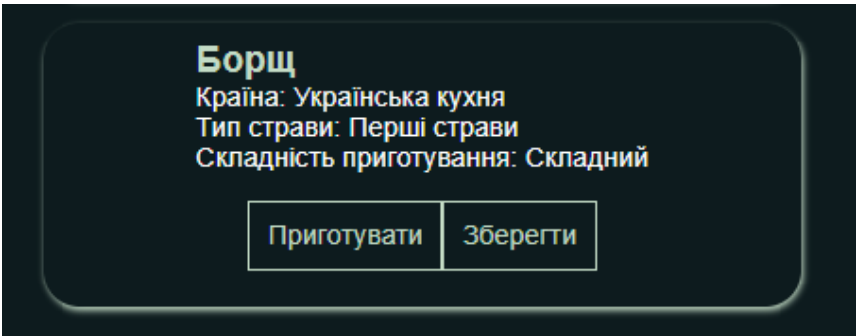


Рисунок 2.10 - Рецепт після редагування

Для видалення рецепта з'являється форма, з підтвердженням видалення (див. рис. 2.11)

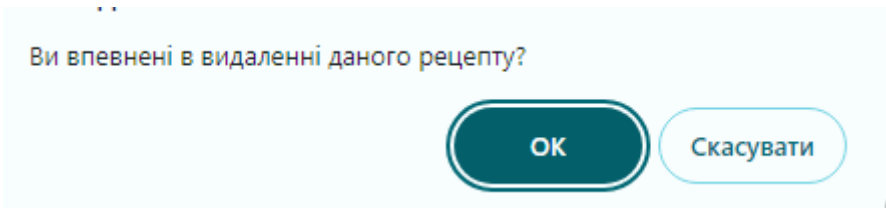


Рисунок 2.11 - Форма підтвердження видалення рецепта

Перевірка продовжується додаванням нового рецепту з такими даними (див. рис. 2.12):

- Суші з лососем
- Країна: японія
- Тип: холодні страви
- Складність: легка

Якщо залишити пусті поля введення виникне помилка створення (див. рис. 2.13)

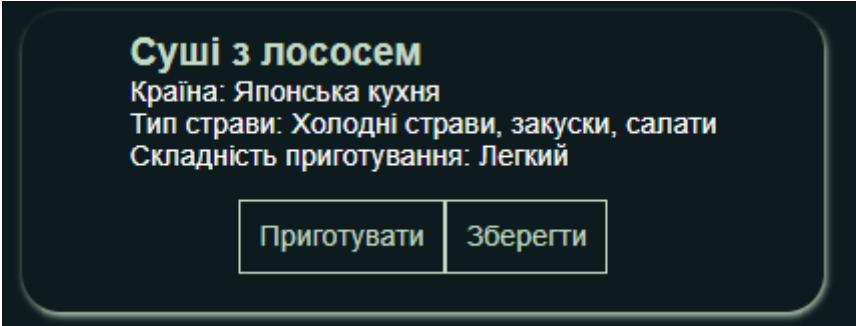


Рисунок 2.12 – Доданий елемент

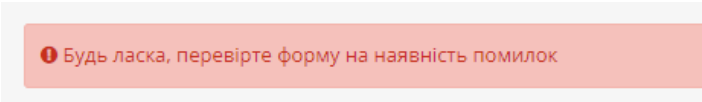


Рисунок 2.13 – Помилка створення нового рецепту

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розміщення сайту в Інтернеті

Розміщення сайту в Інтернеті включає кілька ключових кроків, щоб забезпечити, що він буде доступний для відвідувачів з усього світу. Нижче наведено загальну інструкцію, яка охоплює основні кроки з цього процесу:

Вибір надійного хостинг-провайдера є першим і важливим кроком у розміщенні сайту в Інтернеті. Хостинг-провайдер забезпечує простір для зберігання веб-сайту на своїх серверах і забезпечує доступ до мережі Інтернет. В якості хостинг-провайдера обрано офіційний веб ресурс ukraine.com.ua - "Хостинг Україна". Даний сервіс відомий своєю якісною та оперативною технічною підтримкою, що є важливим критерієм клієнтів і користувачів, що звертаються за допомогою. Даний ресурс пропонує можливість розміщення сайтів на серверах, що знаходяться в Україні. Це забезпечує кращу швидкість завантаження для відвідувачів з України. Компанія пропонує різні тарифні плани, що дозволяє вибрати оптимальний варіант з урахуванням потреб і бюджету.

Доменне ім'я - це унікальне ім'я, яке вказує на веб-сайт. Можна зареєструвати доменне ім'я через свого хостинг-провайдера або інші служби реєстрації доменів. На сайті ukraine.com.ua здійснюється пошук домену зі всіма доступними розширеннями (див. рис.3.1).

Після реєстрації доменного імені потрібно сконфігурувати DNS "Domain Name System" для вказівки доменного імені на IP-адресу хостинг-провайдера (див. рис. 3.2). Цей процес дозволяє браузерам знаходити веб-сайт за допомогою доменного імені.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

Реєстрація доменів

gourmetguru			🔍
			⬇ Сортувати за ціною 🔍 Приховати зайняті
gourmetguru.shop	Зайнятий		☰
gourmetguru.store	2 274.00 ₪ 114.00 ₪		Замовити
gourmetguru.online	Зайнятий		☰
gourmetguru.site	Зайнятий		☰
gourmetguru.tech	2 412.00 ₪ 138.00 ₪		Замовити
gourmetguru.fun	1 800.00 ₪ 78.00 ₪		Замовити
gourmetguru.space	1 212.00 ₪ 78.00 ₪		Замовити
gourmetguru.website	1 212.00 ₪ 72.00 ₪		Замовити

Рисунок 3.1 - Вибір доменного імені.



Рисунок 3.2 - Вибір DNS на сайті ukraine.com.ua

Після налаштування хостинг-акаунту і DNS можна завантажити файли веб-сайту на сервер хостинг-провайдера. Це включає HTML-сторінки, зображення, стилі CSS, скрипти JavaScript та інші необхідні ресурси (див. рис.3.3).

Перед публікацією веб-сайту в Інтернеті важливо провести тестування, щоб переконатися, що всі сторінки відображаються правильно, всі посилання працюють і функціонал сайту працює без помилок.

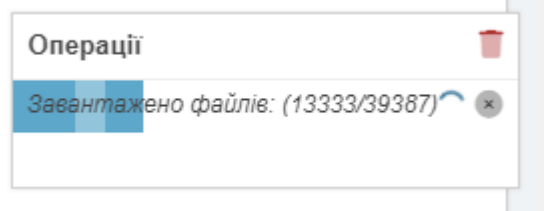


Рисунок 3.3 - Процес завантаження файлів веб-сайту на хостинг

Після успішного завантаження і тестування, веб-сайт готовий до публікації. Щоб переконатися, що сайт доступний публічно необхідно ввести доменне ім'я в поле пошуку браузера.

Після публікації веб-сайту важливо забезпечити його підтримку, включаючи регулярне оновлення контенту і заходи безпеки. Також можна встановити інструменти аналітики, такі як Google Analytics, для відстеження відвідуваності сайту.

3.2 Інструкція з обслуговування та наповнення сайту

Веб-сайт, присвячений вибору рецептів, повинен мати привабливий зовнішній вигляд та функції, які приваблюють відвідувачів та викликають інтерес. Щоб покращити взаємодію з користувачами, важливо систематично оновлювати дизайн вашого сайту. Оскільки гастрономічні тенденції змінюються, а кулінарні навички постійно вдосконалюються, ми регулярно оновлюємо вміст сайту, щоб відобразити нові кулінарні тенденції та кухні.

3.3 Інструкція з поширення та підтримки сайту

Вебсайт-це демонстрація бренду в величезному просторі Інтернету. В вебсайті “Кулінарний асистент” використовується перевірена стратегія просування, щоб він яскраво виділявся і привертав нових відвідувачів і клієнтів:

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

Пошукова оптимізація (SEO)

Необхідно переглядати сайт в пошукових системах, таких як Google.

Також необхідно використовувати відповідні ключові слова, мета-теги та регулювати швидкість завантаження сторінки.

Це допоможе потенційним клієнтам знайти сайт, коли вони шукають те, що він пропонує.

Соціальні мережі

Бути активним в соціальних мережах, де збирається цільова аудиторія, один з ключових аспектів успіху веб-сайтів.

Діліться цікавим контентом, спілкуватись з клієнтами, відповідати на коментарі і питання.

Створювати атмосферу залучення та побудувати міцні стосунки зі своєю аудиторією.

Контент-маркетинг

Створювати цікавий та корисний контент, який відповідає потребам та інтересам цільової аудиторії. Це можуть бути статті, відеоролики, Інфографіка, підкасти та інші формати.

Діліться знаннями і досвідом, щоб завоювати довіру і лояльність аудиторії.

Реклама в Інтернеті

Використовувати різні рекламні платформи, такі як Google Ads, Facebook ads та інші, щоб залучати більше глядачів.

Націлювання реклами на основі інтересів, поведінки та демографічних показників допоможе знайти потенційних клієнтів, які, швидше за все, зацікавляться пропозицією.

Взаємодієти зі спільнотою

Приєднуватись до тематичних форумів, груп соціальних медіа та інших онлайн-спільнот.

Діліться своїми знаннями, відповідати на питання, коментувати і вести конструктивний діалог.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

Це допомагає підвищити обізнаність про бренд, завоювати довіру в галузі та залучити нових клієнтів.

Аналітика та відстеження

Використання інструментів аналітики, такі як Google Analytics, для відстеження відвідуваності сайту, його джерел, поведінки користувачів та конверсій.

Проаналізувати свої дані, щоб зрозуміти, що працює, а що ні, і вдосконалити стратегію просування.

Це допоможе максимально підвищити віддачу від зусиль і домогтися найкращих результатів.

Використання різних стратегій, постійний аналіз результатів і вдосконалення методів дозволять вам з часом значно збільшити свою онлайн-аудиторію і досягти поставлених цілей.

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Економічна частина дипломного проекту "Розробка веб-додатку Кухонний помічник" присвячена ретельному аналізу економічної доцільності розробки даного веб-додатку. Основною метою є визначення прибутковості проекту, що дозволить прийняти обґрунтоване рішення щодо подальшої розробки проекту, його впровадження або відмови від розробки.

Об'єктом дослідження є онлайн-каталог магазину AquaPrime, для якого буде розроблений веб-додаток "Кухонний помічник".

Економічний аналіз буде проводитися в декілька етапів: деталізація процесу розробки, визначення трудомісткості кожного завдання (наприклад, час, необхідний для виконання завдання, кваліфікація персоналу тощо), розрахунок витрат на оплату праці (включаючи внески на соціальне страхування), визначення матеріальних витрат (включаючи складання списку необхідних матеріалів та обладнання), розрахунок витрат на електроенергію, визначення витрат на амортизацію (апаратного та програмного забезпечення), визначення непрямих витрат (загальних та адміністративних витрат, пов'язаних з розробкою), оцінка та визначення поточних витрат, розрахунок ціни роботи (з додаванням запланованого прибутку до робочих витрат), оцінка економічної ефективності та терміну окупності (з розрахунком чистого прибутку, індексу рентабельності та періоду окупності), аналіз результатів для визначення доцільності розробки веб-додатку.

Комплексний економічний аналіз дозволяє об'єктивно оцінити рентабельність проекту "Розробка веб-додатку Кухонний помічник", прийняти обґрунтоване рішення щодо подальшої розробки, впровадження або відмови від розробки, а також оптимізувати витрати на розробку та максимізувати прибуток від проекту.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю.

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції(стадії)	Виконавець	Середній час виконання операції, год.
1	Планування	Керівник проєкту	5
2	Розробка алгоритму вебзастосунку	Програміст	10
		Інженер	12
3	Написання коду вебзастосунку	Програміст	85
		Інженер	30
4	Тестування вебзастосунку	Тестувальник	10
5	Здача вебзастосунку	Програміст	5
		Керівник проєкту	5
Р а з о м			162

Сумарний час виконання операцій технологічного процесу становить 162 години.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки веб-застосунку «Кулінарний Асистент»

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

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c \cdot K_r \quad (4.1)$$

де: T_c – тарифна ставка, грн. (приймаємо для керівника – 123 грн./год, програміста – 112 грн./год., інженера – 89 грн./год., тестувальника – 84 грн./год.);

K_r – кількість відпрацьованих годин.

Отже основна заробітна плата для:

Керівника проекту $Z_{\text{осн}2} = 10 \cdot 123,00 = 1\,230,0$ грн.

Програміста $Z_{\text{осн}3} = 100 \cdot 112,00 = 11\,200,0$ грн.

Інженера $Z_{\text{осн}4} = 42 \cdot 89,00 = 3\,738,0$ грн.

Тестувальника $Z_{\text{осн}5} = 10 \cdot 84,00 = 840,0$ грн.

Сумарна основна заробітна плата становить

$Z_{\text{осн}} = 1\,230,00 + 11\,200,00 + 3\,738,00 + 840,00 = 17\,008,0$ грн.

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} \cdot K_{\text{допл.}} \quad (4.2)$$

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

Керівника проекту $З_{\text{дод}2} = 1\,230,00 * 0,15 = 184,4$ грн.
 програміста $З_{\text{дод}3} = 11\,200,00 * 0,15 = 1\,680,0$ грн.
 Інженера $З_{\text{дод}4} = 3\,738,00 * 0,15 = 560,0$ грн.
 Тестувальника $З_{\text{дод}5} = 840,00 * 0,15 = 126,0$ грн.

Загальна додаткова заробітна плата становить:

$З_{\text{дод}} = 184,4 + 1\,680,00 + 560,00 + 126,00 = 2\,550,40$ грн.

Звідси загальні витрати на оплату праці ($B_{o.n.}$) визначаються за формулою:

$$B_{o.n.} = Z_{o.c.h.} + Z_{\text{дод.}} \quad (4.3)$$

$B_{o.n.} = 17\,008,00 + 2\,550,4 = 19\,558,40$ грн.

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$B_{\text{ЄСВ}} = B_{o.n.} * 0,22 \quad (4.4)$$

$B_{\text{ЄСВ}} = 19\,558,40 * 0,22 = 4\,302,85$ грн.

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівни- ків	Основна заробітна плата, грн.			Додатко- ва заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. $6=3+4+5$
		Тариф- на ставка, грн.	К-сть від- працьов. год.	Фактично нарах. зарплати, грн.			
<i>A</i>	<i>B</i>	1	2	3	4	5	6
1	Кер.проекту	123	10	1 230	184,40	-	-
2	Програміст	112	100	11 200	1 680	-	-
3	Лаборант	89	42	3 738	560	-	-
4	Тестувальн ик	84	10	840	126		
Разом				17 008,0	2 550,40	4302,8 5	23 863,25

Отже, загальні витрати на оплату праці становлять 23 863,25грн.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

4.3 Розрахунок матеріальних витрат

Матеріальні витрати визначаються як добуток кількості витрачених матеріалів та їх ціни:

$$M_{vi} = q_i * p_i, \\ (4.5)$$

де: q_i – кількість витраченого матеріалу i -го виду; p_i – ціна матеріалу i -го виду.

Звідси, загальні матеріальні витрати можна визначити:

$$Z_{м.в.} = \sum M_{vi} \\ (4.6)$$

Проведені розрахунки занесемо у таблицю 4.3.

Таблиця 4.3 - Зведені розрахунки матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна 1-ці, грн.	Загальна сума витрат, грн.
1	CD диски	шт.	4	15,0	60,0
2	Друк	листів	100	3	300,0
3	Друк ватманів	листів	4	30,0	120,0
Р а з о м					480,0

Отже, загальна сума матеріальних витрат на розробку програми становить 480,0 грн

4.4 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_v = W * T * S \\ (4.7)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання; S – вартість кіловат-години електроенергії (приймаємо 7 грн).

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: ноутбук – 0,23 кВт/год.

$$З_{ек} = 0,23 * 162 * 7 = 261 \text{ грн.}$$

Витрати на електроенергію становлять 261 грн.

4.5 Розрахунок суми амортизаційних відрахувань вебсайту «Кулінарний асистент»

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{БВ * Н_A}{150\%} * T, \quad (4.8)$$

де: А – амортизаційні відрахування за звітний період, грн.;

БВ – балансова вартість групи основних фондів на початок звітного періоду, грн.;

Н_А – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ноутбук, вартістю 25900,00 грн., то сума амортизаційних відрахувань становитиме:

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

$$A = \frac{25\,900 * 0,04}{150\%} * 162 = 1119 \text{ грн.}$$

4.6 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = B_{o.п.} * 0,2..0,6$$

(4.9)

де: H_B – накладні витрати.

$$H_B = 19558,4 * 0,4 = 7\,823,36 \text{ грн.}$$

4.7 Складання кошторису витрат та визначення собівартості веб-застосунку «Кулінарний Асистент»

Для складання кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат веб-застосунку «Кулінарний Асистент»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці (основну і додаткову заробітну плату)	19 558,4	58,6
2.	Єдиний соціальний внесок	4 302,85	13
3.	Матеріальні витрати	480,00	1,4
4.	Витрати на електроенергію	261	0,7
5.	Амортизаційні відрахування	1119	3,3
6.	Накладні витрати	7823,36	23
7.	Собівартість	33 540,61	100

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

Собівартість (C_B) НДР розраховуємо за формулою:

$$C_B = B_{o.п} + B_{c.з} + 3M + 3e + A + H_B \quad (4.10)$$

Отже, собівартість дорівнює $C_B = 33\,540,61$ грн.

4.8 Розрахунок ціни робіт

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

$$Ц = C_B * (1 + P_{рен}) * (1 + ПДВ), \quad (4.11)$$

де: C_B – собівартість; $P_{рен}$ – рівень рентабельності, 30 %; ПДВ – ставка податку на додану вартість, 20 %

$$Ц = 33\,540,61 * (1 + 0,3) * (1 + 0,2) = 52\,322,4 \text{ грн.}$$

4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Розрахунок прибутку визначається з а формулою:

$$П = Ц - C_B \quad (4.12)$$

де: $Ц$ – ціна розробки, грн.; C_B – собівартість розробки, грн.

$$П = 52\,322,4 - 33\,540,61 = 18\,781,79 \text{ грн.}$$

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$ЧТВ = -C_B + \sum_{i=1}^t \frac{\Gamma_{П}}{(1+i)^t}, \quad (4.13)$$

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

де: C_v – собівартість розробки; Γ_{π} – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$\begin{aligned} \text{ЧТВ} &= -33\,540,61 + (18\,781,79/(1+0,1)^1) + (18\,781,79/(1+0,1)^2) + \\ &(18\,781,79/(1+0,1)^3) = -33\,540,61 + 17\,074,35 + 15\,522,14 + 14\,111,03 = \\ &= 13\,166,91 \end{aligned}$$

Якщо $\text{ЧТВ} \geq 0$, то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{\text{OK}} = T_{\text{ПВ}} + \frac{H_B}{\Gamma_{\text{ПР}}} \quad (4.14)$$

де: $T_{\text{ПВ}}$ – період до повного відшкодування витрат, років; H_B – невідшкодовані витрати на початок року, грн.; $\Gamma_{\text{ПР}}$ – грошовий потік на початок року, грн.

$$H_B = (18\,781,79/(1+0,1)^3) - 13\,166,91 = 993,53$$

$$T_{\text{OK}} = 2,78 + (993,53 / 18\,781,79) = 2,78 + 0,05 = 2,83$$

Всі дані внесемо в зведену таблицю 4.5.

Таблиця 4.5 – Техніко-економічні показники веб-застосунку «Кулінарний Асистент»

№ п/п	Показник	Значення
1.	Собівартість, грн.	33 540,61
2.	Плановий прибуток або грошовий потік, грн.	18 781,79
3.	Ціна, грн.	52 322,4
4.	Чиста теперішня вартість, грн.	13 166,91
5.	Термін окупності, рік	2,83

Загальна вартість розробеного програмного продукту становить 52 322,40

гривні. Чистої теперішньої вартості ми досягаємо на 13 166,91рік.

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2,83 року. Отже, на основі отриманих показників можна зробити висновок, що розробка веб-застосунок «Кулінарний асистент» є доцільною з економічної точки зору.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1. Заходи щодо забезпеченню безпечної роботи при ремонті технологічного обладнання.

Безпека під час ремонту технологічного обладнання є одним з найважливіших аспектів виробничої діяльності. Недотримання правил безпеки може призвести до серйозних травм, нещасних випадків, пожеж та інших аварій. Тому важливо ретельно готувати до ремонту, чітко виконувати всі інструкції та використовувати необхідні засоби індивідуального та колективного захисту.

Загальні заходи безпеки

Перед початком ремонту необхідно:

- Отримати наряд-дозвіл на ремонт.
- Ознайомитися з інструкцією з експлуатації та ремонту обладнання.
- Відключити обладнання від електромережі, газ, води та інших джерел енергії.
- Провести огляд обладнання та виявити можливі дефекти.
- Підготувати необхідні інструменти, пристосування та матеріали.
- Забезпечити місце ремонту необхідними засобами колективного захисту (огорожі, знаки безпеки, освітлення).
- Провести інструктаж з безпеки з персоналом, який буде задіяний у ремонті.

Заходи безпеки при ремонті різних видів обладнання

При ремонті різних видів обладнання необхідно дотримуватися specific safety measures, які враховують особливості його конструкції та експлуатації.

Наприклад:

При ремонті електричного обладнання:

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

- Працювати в гумових рукавичках та діелектричних калошах.
- Використовувати інструмент з ізольованими рукоятками.
- Не підключати обладнання до електромережі до повного закінчення ремонту.

При ремонті механічного обладнання:

Використовувати захисні окуляри та щитки.

Не працювати з рухомими частинами обладнання.

Не перевантажувати обладнання.

При ремонті газового обладнання:

Працювати в добре провітрюваному приміщенні.

Використовувати газовий респіратор.

Не допускати витоків газу.

При ремонті посудин, що знаходяться під тиском:

Перед початком ремонту зняти тиск з посудини.

Не працювати з пошкодженими посудинами.

Використовувати захисні окуляри та щитки.

Засоби індивідуального та колективного захисту

Для забезпечення безпеки при ремонті технологічного обладнання необхідно використовувати такі засоби індивідуального та колективного захисту:

Засоби індивідуального захисту:

- Спецодяг (комбінезони, костюми)
- Захисні окуляри та щитки
- Респіратори
- Рукавички
- Каски
- Вушні вкладиші

Засоби колективного захисту:

- Огорожі

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

- Знаки безпеки
- Заземлення
- Блокувальні пристрої
- Системи вентиляції

Відповідальність за безпеку при ремонті

Відповідальність за безпеку при ремонті технологічного обладнання несе:

- Власник або керівник підприємства.
- Особа, відповідальна за безпеку при ремонті.
- Працівники, які виконують ремонт.

Навчання та інструктаж

Всі працівники, які задіяні в ремонті технологічного обладнання, повинні пройти навчання та інструктаж з безпеки.

Дотримання правил безпеки при ремонті технологічного обладнання є запорукою запобігання травматизму, нещасних випадків та аварій. Тому важливо, щоб всі працівники, які задіяні в ремонті, чітко знали та виконували ці правила.

5.2. Фізичні основи електробезпеки.

Електробезпека є одним із найважливіших аспектів роботи з електричними системами. Розуміння фізичних принципів, які лежать в основі електробезпеки, є ключовим для запобігання травмам та нещасним випадкам.

Електричний струм та його вплив на організм людини

Електричний струм - це потік електричних зарядів через провідник. Він може мати різний вплив на організм людини, залежно від його величини, тривалості дії та шляху проходження через тіло.

Основні фактори, що визначають вплив електричного струму на людину:

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

- Сила струму: Чим більша сила струму, тим більший його вплив на організм.
- Час дії: Чим довше електричний струм проходить через тіло людини, тим більший його вплив.
- Шлях проходження: Шлях проходження електричного струму через тіло людини визначає ступінь його небезпеки. Найбільш небезпечним є проходження струму через голову, грудну клітку та спину.

Ефекти електричного струму на організм людини:

- Тепловий ефект: Електричний струм може призвести до опіків.
- Електрохімічний ефект: Електричний струм може викликати електролітичні розлади в організмі.
- Біологічний ефект: Електричний струм може призвести до порушення роботи м'язів, нервів та серця.

Електричний опір та його роль в електробезпеці

Електричний опір - це міра здатності матеріалу протистояти проходженню електричного струму. Чим більший електричний опір матеріалу, тим менший струм буде через нього протікати.

Роль електричного опору в електробезпеці:

Заземлення: Заземлення - це з'єднання електричного обладнання з землею за допомогою провідника. Заземлення використовується для відведення струму витoku на землю, що знижує ризик ураження людини.

Захисні заходи: Використання захисних заходів, таких як ізоляція, заземлення та диференціальні автомати, допомагає зменшити ризик ураження людини електричним струмом.

Діелектрики та їх роль в електробезпеці

Діелектрики - це матеріали з високим електричним опором, які не проводять електричний струм. Вони використовуються для ізоляції провідників, кабелів та електричного обладнання.

Роль діелектриків в електробезпеці:

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

Ізоляція: Діелектрики використовуються для ізоляції провідників, щоб запобігти контакту людини з оголеними струмоведучими частинами.

Захисні засоби: Використання захисних засобів, таких як гумові рукавички та калоші, допомагає захистити людину від ураження електричним струмом.

Розуміння фізичних принципів, які лежать в основі електробезпеки, є ключовим для запобігання травмам та нещасним випадкам. Використання захисних заходів, таких як ізоляція, заземлення та диференціальні автомати, допомагає зменшити ризик ураження людини електричним струмом.

ВИСНОВКИ

В цій дипломній роботі було описано розробку веб-сайту по підбору кулінарних рецептів “Куліарний асистент”. Сервіс який впроваджений на основі RESTful API серверній частині та до якої впроваджений інтерактивний web-інтерфейс. Серверна частина обробляє http запити по заданих end-point, видає оброблену інформацію яку черпає з бази даних або з замого http запиту. Виконана програма задовільнила всі поставлені вимоги технічного завдання.

Для написання дипломного проєкту було використано back-end фреймворки Spring Boot, SPRING WEB, SPRING DATA JPA, SPRING SECURITY, SPRING на базі мови програмування Java, що зробило розробку серверної частини застосунку набагато лекшим, надійнішим та дозволило розробити всю взаємодію згідно принципам REST архітектури.

Для розробки front-end частини використано бібліотеку Bootstrap для верстки макету а також мови розмітки HTML та CSS, за основу клієнтського інтерфейсу було обрано мову програмування JAVASCRIPT та фреймворки до нього REACT, REACT-ROUTER-DOM, AXIOS, JWTDECODE.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Стоян Стефанов. React: Up & Running: Бостон : Вид-во “O’Reilly Media”, 2021.233с.
- 2) Swagger: веб-сайт. URL: <https://swagger.io/> (дата звернення: 2.06.2024).
- 3) Хостинг Україна: веб-сайт. URL: <https://www.ukraine.com.ua/uk/> (дата звернення: 13.06.2024).
- 4) Документація React : веб-сайт. URL: <https://uk.legacy.reactjs.org/docs/getting-started.html> (дата звернення: 29.05.2024)
- 5) ChatGpt : веб-сайт. URL: <https://chatgpt.com/> (дата звернення: 5.06.2024)
- 6) Google Search Console. URL: <https://search.google.com/search-console/about/> (дата звернення: 7.06.2024)
- 7) PostgreSQL. URL: <https://www.postgresql.org/> (дата звернення: 23.05.2024)
- 8) Swagger: веб-сайт. URL: <https://swagger.io/> (дата звернення: 26.05.2024).
- 9) Олексій Васильєв. Програмування мовою Java : Вид-во “Навчальна книга - Богдан”, 2020.696с.
- 10) Документація Axios: веб-сайт. URL: <https://axios-http.com/uk/docs/intro> (дата звернення: 4.06.2024)
- 11) Документація React-Router: веб-сайт. URL: <https://reactrouter.com/en/main> (дата звернення: 7.06.2024)
- 12) Володимир Пасічник. Веб-технології: Вид-во “Магнолія”, 2024.336с.

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А – лістинг файлу “APP.js”

```
import './App.css';

import Home from './pages/Home';

import About from './pages/About';

import Login from './pages/Login'

import Register from './pages/Register'

import Recipe from './pages/Recipe'

import { Route, Routes } from 'react-router-dom';

import React,{useState,useContext} from 'react'

import Recipes from './pages/Recipes';

import UserAccount from './pages/userAccount'

function App() {

return (

  <><Routes>

  <Route path="/recipe" element={ <Recipe/> } />

  <Route path="/" element={ <Home/> } />

  <Route path="/Recipes" element={ <Recipes/> } />

  <Route path="/Register" element={ <Register/> } />

    <Route path="/Login" element={ <Login/> } />

    <Route path="/UserAccount" element={ <UserAccount/> } />

    <Route path="/About" element={ <About/> } />

  </>

);

}
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

</Routes>

</>);

}

export default App;

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Б – лістинг файлу “Home.jsx”

```
import React, { useContext } from 'react';

import Header from '../components/Header';

import RecipeList from '../components/recipeList';

import { AuthContext } from '../api/AuthContext';

function Home() {

  const { isAuthorised } = useContext(AuthContext);

  return (

    <>

      <Header isAuthorised={isAuthorised} />

      {isAuthorised ? (

        <>

          <div className="body-logo">

            <h2 className="body-logo-text">Best Recipes</h2>

          </div> </> ) : null}

      <RecipeList route="http://localhost:8080/api/dishes" />

    </> );

}

export default Home;
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток В – лістинг файлу “RecipeItem.jsx”

```
import React, { useState, useContext } from 'react';

import { AuthContext } from '../api/AuthContext';

import DataProvider from '../api/dataProvider';

import { useNavigate, useLocation } from 'react-router-dom';

import { checkUserRole } from "../api/authorised";

function RecipeItem({ recipe }) {

  const navigate = useNavigate();

  const location = useLocation();

  const { isAuthorised } = useContext(AuthContext);

  const { name, type, difficulty, cuisine } = recipe;

  const countryName = cuisine.name;

  const [selectedItemId, setSelectedItemId] = useState();

  const handleClickCook = async () => {

    const newSelectedItemId = recipe.id;

    setSelectedItemId(prevSelectedItemId => {

      if (prevSelectedItemId === newSelectedItemId) {

        return null;

      }

      return newSelectedItemId;

    });

    const route = `http://localhost:8080/api/dishes`;

    const token = localStorage.getItem('userToken');
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		


```

try {
  if (isAuthorised) {
    const data = await DataProvider.getOne(newSelectedItemId, route,
token);

    console.log(data);

    navigate('/Recipe', { state: { recipeData: data } });
  }
} catch (error) {
  console.error('Error fetching recipe:', error);
}
};

const handleClickDelete = async () => {
  const newSelectedItemId = recipe.id;

  setSelectedItemId(prevSelectedItemId => {
    if (prevSelectedItemId === newSelectedItemId) {
      return null;
    }

    return newSelectedItemId;
  });

  // Використовуємо новий Promise для підтвердження видалення
  const confirmDelete = new Promise((resolve, reject) => {
    if (window.confirm("Ви впевнені в видаленні даного рецепту?")) {
      resolve(true); // користувач підтвердив видалення
    } else {

```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        reject(new Error('Видалення скасовано')); // користувач скасував
        видалення
    }
});

confirmDelete.then(async () => {

    const route = `http://localhost:8080/api/dishes`;

    const token = localStorage.getItem('userToken');

    try {

        if (isAuthorised) {

            const data = await DataProvider.deleteOne(newSelectedItemId, route,
token);

            console.log(data);

            navigate('/'); }

        } catch (error) {

            console.error('Error delete recipe:', error);

        }

    }).catch((error) => {

        console.log(error.message);    }); });

const      isAdmin      =      isAuthorised      &&
checkUserRole(localStorage.getItem('user_role'));

const isUserAccountPage = location.pathname === '/UserAccount';

return (

    <div className="recipe-item">

        <div className="card-bottom">

```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    <h3 className="card-name">{ name }</h3>

    <div className="card-description">

      <ul>

        <li>Країна: { countryName }</li>

        <li>Тип страви: { type }</li>

        <li>Складність приготування: { difficulty }</li>

      </ul></div>

    { isUserAccountPage && isAdmin && (

      <div className="btn-block">

        <button className="card-btn">Редагувати</button>

        <button
                                className="card-btn"
onClick={ handleClickDelete }>Видалити</button>

        </div>)}

    { !isUserAccountPage && (

      <div className="btn-block">

        <button
                                className="card-btn"
onClick={ handleClickCook }>Приготувати</button>

        <button className="card-btn">Зберегти</button></div>

      )}

    </div>

  </div>

);

}

export default RecipeItem;

```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Г – лістинг файлу “AuthContext.jsx”

```
import React, { createContext, useState, useEffect } from 'react';

export const AuthContext = createContext({
  isAuthorised: "",
  setToken: () => {},
  deleteToken: () => {}
});

export const AuthProvider = ({ children }) => {
  const [isAuthorised, setIsAuthorised] = useState(null);

  useEffect(() => {
    const token = localStorage.getItem('token');

    if (token) {
      setIsAuthorised(token);
    }
  }, []);

  const setToken = (token) => {
    localStorage.setItem('token', token);
    setIsAuthorised(token);
  };

  const deleteToken = () => {
    localStorage.removeItem('token');
    localStorage.removeItem('user_role');
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

```
localStorage.removeItem('user_sub');

localStorage.removeItem('user_email');

setIsAuthorised(null);

};

return (

  <AuthContext.Provider

    value={{ { isAuthorised, setToken, deleteToken } }}

  >

    {children}

  </AuthContext.Provider>

);

};
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Е – лістинг файлу “authorised.js”

```
import React, { useState, useEffect } from 'react';

export function checkUserRole(role) {

  try {

    if (!role) {

      throw new Error('Користувач не знайдений');

    }

    if (role === 'ROLE_ADMIN') {

      return true;

    }

    return false;

  }

  catch (error) {

    console.error(error.message);

    return false;

  }

}

export function deleteToken() {

  try {

    localStorage.removeItem('userToken');

    console.log('Токен успішно видалено');
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						66
Зм.	Арк.	№ докум.	Підпис	Дата		

```
}  
  
catch (error) {  
  
    console.error('Помилка видалення токена:', error);  
  
}  
  
}
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Є – лістинг файлу “UserAccount.jsx”

```
import React, { useEffect, useState, useContext } from 'react';

import { checkUserRole } from "../api/authorised";

import { AuthContext } from '../api/AuthContext';

import Header from "../components/Header";

import UserAccInfo from './userAccPages/userAccInfo';

import UserFavoriteDishes from './userAccPages/UserFavoriteDishes';

import EditRecipes from './userAccPages/EditRecipes';

import UpdateItem from './userAccPages/UpdateItem';

function UserAccount() {

  const { isAuthorised } = useContext(AuthContext);

  const storedRole = localStorage.getItem('user_role');

  const storedSub = localStorage.getItem('user_sub');

  const storedEmail = localStorage.getItem('user_email');

  const [userRole, setUserRole] = useState(null);

  const [selectedPage, setSelectedPage] = useState(null);

  useEffect(() => {

    if (isAuthorised) {

      const role = checkUserRole(storedRole);

      setUserRole(role);

    }

  });

}
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		


```

    }

    }, [isAuthorised, storedRole]);

    const handleNavigateToUserAccInfo = () => {

        setSelectedPage('userAccInfo');

    };

    const handleNavigateToFavouriteDishes = () => {

        setSelectedPage('userFavouriteDishes');

    };

    const handleNavigateToEditRecipes = () => {

        setSelectedPage('editRecipes');

    };

    return (

        <>

        <Header isAuthorised={isAuthorised} />

        <div className="main-content-top">

            <ul>

                <li><a          onClick={handleNavigateToFavouriteDishes }><h4>Збережені
рецепти</h4></a></li>

                {userRole && (

                    <>

                        <li><a          onClick={handleNavigateToEditRecipes }><h4>Редагувати
рецепти</h4></a></li><li><a><h4>Добавити рецепти</h4></a></li></>

                    )}

            </ul>

```

<h4>Персональна інформація</h4>

</div>

{selectedPage === 'userAccInfo' && <UserAccInfo role={storedRole} email={storedEmail} sub={storedSub} />}

{selectedPage === 'userFavouriteDishes' && <UserFavoriteDishes />}

{selectedPage === 'editRecipes' && <EditRecipes />}

{selectedPage === 'UpdateItem' && <UpdateItem />}

</>

);}

export default UserAccount;

Додаток Ж – лістинг файлу “DishController.java”

```
Import io.swagger.v3.oas.annotations.tags.Tag;
import org.springframework.http.ResponseEntity;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.web.bind.annotation.*;
import ua.te.gourmetguru.dto.DishDto;
import ua.te.gourmetguru.service.DishService;

import java.util.List;

@RestController
@RequestMapping("/api/dishes")
@Tag(name = "Страви")
public class DishController {
    private final DishService dishService;

    public DishController(DishService dishService) {
        this.dishService = dishService;
    }

    @PostMapping("/recommend")
    public ResponseEntity<List<DishDto>> recommendDishes(@RequestBody
List<String> ingredientNames,
                                                         @RequestParam(name = "cuisine", required =
false) String cuisineName) {
        List<DishDto> recommendedDishes =
dishService.recommendDishes(ingredientNames, cuisineName);
        return ResponseEntity.ok(recommendedDishes);
    }
}
```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

}

@GetMapping

```
public List<DishDto> getAllDishes() {  
    return dishService.getAllDishes();  
}
```

@GetMapping("/Types/")

```
public List<DishDto> getDishesByType() {  
    return null; //Todo  
}
```

```
public List<DishDto> getDishesByDifficulty() {  
    return null; //Todo  
}
```

@GetMapping("/{id}")

```
public ResponseEntity<DishDto> getDishById(@PathVariable Long id) {  
    DishDto dishDto = dishService.getDishById(id);  
    return dishDto != null ? ResponseEntity.ok(dishDto) :  
    ResponseEntity.notFound().build();  
}
```

@PostMapping

@PreAuthorize("hasRole('ADMIN')")

```
public DishDto createDish(@RequestBody DishDto dishDto) {  
    return dishService.createDish(dishDto);  
}
```

@PutMapping("/{id}")

@PreAuthorize("hasRole('ADMIN')")

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    public ResponseEntity<DishDto> updateDish(@PathVariable Long id,
@RequestBody DishDto dishDetails) {
        DishDto updatedDish = dishService.updateDish(id, dishDetails);
        return updatedDish != null ? ResponseEntity.ok(updatedDish) :
ResponseEntity.notFound().build();
    }

    @DeleteMapping("/{id}")
    @PreAuthorize("hasRole('ADMIN')")
    public ResponseEntity<Void> deleteDish(@PathVariable Long id) {
        dishService.deleteDish(id);
        return ResponseEntity.noContent().build();
    }
}

```

					2024.ДП.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73