НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Изображение выглядит как текст, Шрифт, снимок экрана, Цвет электрик

Автоматически созданное описание

Шифр \_\_\_\_\_\_\_\_\_\_ ЗАТВЕРДЖУЮ

(необхідно вказати № інд.навч.плану) Зав. кафедри ПЗКС

д.т.н., проф.\_\_\_\_\_М.О.Алексєєв

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2024 р.

**КУРСОВА РОБОТА**

з дисципліні “Web-технології та web-дизайн”

на тему: Розробка веборієнтованої інформаційної системи …

(необхідно вказати назву сайту)

Пояснювальна записка

Студента (ки) 1 курсу, групи 121-23ск-2

Спеціальності 121

Інженерія програмного забезпечення

Новоженіна Ігора Євгеновича

(прізвище та ініціали студента (ки))

Керівник: к.т.н., доцент В.В.Спірінцев

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Університетська шкала: \_\_\_\_\_\_\_\_\_\_\_

Оцінка ECTS: \_\_\_\_

Члени комісії: \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

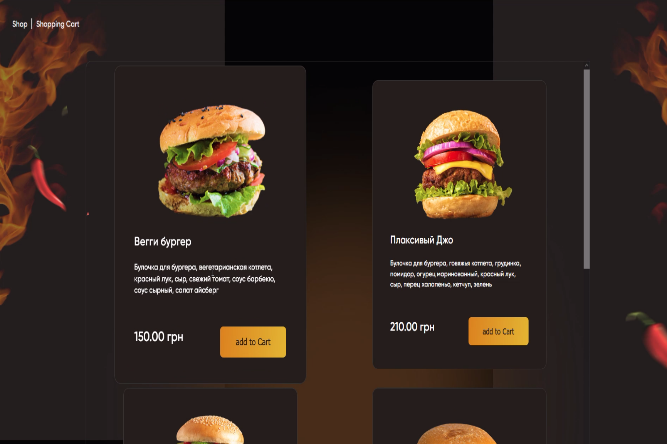
(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

Дніпро

2024

Завдання на курсову роботу. Варіант № 14

**ВСТУП**

Найменування:«Веборієнтована інформаційна система для онлайн заказу бургерів». Ця система допомагає користувачам замовити доставку бургерів онлайн.

**ПІДСТАВА ДЛЯ РОЗРОБКИ**

Підставою для розробки є тематика курсової роботи, що затверджена на засіданні кафедри ПЗКС.

**ПРИЗНАЧЕННЯ РОЗРОБКИ**

Курсова робота призначена для застосування практичних навиків з розробки і менеджменту інформаційних систем, на прикладі проекту розробки інформаційної системи онлайн заказу бургерів. Проект повинен мати… «»

**ВИМОГИ ДО ПРОГРАМИ**

Макет інформаційної системи повинен бути зверстаний згідно заданого дизайну з використанням правил CSS і HTML, і перевірений на коректність сервісом www.w3.org.

Головна сторінка повинна містити список продуктів з інформацією про них, можливістю додати обрані продукти у кошик, після чого має відбуватися перехід на сторінку підтвердження заказу.

На сторінці кошику має знаходитися інформація про обрані користувачем продукти, можливість збільшити або зменшити кількість обраного товару, кнопка підтвердження покупки має бути неактивною, допоки користувач не пройде валідацію. При зменшенні/збільшенні кількості товару, на сторінці кошику має обновлюватися загальна ціна обраних продуктів.

# ЗМІСТ

|  |  |  |
| --- | --- | --- |
| ЗАВДАННЯ НА КУРСОВУ РОБОТУ ………………………………….... | | 2 |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ………………………….…………. | | 5 |
| ВСТУП…………………………………………………………………….... | | 6 |
| РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ………………………...…. | | 7 |
| 1.1. | Загальні відомості з предметної галузі…………………………… | 7 |
| 1.2. | Призначення розробки та галузь застосування…………….….…. | 9 |
| 1.3. | Постановка завдання………………………………………………. | 9 |
| 1.4. | Вимоги до інформаційної системи…………..…………………… | 10 |
| РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ………………………………………………………………… | | 11 |
| 2.1. | Опис використаної архітектури та шаблонів проектування…….. | 11 |
| 2.2. | Опис використаних технологій та мов програмування…………. | 12 |
|  | HTML ………………………..…………………………………….. | 13 |
|  | CSS …….…………………………………………………………… | 15 |
|  |  |  |
|  | JavaScript …………………………….……………………………..  PHP …………………………………………………………………. | 18  22 |
| ВИСНОВКИ…………………………………………………..…………….. | | 25 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ………………………….………. | | 63 |
| Додаток А. Код програми………………………………………………..… | | 65 |

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

|  |  |
| --- | --- |
| HTML – | Стандартизована мова розмітки гіпертексту документа для перегляду вебсторінок у браузері; |
| CSS – | Це мова ієрархічних правил (таблиць стилів), що використовується для представлення зовнішнього вигляду документа, написаного на HTML або XML; |
| БД – | База даних; |
| SQL – | Декларативна мова програмування, яку застосовують для створення, модифікації та управління даними в реляційній базі даних; |
| PHP – | Скриптова мова загального призначення, яку інтенсивно застосовують для розробки вебдодатків; |
| JS – | Мультипарадигменна мова програмування, підтримує об'єктно-орієнтований, імперативний та функціональний стилі. |
| SSD – | комп'ютерний енергонезалежний немеханічний запам'ятовувальний пристрій на основі мікросхем пам'яті, альтернатива жорстким дискам (HDD); |
| СУБД – | комплекс програм, що дають змогу створити базу даних і маніпулювати даними (вставляти, оновлювати, видаляти і вибирати); |
| ПЗ – | Програмне забезпечення; |
| WEB –  NoSQL – | Інтернет-простір;  Позначення широкого класу різнорідних систем управління базами даних |

**ВСТУП**

Сучасний світ характеризується стрімким розвитком інформаційних технологій, які проникають у всі сфери діяльності, зокрема і в сферу онлайн торгівлі. Ефективне функціонування сайтів для замовлення їжі неможливе без використання спеціалізованих веборієнтованих систем, що допомагають автоматизувати робочі процеси та підвищити якість обслуговування клієнтів. Тому розробка такої системи є актуальним завданням.

Метою даної курсової роботи є створення навчального проекту веборієнтованої інформаційної системи для онлайн замовлення бургерів, що дозволить оптимізувати процеси замовлення, обробки та доставки їжі, а також покращити взаємодію з клієнтами.

Для досягнення поставленої мети необхідно виконати наступні завдання:

1.Дослідити особливості функціонування сайтів для замовлення їжі та визначити їхні потреби у веборієнтованих системах.

2.Проаналізувати існуючі аналоги та визначити їхні переваги і недоліки.

3.Спроектувати архітектуру системи та визначити необхідний функціонал.

4.Розробити вебдодаток із використанням сучасних технологій та підходів.

5.Протестувати роботу системи та підготувати інструкції користувача.

Об'єктом дослідження є процес функціонування сайтів для замовлення їжі та їх взаємодія з веборієнтованими системами управління замовленнями.

Предметом дослідження виступає веборієнтована інформаційна система для онлайн замовлення бургерів.

Практичне значення полягає в можливості використання розробленої системи в реальних закладах харчування для підвищення ефективності роботи з замовленнями та покращення якості обслуговування клієнтів.

В ході роботи був проведений аналіз предметної області, спроектована архітектура системи і реалізовано вебдодаток із застосуванням HTML, CSS, JavaScript та PHP.

**1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ**

**1.1. Загальні відомості з предметної галузі**

Тематика бургерів є однією з найбільш актуальних і широко обговорюваних у сфері громадського харчування. Захоплення бургерами є поширеним серед різних верств суспільства, інтерес до нових смаків, рецептів та технік приготування постійно зростає. Завдяки постійному впровадженню нових технологій та інновацій у галузі кулінарії, інформаційна підтримка цієї сфери є важливою. Світ бургерів насичений новинами та інформацією про останні події, рецепти, нові тренди та заклади. Існує багато кулінарних медіа, які покривають цю сферу, включаючи журнали, блоги, відеоканали та соціальні мережі.

На сьогоднішній день інформаційні системи для онлайн замовлення їжі вже широко використовуються в Інтернеті. Однак, тематика бургерів є популярною, і існує попит на створення спеціалізованих платформ для замовлення та доставки бургерів. Основні проблеми, які виникають при розробці інформаційних систем такого типу, полягають у забезпеченні швидкого та зручного доступу до меню, високоякісного візуального представлення страв, а також забезпечення можливостей взаємодії з користувачами через відгуки та рейтинги.

Проведений критичний аналіз існуючих магазинів онлайн замовлення бургерів показав, що багато з наявних платформ не відповідають сучасним вимогам щодо зручності використання.

Один з проаналізованих сайтів це Kanapulka. На рисунку 1.1 зображено головну сторінку сайту.

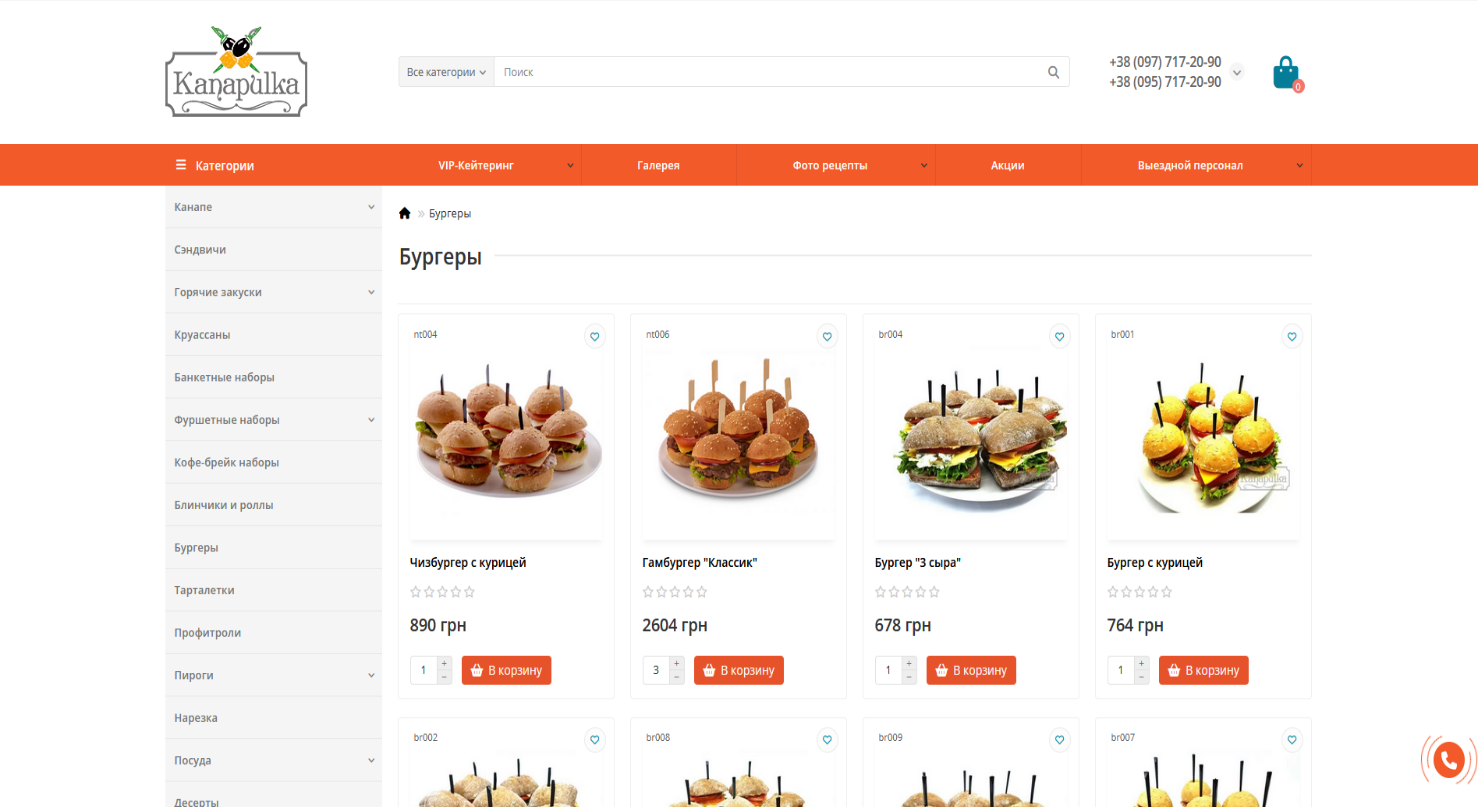


Рис. 1.1. Головна сторінка сайту Kanapulka

Головні недоліки функціоналу та дизайну сайту Kanapulka:

— відсутність темної теми, що значно ускладнює перегляд сайту;

— неякісні фотографії продукції, що не визиваюсь апетиту;

— недостатній опис страв, відсутність інформації про склад, калорійність тощо.

Схожий за тематикою та призначенням є сайт Instafood представлений на рисунку 1.2.

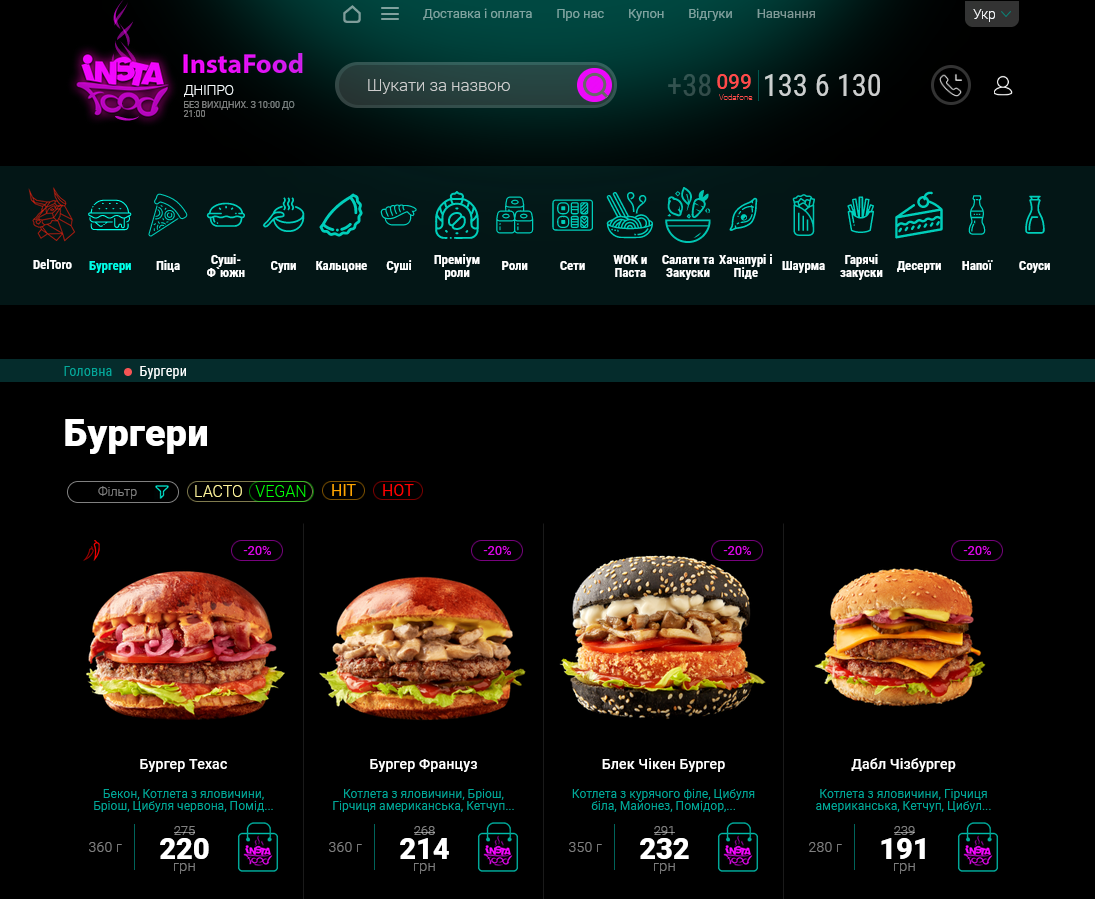


Рис. 1.2. Головна сторінка сайту Instafood

Перевагами цього сайту є можливість залишити «лайк» на продукції, а також можливість надіслати відгук про якість продукції. Серед недоліків цього сайту варто виділити неможливість видалення поставленого «лайку» і відсутня можливість побачити, які відгуки залишили інші користувачі.

У підсумку більшість існуючих систем показали певні недоліки у сфері дизайну та функціональності. Враховуючи ці фактори, розробка нової веборієнтованої інформаційної системи стає актуальною та цікавою задачею з точки зору покращення користувацького досвіду. Саме тому, темою виконання курсової роботи було обрано: веборієнтовану інформаційну систему для онлайн заказу бургерів.

**1.2. Призначення розробки та галузь застосування**

Тематика бургерів є однією з найбільш актуальних і широко обговорюваних у сфері громадського харчування. Захоплення бургерами є поширеним серед різних верств суспільства, інтерес до нових смаків, рецептів та закладів постійно зростає. Завдяки постійному впровадженню нових технологій та інновацій у галузі кулінарії, інформаційна підтримка цієї сфери є важливою.

Розробка веборієнтованої інформаційної системи для онлайн замовлення бургерів має на меті створення платформи для зручного і швидкого замовлення їжі. Така система буде забезпечувати користувачів можливістю легко вибирати та замовляти бургери, отримувати інформацію про нові страви та акції, а також залишати відгуки та оцінки. Сайт буде представляти собою інтегровану інформаційну платформу, яка об'єднає всіх зацікавлених учасників – від клієнтів до працівників закладів харчування. Це сприятиме підвищенню ефективності обслуговування та покращенню взаємодії з клієнтами.

**1.3. Постановка завдання**

Завданням курсової роботиє обґрунтування та розробка веборієнтованої інформаційної системи замовлення бургерів, що включає в себе розробку програмного забезпечення з детальним опрацюванням функціоналу, дизайну.

Для досягнення поставленої задачі в роботі необхідно пройти наступні етапи :

* вибрати архітектурний шаблон(патерн);
* вибрати мову програмування ;
* розробити бізнес-логіку програми;
* налаштувати зв’язок користувача з бізнес-логікою;
* протестувати ПЗ на наявність помилок у бізнес-логіці;
* розробити дизайн інтерфейсу користувача.

**2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ**

2.1. Опис використаної архітектури та шаблонів проектування

Архітектура ПЗ це ряд встановлених методів та правил по яким створюють додаток, налаштовують взаємодію його модулів та компонентів.

До компонентів архітектури відносять:

– Веббраузер. Основний компонент клієнтської сторони, що взаємодіє з користувачем, відповідає за отримання і перевірку у разі необхідності вхідних даних та контролювання взаємодії користувача з застосунком.

– Вебсервер. Компонент серверної сторони, виконує обробку бізнес-логіки сайту, запитів що надходять від клієнтської частини, надсилає запити до необхідного компоненту програми, керує всіма операційними процесами додатку.

– Сервер бази даних. Відповідає за збереження та надання даних необхідних програмі.

У розробці інформаційної системи використовувалась Трирівнева архітектура WEB –застосунку.

Трирівнева архітектура веб-застосунку - це підхід до розробки програмного забезпечення, який розділяє програму на три рівні: клієнтський, серверний та даних.

1. **Клієнтський рівень (презентаційний рівень)**: Це фронтенд, тобто частина програми, яка взаємодіє з користувачем. Клієнтський рівень включає в себе веб-інтерфейс, який користувачі використовують для взаємодії з програмою. Цей рівень може включати HTML, CSS та JavaScript для створення веб-сторінок, які відображаються у веб-браузері.
2. **Серверний рівень (логічний рівень)**: Серверний рівень - це середній шар програми, який обробляє запити користувачів і взаємодіє з базою даних для отримання необхідної інформації. Він містить бізнес-логіку та обробників запитів. Тут можуть використовуватися мови програмування, такі як Python, Java, PHP, а також фреймворки, що допомагають швидко розробляти веб-додатки.
3. **Рівень даних (дані)**: Це рівень, де зберігаються дані, необхідні для роботи програми. Це може бути база даних, файлове сховище або інші джерела даних. Зазвичай використовуються реляційні бази даних, такі як MySQL, PostgreSQL, або NoSQL бази даних.

Трирівнева архітектура дозволяє розділити функціональність програми на логічно відокремлені частини, що полегшує розробку, тестування та супровід коду. Крім того, вона сприяє масштабованості, оскільки кожен рівень може бути масштабований окремо залежно від потреб програми.

Приклад трирівневої архітектури зображено на малюнку 2.1

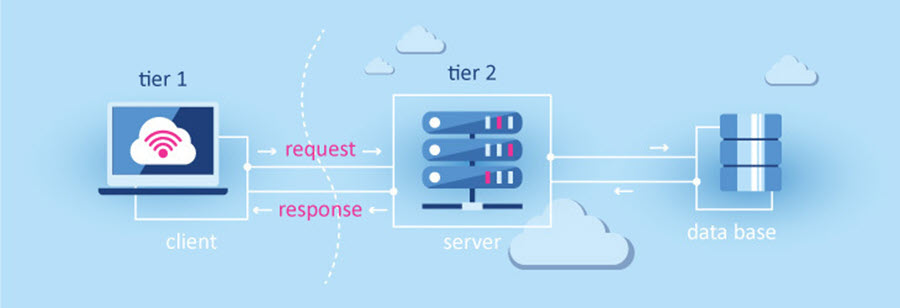


Рис. 2.1. Трирівнева архітектура web-застосунку

**2.2. Опис використаних технологій та мов програмування**

Використані технології та мови програмування можна розділити на клієнтську та серверну частини.

**Клієнтська частина** дозволяє користувачам додатку взаємодіяти з даними та логікою через браузер. Саме на цьому рівні користувач взаємодіє з UI/UX дизайном сайту, інтерактивними елементами, інформаційними панелями. Далі будуть представлені технології та мови, що були використані для створення дизайну клієнтської частини:

– **HTML**. Мова розмітки сторінки, необхідна для структурування наповнення сторінки розробниками.

– **CSS**. Мова таблиць стилів, дозволяє розділяти макет вебзастосунків та їх вміст для сайтів, які розробляються мовами розмітки. CSS допомагає скорочувати код використовуючи створені стилі повторно.

– **JavaScript**. Мова програмування яка допомагає зробити сторінки динамічними, змінюватись сторінкам в залежності від дій користувача без перезавантаження самої сторінки. Використання цієї мови допомагає роботи сторінки більш складними та цікавими при цьому не впливаючи на навантаження сервера.

**Серверна частина** є основною у застосунку, адже саме вона отримує запити від користувачів, які обробляє у бізнес-логіці та повертає відповіді на них з необхідними даними. Включає в себе бази даних, сервери, вебсервіси. Технологія, що була використана для розробки цієї частини:

**– PHP** є скриптовою мовою програмування, яка призначена саме для розробки вебдодатків. Її особливість полягає в тому, що вона не компілюється, а виконується одразу на вебсервері. Це значно полегшує написання, тестування та налагодження вебдодатків. Має широку базу прихильників та активну спільноту, розширену бібліотеку функцій для різних задач.

Опис використаних технологій:

**HTML (HyperText Markup Language)**

HTML (HyperText Markup Language) - це мова розмітки, що використовується для створення веб-сторінок. Вона описує структуру та вміст веб-сторінки, використовуючи теги та атрибути. Браузери інтерпретують код HTML та візуалізують його на екрані користувача.

Структура HTML-документа

Основні елементи HTML-документа:

* <!DOCTYPE html>: Цей декларація вказує тип документа браузеру.
* <html>: Цей тег є кореневим елементом HTML-документа і містить усі інші елементи.
* <head>: Цей тег містить метадані про веб-сторінку, такі як заголовок, опис та посилання на зовнішні файли CSS.
* <body>: Цей тег містить видимий вміст веб-сторінки, який бачить користувач.

HTML-теги

HTML-теги використовуються для визначення елементів веб-сторінки. Теги записуються у парі, де відкриваючий тег містить <, назву тега, атрибути (за бажанням) та >. Закриваючий тег містить </, назву тега та >.

Приклади HTML-тегів:

* <h1>: Цей тег використовується для створення заголовку першого рівня.
* <p>: Цей тег використовується для створення абзацу тексту.
* <a>: Цей тег використовується для створення гіперпосилання.
* <img>: Цей тег використовується для вставки зображення.

Атрибути HTML

Атрибути використовуються для надання додаткової інформації про HTML-теги. Атрибути записуються у парі, де ім'я атрибута та його значення розділені знаком =.

Приклади HTML-атрибутів:

* href: Цей атрибут використовується для визначення адреси гіперпосилання.
* src: Цей атрибут використовується для визначення шляху до зображення.
* alt: Цей атрибут використовується для визначення альтернативного тексту для зображення.

Приклади HTML-коду

1. Проста веб-сторінка:

HTML

<!DOCTYPE html>

<html>

<head>

<title>Моя перша веб-сторінка</title>

</head>

<body>

<h1>Вітаю на моїй веб-сторінці!</h1>

<p>Це лише тестова сторінка.</p>

</body>

</html>

2. Гіперпосилання:

<a href="https://www.example.com">Перейти до Example.com</a>

3. Зображення:

<img src="image.jpg" alt="Зображення">

4. Списки: Існують ненумеровані (<ul>) та нумеровані (<ol>) списки. Кожен елемент списку визначається тегом <li>.

<ul>

<li>Пункт списку 1</li>

<li>Пункт списку 2</li>

</ul>

5.Таблиці: Використовуються для відображення табличних даних.

<table>

<tr>

<th>Заголовок 1</th>

<th>Заголовок 2</th>

</tr>

<tr>

<td>Дані 1</td>

<td>Дані 2</td>

</tr>

</table>

HTML5

HTML5 — це остання версія HTML, яка включає нові елементи, атрибути та поведінку, що дозволяють створювати більш інтерактивні та динамічні веб-додатки. Основні нововведення HTML5 включають:

Нові семантичні елементи: <article>, <section>, <nav>, <header>, <footer>, <aside> та інші, що полегшують структурування контенту.

Форми та введення даних: Нові типи полів для форм, такі як email, date, number, а також нові атрибути, такі як placeholder.

Мультимедіа: Вбудовані елементи для аудіо (<audio>) та відео (<video>), що спрощують інтеграцію мультимедіа без використання сторонніх плагінів.

API для роботи з браузером: Наприклад, API для малювання графіки на полотні (<canvas>), зберігання даних на стороні клієнта (Local Storage), геолокації та інших інтерактивних можливостей.

**CSS (Cascading Style Sheets)**

**CSS (Cascading Style Sheets)** - це мова опису зовнішнього вигляду веб-сторінок. Вона використовується для форматування HTML-елементів, таких як заголовки, абзаци, зображення та посилання. CSS дає веб-дизайнерам більше контролю над зовнішнім виглядом своїх веб-сторінок, роблячи їх більш привабливими та зручними для користувачів.

Переваги використання CSS

* Покращений контроль над зовнішнім виглядом: CSS дає веб-дизайнерам більше контролю над зовнішнім виглядом своїх веб-сторінок, ніж HTML. Це дозволяє їм створювати більш послідовні та привабливі веб-сторінки.
* Зменшення розміру коду: CSS може допомогти зменшити розмір HTML-коду, оскільки правила стилю можна застосовувати до декількох елементів одночасно.
* Покращена доступність: CSS може допомогти зробити веб-сторінки більш доступними для людей з інвалідністю, наприклад, для людей з вадами зору.
* Легкість обслуговування: CSS робить веб-сторінки легшими для обслуговування, оскільки зміни в зовнішньому вигляді можна вносити в одному місці, а не повторювати їх на кожній сторінці.

Основні компоненти CSS

CSS складається з трьох основних компонентів:

* Селектори: Селектори використовуються для вибору HTML-елементів, до яких потрібно застосувати правила стилю.
* Властивості: Властивості визначають атрибути стилю, які потрібно застосувати до вибраних елементів.
* Значення властивостей: Значення властивостей визначають конкретні значення атрибутів стилю.

Структура CSS

Типовий CSS-правилом має наступну структуру:

селектор {

властивість: значення;

}

Наприклад:

body {

background-color: #f0f0f0;

font-family: Arial, sans-serif;

}

Селектори

Селектори в CSS дозволяють застосовувати стилі до конкретних елементів HTML або груп елементів. Існує кілька типів селекторів:

Елементні селектори: Застосовують стилі до всіх елементів певного типу.

p {

color: blue;

}

Класові селектори: Застосовують стилі до елементів з певним класом. Класи визначаються в HTML за допомогою атрибута class.

.highlight {

background-color: yellow;

}

Ідентифікаторні селектори: Застосовують стилі до елементів з певним ідентифікатором. Ідентифікатори визначаються в HTML за допомогою атрибута id.

#main-header {

font-size: 24px;

}

Комбінатори: Селектори можуть комбінуватися для вибору елементів залежно від їхнього розташування в документі.

div p {

margin-bottom: 20px;

}

Атрибутні селектори: Застосовують стилі до елементів з певними атрибутами.

a[target="\_blank"] {

color: red;

}

Властивості та значення

CSS має безліч властивостей, які можна застосовувати до елементів. Ось деякі з найпоширеніших:

Колір та фон:

color: #333;

background-color: #fff;

Текст та шрифти:

font-size: 16px;

font-family: 'Arial', sans-serif;

text-align: center;

Розміри та відступи:

width: 100%;

height: 50px;

margin: 10px;

padding: 20px;

Кордон та обгортка:

border: 1px solid #000;

border-radius: 5px;

Відображення та позиціонування:

display: flex;

position: absolute;

top: 50px;

left: 100px;

Каскадність та спадковість

CSS підтримує концепцію каскадності та спадковості. Каскадність означає, що стилі можуть бути "каскадовані" або переважати один одного залежно від специфічності селектора та його розташування в коді. Спадковість означає, що деякі властивості можуть успадковуватися від батьківських елементів до дочірніх.

Медійні запити

Медійні запити дозволяють застосовувати стилі залежно від типу пристрою або характеристик екрану, таких як ширина, висота, роздільна здатність. Це робить можливим створення адаптивних веб-дизайнів, що виглядають добре на різних пристроях.

@media (max-width: 600px) {

body {

background-color: lightblue;

}

}

CSS3

CSS3 — це сучасна версія CSS, яка включає багато нових властивостей і модулів, що розширюють можливості стилізації. Основні нововведення CSS3 включають:

Градієнти:

background: linear-gradient(to right, red, yellow);

Тіні та трансформації:

box-shadow: 2px 2px 5px rgba(0,0,0,0.5);

transform: rotate(45deg);

Анімації та переходи:

transition: all 0.3s ease;

animation: example 5s infinite;

@keyframes example {

from {background-color: red;}

to {background-color: yellow;}

}

Гнучкі блоки (Flexbox):

display: flex;

justify-content: center;

align-items: center;

Сіткові макети (CSS Grid):

display: grid;

grid-template-columns: 1fr 1fr;

gap: 20px;

**JavaScript**

JavaScript (JS) — це високорівнева, динамічна мова програмування, яка використовується для створення інтерактивних веб-сторінок. Вона була розроблена в 1995 році Бренданом Айком під час роботи в Netscape і швидко стала однією з трьох основних технологій веб-розробки поряд з HTML і CSS.

Основні принципи

JavaScript є мовою сценаріїв, що виконується на стороні клієнта, хоча сучасні серверні платформи, такі як Node.js, дозволяють використовувати її також на серверній стороні. Вона підтримує об'єктно-орієнтоване, імперативне та функціональне програмування, що робить її дуже гнучкою для розробників.

Структура коду

JavaScript-код можна вбудовувати безпосередньо в HTML-документ або зберігати в окремих файлах з розширенням .js, які підключаються до HTML за допомогою тегу <script>.

Приклад вбудованого коду:

<script>

document.querySelector('h1').textContent = 'Привіт, JavaScript!';

</script>

Основні концепції

Змінні та типи даних:

Змінні можуть оголошуватися за допомогою ключових слів var, let та const.

Підтримуються різні типи даних: числа, рядки, булеві значення, об'єкти, масиви, функції тощо.

let число = 10;

const текст = 'Привіт';

var булеве = true;

Функції:

Функції можуть бути оголошені різними способами: функціональні вирази, стрілкові функції, анонімні функції тощо.

function привітання(ім'я) {

return `Привіт, ${ім'я}!`;

}

const додати = (a, b) => a + b;

Умовні оператори:

Використовуються для виконання різних дій залежно від умов.

if (число > 5) {

console.log('Число більше за 5');

} else {

console.log('Число менше або дорівнює 5');

}

Цикли:

JavaScript підтримує кілька видів циклів для ітерації: for, while, do...while, for...of, for...in.

for (let i = 0; i < 5; i++) {

console.log(i);

}

Об'єкти та масиви:

Об'єкти та масиви є основними структурами даних в JavaScript.

const людина = {

ім'я: 'Іван',

вік: 30

};

const масив = [1, 2, 3, 4, 5];

Події та маніпуляція DOM

JavaScript дозволяє взаємодіяти з DOM (Document Object Model), що представляє структуру HTML-документу у вигляді дерева. Це дозволяє створювати динамічні веб-сторінки, змінюючи елементи в реальному часі.

Обробка подій:

Події, такі як кліки, завантаження, відправлення форми, можуть бути оброблені за допомогою JavaScript.

document.querySelector('button').addEventListener('click', () => {

alert('Кнопка натиснута!');

});

Маніпуляція елементами:

Зміна тексту, стилів, атрибутів елементів.

const заголовок = document.querySelector('h1');

заголовок.style.color = 'синій';

заголовок.textContent = 'Новий текст заголовка';

Асинхронне програмування

JavaScript підтримує асинхронне програмування, що дозволяє виконувати завдання, які займають час (наприклад, запити до сервера) без блокування основного потоку виконання коду.

Callback-функції:

Передача функції як аргументу для виконання після завершення асинхронної операції.

setTimeout(() => {

console.log('Через 2 секунди');

}, 2000);

Promises:

Обіцянки (Promises) забезпечують більш елегантний спосіб обробки асинхронних операцій.

fetch('https://api.example.com/data')

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.error(error));

Async/Await:

Синтаксичний цукор для обробки обіцянок, що робить код більш читабельним.

async function отриматиДані() {

try {

const response = await fetch('https://api.example.com/data');

const data = await response.json();

console.log(data);

} catch (error) {

console.error(error);

}

}

отриматиДані();

Модулі

Сучасний JavaScript підтримує модулі, що дозволяють розбивати код на окремі файли та імпортувати їх за потреби. Це робить код більш організованим та підтримуваним.

Експорт та імпорт модулів:

// math.js

export function додати(a, b) {

return a + b;

}

// main.js

import { додати } from './math.js';

console.log(додати(2, 3)); // 5

Бібліотеки та фреймворки

JavaScript має багатий екосистему бібліотек та фреймворків, що спрощують розробку веб-додатків:

React: Бібліотека для побудови користувацьких інтерфейсів, розроблена Facebook.

Angular: Фреймворк для побудови складних односторінкових додатків, розроблений Google.

Vue.js: Прогресивний фреймворк для створення користувацьких інтерфейсів.

**PHP (Hypertext Preprocessor)**

PHP (Hypertext Preprocessor) — це популярна мова сценаріїв загального призначення, яка спеціально створена для веб-розробки. Вона була розроблена в 1994 році Расмусом Лердорфом і з того часу стала однією з найпоширеніших мов для створення динамічних веб-сторінок та веб-додатків.

Основні принципи

PHP є мовою сценаріїв, яка виконується на сервері. Це означає, що код PHP обробляється на сервері, а результати виконання (зазвичай у вигляді HTML) відправляються клієнту. PHP легко вбудовується в HTML, що дозволяє створювати динамічний контент на веб-сторінках.

Структура коду

Код PHP вставляється в HTML-документ за допомогою тегів <?php ... ?>. Типовий PHP-скрипт виглядає наступним чином:

<!DOCTYPE html>

<html>

<head>

<title>Приклад PHP</title>

</head>

<body>

<h1><?php echo 'Привіт, світ!' ?></h1>

</body>

</html>

Основні концепції

Змінні та типи даних:

Змінні в PHP оголошуються за допомогою знака долара $. Вони не мають визначеного типу, тому PHP автоматично визначає тип змінної.

$число = 10;

$текст = "Привіт";

$булеве = true;

Функції:

Функції можуть бути оголошені за допомогою ключового слова function. Вони можуть приймати параметри та повертати значення.

function привітання($ім'я) {

return "Привіт, $ім'я!";

}

Умовні оператори:

PHP підтримує стандартні умовні оператори: if, else, elseif, switch.

if ($число > 5) {

echo "Число більше за 5";

} else {

echo "Число менше або дорівнює 5";

}

Цикли:

PHP підтримує кілька видів циклів: for, while, do...while, foreach.

for ($i = 0; $i < 5; $i++) {

echo $i;

}

Масиви:

Масиви в PHP можуть бути індексованими або асоціативними.

$масив = array(1, 2, 3, 4, 5);

$асоціативний\_масив = array("ім'я" => "Іван", "вік" => 30);

Робота з формами

PHP часто використовується для обробки даних з веб-форм. Це включає отримання даних через глобальні масиви $\_GET, $\_POST, та $\_REQUEST.

// Отримання даних з форми методом POST

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$ім'я = $\_POST['ім'я'];

echo "Привіт, $ім'я!";

}

Підключення до бази даних

PHP надає різні способи для роботи з базами даних, найпопулярнішим з яких є використання MySQL через розширення mysqli або PDO (PHP Data Objects).

// Підключення до бази даних MySQL за допомогою mysqli

$з'єднання = new mysqli('localhost', 'користувач', 'пароль', 'база даних');

// Перевірка з'єднання

if ($з'єднання->connect\_error) {

die("З'єднання не вдалося: " . $з'єднання->connect\_error);

}

// Виконання SQL-запиту

$sql = "SELECT \* FROM користувачі";

$результат = $з'єднання->query($sql);

if ($результат->num\_rows > 0) {

while($рядок = $результат->fetch\_assoc()) {

echo "id: " . $рядок["id"]. " - Ім'я: " . $рядок["ім'я"]. "<br>";

}

} else {

echo "0 результатів";

}

$з'єднання->close();

Робота з файлами

PHP дозволяє читати, записувати та маніпулювати файлами на сервері.

// Запис у файл

$file = fopen("test.txt", "w");

fwrite($file, "Привіт, файл!");

fclose($file);

// Читання з файлу

$file = fopen("test.txt", "r");

echo fread($file, filesize("test.txt"));

fclose($file);

Сесії та куки

PHP забезпечує роботу з сесіями та куками для зберігання даних про користувача між різними запитами.

// Запуск сесії

session\_start();

// Встановлення змінної сесії

$\_SESSION['користувач'] = 'Іван';

// Доступ до змінної сесії

echo $\_SESSION['користувач'];

// Встановлення куки

setcookie("користувач", "Іван", time() + 3600); // Кука діє 1 годину

// Доступ до куки

echo $\_COOKIE['користувач'];

Фреймворки PHP

Для полегшення розробки складних веб-додатків PHP має багато потужних фреймворків, таких як:

Laravel: Один з найпопулярніших фреймворків, відомий своєю елегантністю та простотою.

Symfony: Потужний фреймворк з великою кількістю компонентів, що використовуються в багатьох інших проєктах.

CodeIgniter: Легкий фреймворк, ідеальний для невеликих проєктів та початківців.

Yii: Високопродуктивний фреймворк з великими можливостями для створення веб-додатків.

**ВИСНОВКИ**

У висновках слід підсумувати основні результати дослідження та розробки навчального проекту веборієнтованої інформаційної системи для онлайн замовлення бургерів.

В ході виконання курсової роботи було проведено ґрунтовний аналіз предметної області, вивчено особливості функціонування сайтів для замовлення їжі та їхні потреби у веборієнтованих системах. На основі зібраних даних було сформульовано технічне завдання та спроектовано архітектуру інформаційної системи, яка забезпечує необхідний функціонал для ефективного управління замовленнями, надання інформації про меню, організації процесів доставки тощо.

Практичною реалізацією проекту стала розробка вебдодатку з використанням сучасних технологій та підходів, таких як HTML, CSS, JavaScript та PHP. Впровадження даної системи дозволить закладам громадського харчування оптимізувати свої робочі процеси, покращити якість обслуговування та забезпечити зручний та функціональний вебінтерфейс для користувачів.

Результати дослідження та розробки можуть стати основою для подальшого вдосконалення та масштабування системи, додавання нових модулів та інтеграції з іншими сервісами. Крім того, отриманий досвід може бути корисним для створення аналогічних веборієнтованих інформаційних систем в інших сферах громадського харчування та інших галузях.