



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної підготовки

випускна кваліфікаційна робота бакалавра

**«ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ СЕЛЕКЦІЙНИМ
КОНТЕНТОМ»**

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проєкту

№	П.І.Б.	Група
1	Платов А. М.	КН-Д-191
2	Карпова А. М.	КН-П-192
3	Гуртовий М. В.	КН-П-191
4	Філатова О. С.	КН-П-191

Автор звіту

_____ Філатова Олександра Сергіївна

АНОТАЦІЯ

УДК 004.9

Ф-51

Гуртовий М.В. Інформаційна система управління селекційним контентом: Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса: ОТУШ, 2023 - 14 с.

Важливість цієї роботи зумовлена необхідністю програмного забезпечення з управління селекційним контентом різного призначення, яке базується на фільтруванні результатів за різними критеріями. Ці критерії включають такі фактори, як категорія, оцінки якості і цінові категорії продукту. Розгляд цих критеріїв важливий для проведення комплексної оцінки волатильності результатів і прийняття обґрунтованих рішень. Важливо зазначити, що застосування цих критеріїв може зіткнутися з проблемами, такими як неточність даних або неповна інформація. Отже, це сприяє постановці задач в умовах невизначеності або неповної визначеності вхідних даних. Крім того, актуальність цієї теми підкреслюється необхідністю впровадження ефективних мобільних інформаційних систем і посилення заходів захисту інформації, особливо коли йдеться про забезпечення конфіденційності та безпеки особистих даних.

Предметом дослідження було обрано впровадження інформаційної системи з метою покращення торговельних взаємовідносин між користувачами. У якості задач досліджень було встановлено наступне:

- Аналіз спрямований на визначення характерних особливостей і областей для вдосконалення в інформаційних системах управління селекційним контентом, з фокусом на оптимізації функцій, пов'язаних з управлінням контентом.
- Оцінити програмні засоби та інтерфейси, які забезпечують ефективну співпрацю між користувачами в інформаційній системі
- Впровадити в систему результат попередніх досліджень, щоб полегшити взаємодію користувачів та оптимізувати процес взаємодії.
- Застосувати надійні заходи безпеки в інформаційній системі, щоб захистити дані користувачів і конфіденційність особистої інформації.

Методологія дослідження, яка використовується для досягнення поставлених цілей, передбачає поєднання системного аналізу, моделювання та когнітивних підходів. Методи дослідження перетинаються, щоб сформувати комплексну основу для проведення досліджень.

ЗМІСТ

ВСТУП _____	3
Технічне завдання до проекту _____	4
РОЗДІЛ 1. Загальна характеристика системи _____	6
РОЗДІЛ 2. Реалізація клієнтської частини _____	7
РОЗДІЛ 3. Результати випробувань _____	11
ВИСНОВКИ _____	14
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ _____	16

ВСТУП

Розвиток і широке впровадження інформаційних технологій призвели до оцифрування та міграції численних послуг в онлайн-сферу. У стратегії розвитку інформаційного суспільства в Україні комунікаційні технології визнано необхідними інструментами соціально-економічного прогресу та ключовим каталізатором інноваційного економічного розвитку [1].

З поширенням сучасних мобільних пристроїв, таких як смартфони, планшети та ноутбуки, стали повсякденною нормою нашого життя. Ці пристрої зробили революцію в комунікації людей, сприяючи ефективним і персаналізованим обміном послуг. З появою платформ соціальних мереж, додатків для обміну повідомлень та онлайн-ринків, соціальний аспект спілкування між клієнтом і продавцем зазнав помітних змін. Це динамічне комунікаційне середовище змінило відносини клієнт-продавець, зробивши їх більш інтерактивними, прозорими та орієнтованими на клієнта.[2]

Однак це широке використання також вимагає виникнення потреб в постійному розвитку для просування торгівлі і комерції[3]. Це вимагає постійної еволюції та вдосконалення галузі не лише для оптимізації існуючих процесів, але й для дослідження нових шляхів, які сприяють зростанню та розширенню торгівлі.[3] Отже, це підкреслює необхідність постійних досліджень і розробок у цій конкретній галузі, що слугує каталізатором розвитку торгівлі.

В якості предмету цього дослідження було обрано досить об'ємну систему взаємодії користувачів, яка охоплює отримання пропозицій від певних користувачів і пошук відповідних пропозицій від інших.

Крім того більшість подібних систем повинні включати можливість пошуку бажаних пропозицій, геопозиціювання пропозицій і дотримання торговельних узгоджень, що в свою чергу вимагають надійних заходів авторизації та автентифікації, для забезпечення доступу, а також запобіжних заходів для несанкційного доступу.

Нижче ви знайдете технічне завдання з детальним описом до проекту. Цей звіт зосереджується на серверній частині, який стосується обміну даними із клієнтськими частинами проекту. Звіт складається з трьох розділів: загальна характеристика системи, деталі створення серверної частини та аналіз результатів впровадження та тестування. Висновки містять колективні результати, задокументовані усіма учасниками проекту.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЕКТУ

Загальний опис.

Програмне забезпечення (ПЗ) призначене для створення інформаційної системи управління селекційним контентом. Головна функція ПЗ полягає в наданні користувачу можливості замовити доставку продукції популярних закладів свого міста. Додаткова функція ПЗ має дозволити залишати відгуки та оцінки щодо якості доставки і продукції закладів.

Призначення:

ПЗ орієнтується на різні види закладів за видом послуг: ресторани, кафе, магазини тощо. ПЗ створюється за прикладом сервісу доставки, водночас має бути якомога якіснішим, щоб задовольнити потреби користувача.

Вимоги до функціоналу:

Регістрація користувачів:

Для автентифікації користувачу потрібно створити обліковий запис. Має бути можливість це зробити як за допомогою авторизаційних даних (номера телефону/паролю), так і за допомогою облікових записів популярних соціальних мереж (Google, Facebook, тощо). Кожному користувачу потрібно вказати своє ім'я та/або прізвище для підтвердження своєї особистості кур'єру та спілкування з персоналом за надібністю.

Платформа для розміщення закладів:

Основною функцією ПЗ має бути розміщення та перегляд більшості популярних закладів свого міста для кожного користувача. Має забезпечити усі необхідні засоби для

- перегляду асортименту послуг
- замовлення послуги закладу
- складання рейтингу на основі оцінок усіх користувачів

Асортимент товарів:

Кожен заклад, розташований на платформі, повинен мати свій асортимент контенту (товарів/послуг). ПЗ має забезпечити можливість створювати, видаляти та повертати категорії, підкатегорії послуг, робити виборку по категоріях.

Онлайн кошик для покупок:

Для зберігання обраних послуг/товарів ПЗ повинне для кожного закладу реалізувати функції кошика:

- додавання товару
- видалення товару
- підрахунок загальної вартості всіх доданих товарів

Кошик для конкретного закладу має зберігати інформацію при перегляді ПЗ на різних пристроях (веб-додаток та мобільний додаток).

Оформлення замовлення:

Після додавання товарів у кошик користувачу надається можливість замовити послугу доставки за вказаною адресою. ПЗ має розрахувати повну вартість з урахуванням вартості доставки від окремого закладу до адреси користувача.

Зворотній зв'язок

Після завершення замовлення і успішного надання послуги доставки користувачу пропонується надати відгук та скласти оцінку якості роботи кур'єрів для покращення роботи сервісу. Оцінки та відгуки мають відображатися через історію попередніх замовлень.

Архітектура

ПЗ має бути спроектоване за клієнт-серверною архітектурою з розподіленими вузлами:

Backend - серверне програмне забезпечення, що забезпечує збереження та оброблення поточних даних, взаємодію з базою даних. Надає програмний інтерфейс взаємодії (API) з іншими вузлами.

Web App - додаток, доступний з веб-простору (сайт), що забезпечує візуальний інтерфейс користувачів усіх категорій.

Mobile App - додаток, що встановлюється на мобільних пристроях (смартфон, планшет), додатково до Web App забезпечує моніторинг замовлень у фоновому режимі, дозволяє накопичувати дані при відсутності мережного з'єднання (офлайн) з подальшою синхронізацією з Backend.

Технології

Зберігання вихідних кодів ПЗ повинно здійснюватись за допомогою системи контролю версій (Git, за узгодженням групи проекту). Структура файлів та каталогів повинна відбивати архітектуру ПЗ :

- Backend,
- Web,
- Mobile

Крім того, кожен із модулів повинен розміщуватися в окремому репозиторії.

Для кожного з модулів рекомендується (з можливістю часткової зміни) наступні технології створення:

Backend - JS, TS, або JS фреймворки, зокрема NestJS, Express.js

Web - JS, HTML, CSS, або JS фреймворки, зокрема ReactJS, AngularJS

Mobile - Java, або Kotlin, або Flutter

Вихідний код повинен оформлятися у відповідності до загальноприйнятих стандартів, на зразок стандартів GNU <http://www.gnu.org/prep/standards/>, у тому числі з додержанням вимог до коментування та документування коду.

РОЗДІЛ 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

У відповідності до технічного завдання (ТЗ) система була спроектована за клієнт-серверною розподіленою архітектурою. У складі системи було умовно відокремлено наступні архітектурні частини:

- «Mockup»,
- «Backend»,
- «Frontend».

Графічна частина «Mockup» являє собою набір моделей та описів поведінки користувача. Модель подає веб-інтерфейси та мобільні інтерфейси взаємодії користувачів різного типу з інформаційною системою. При створенні інтерфейсів застосовано алгоритми розробки карти застосунку, прототипування, створення UI кітів, принципи із підходами UI/UX, висновки колористики та ергономіки мобільних пристроїв. Частина розміщена за адресою <https://www.figma.com/file/9xXtbafRd6wiTVc0YZaZEK/HERMES-Delivery?type=design&node-id=0-1&mode=design&t=R7JnLO2P93IMPhHZ-0>

Серверна частина «Backend» реалізована як централізований вузол збереження та оброблення даних, що створюються та модифікуються протягом життєвого циклу інформаційної системи. Дана частина включає у себе віртуальний приватний сервер (VPS), базу даних та програмний інтерфейс (API) доступу до основних функцій бізнес-логіки, зокрема, автентифікація, перегляд контенту закладів, категорій та товарів, управління онлайн-кошиком, оформлення замовлення.

Частина «Backend» розміщена на віртуальному сервері Amazon Linux, виконана на базі програмного забезпечення Node.js та СУБД MongoDB. Додатково використані платформа для створення серверних додатків NestJS та програмні пакети Express, Mongoose.

Вихідний код частини розміщено у репозиторії <https://github.com/hermes-delivery-app/hermes-backend-api.git>. Подальший звіт присвячений детальному опису цієї частини.

Частина «Frontend» є шлюзом для доступу користувачів до функціональних можливостей системи через візуальний інтерфейс. Ця система не тільки забезпечує безперебійну взаємодію, але й містить функції для локального зберігання даних, що дозволяє частково використовувати систему за відсутності мережі передачі даних.

Вихідні коди та ресурси даної частини розміщені на репозиторії https://github.com/hermes-delivery-app/hermes_delivery_android_version.

РОЗДІЛ 2

РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ЧАСТИНИ

Клієнтська частина інформаційної системи може бути реалізована з використанням наступних технологій:

Технології розроблення: мова програмування JavaScript ; HTML і CSS для створення розмітки та стилізації.

Функціональні можливості, які можуть бути реалізовані в клієнтській частині:

Аутентифікація та реєстрація користувача:

- Відображення форми входу/реєстрації для користувача.
- Валідація введених даних та відправка їх на сервер для перевірки.
- Збереження токена аутентифікації (наприклад, використовуючи механізм локального сховища, такий як localStorage або sessionStorage).

Перелік контенту:

- Запит до сервера для отримання списку закладів, категорій та товарів.
- Відображення цих даних на веб-сторінці, можливо, використовуючи компоненти, таблиці або фільтри.

Онлайн-кошик:

- Додавання товарів до кошика після натискання на відповідну кнопку або іншу дію.
- Збереження стану кошика на клієнтській стороні, можливо, використовуючи локальне сховище.
- Оновлення та видалення товарів з кошика.

Оформлення замовлення та відгуки:

- Заповнення форми замовлення з необхідними даними (адреса доставки, контактна інформація і т.д.).
- Відправка даних замовлення на сервер для обробки.

Автентифікація

Працює таким чином, що при введенні користувачем даних логіна надсилається запит на сервер, і в разі, якщо отримано значення сервера 200 і дані автентифікації, як-от access token і refresh token, то ми активуємо сесію для користувача з його авторизаційними даними, що зберігаються тепер уже не просто на сервері, а й у локальному сховищі браузера.

```
const loginForm = document.getElementById('loginForm');

loginForm.addEventListener('submit', function (event) {
  event.preventDefault(); // Prevent form submission

  const phone = loginForm.elements.phone.value
  const password = loginForm.elements.password.value

  const data = {
    "phone": phone,
    "password": password
  };

  fetch('http://ec2-107-23-252-231.compute-1.amazonaws.com:3000/auth/signin', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  })
  .then(response => response.json())
  .then(result => {
    // Handle the server response
    console.log(result['statusCode'])
    const accesstoken = result['accessToken']
    const refreshToken = result['refreshToken']
    localStorage.setItem('accessToken', accesstoken)
    localStorage.setItem('refreshToken', refreshToken)

    if (result['statusCode'] == undefined) { return }
    if (result['statusCode'] != 200) {

      const resp = document.getElementById('ServerResponse')
      resp.textContent = `*${result['message']}`
    }
  })
})
```

Показ користувачеві всіх продуктів

Користувач під час переходу на головну або сторінку з продуктами неявно відправляє запит на сервер для отримання списку категорій продуктів магазинів і самих продуктів. Потім сервер обробляє цей запит і повертає відповідну інформацію. Сторінка, на якій відображається список продуктів, використовує JavaScript для динамічного створення контенту і застосування стилів до нього.

```
3
4
5 window.onload = function () {
6     const elementRow = document.getElementById("elementRowp");
7     var xmlhttp = new XMLHttpRequest();
8     xmlhttp.open("GET", "http://ec2-107-23-252-231.compute-1.amazonaws.com:3000/categories", false); // false for synchronous re
9     xmlhttp.send(null);
10    console.log(xmlhttp.responseText);
11    let a = JSON.parse(xmlhttp.responseText)
12    let k = 0
13    for (var i of a['categories']) {
14        k++;
15        if (k > 8) {
16            break
17        }
18        const element = document.createElement("div");
19        element.style.width = "200px"
20        element.style.height = "100px"
21        element.style.textAlign = "center"
22        element.style.borderRadius = "10px"
23        element.style.textShadow = "#FC0 1px 0 10px;"
24        element.style.border = '1px solid black';
25        element.style.boxShadow = '2px 2px 4px rgba(0, 0, 0, 0.4)';
26        element.style.boxShadow = ' -12px 0 8px -4px rgba(0, 0, 0, 0.8)';
27        element.style.color = "white"
28        element.style.fontSize = "25px"
29        element.style.margin = "5px"
30        element.classList.add("col-md-3", "mb-3");
31        element.style.textShadow = "2px 2px 4px rgba(0, 0, 0, 1)"
32        element.id = `${i["_id"]}`
33        // Додаємо класи Bootstrap для створення 4 стовпців в ряді
34        element.textContent = i['name'];
35        element.style.backgroundImage = `url(${i['image']})`
36        element.style.backgroundSize = "cover"
37        element.style.backgroundRepeat = "no-repeat"
38        element.style.transitionDuration = "0.4s"
39        element.addEventListener('click', function () {
40            // d.style.width = "210px"
41            // d.style.height = "110px"
42            window.location.href = `/category.html?categoryId=${encodeURIComponent(element.id)}`;
43        });
44        element.addEventListener('mouseenter', function () {
45            // d.style.width = "210px"
46            // d.style.height = "110px"
47            element.style.transform = "scale(1.3)"
48            element.style.backgroundImage.filter = 'blur(5px)';
49        });
50        element.addEventListener('mouseleave', function () {
51            // d.style.width = "200px"
52            // d.style.height = "100px"
53            element.style.backgroundImage.filter = 'blur(0)';
54            element.style.transform = "scale(1)"
55        });
56    }
57 }
```

Замовлення і відображення в кошику

Коли користувач відвідує сторінку продукту в магазині, він може вибрати кількість товару для покупки. Після натискання на кнопку "Купити", товар додається в кошик конкретного користувача. Для цього відправляється запит на сервер з ідентифікатором користувача, і товар заноситься в таблицю "cart" для цього користувача.

Потім користувач заповнює певну форму, в якій вказує свої контактні дані та адресу для доставки. Якщо користувач уже авторизований, дані автоматично підтягуються у форму.

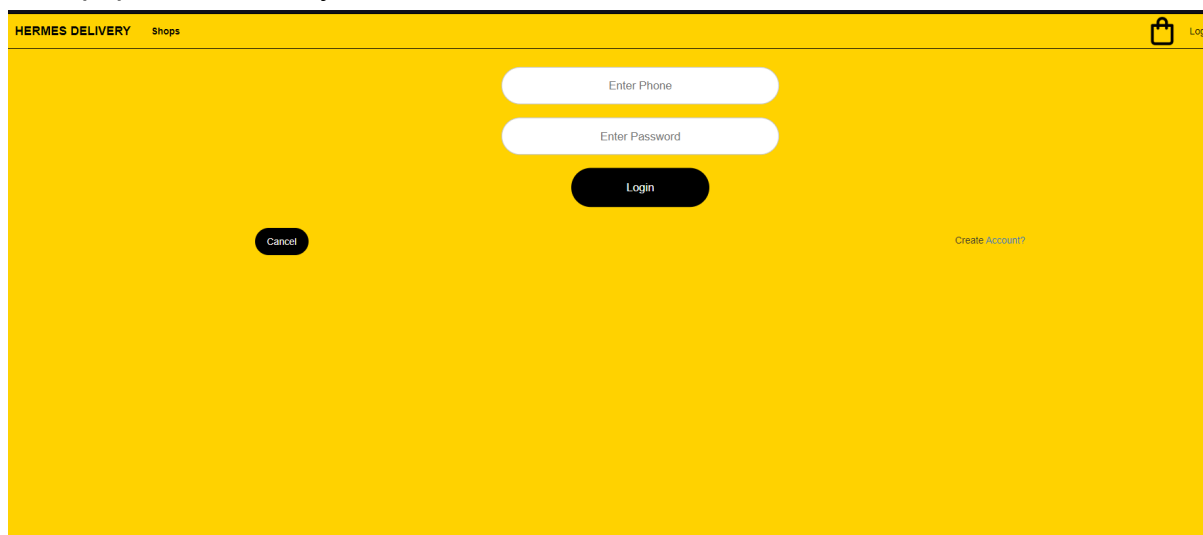
РОЗДІЛ 3 РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

При відвідуванні головної сторінки користувачеві надається широкий вибір продукції та категорій для покупки. Крім того, система логіна і реєстрації бездоганно функціонує, включаючи всі необхідні валідатори даних для забезпечення коректного введення інформації.

При здійсненні покупки товару ефективно працює механізм додавання даних до бази даних. Це гарантує, що всі замовлення зберігаються в безпеці та надійності, забезпечуючи безперебійну роботу системи навіть за високих навантажень. Надійність і стабільність функціонування системи важливі, щоб забезпечити позитивний досвід користувачів і запобігти можливим помилкам в обробці замовлень.

Завдяки ретельному моніторингу та тестуванню, система готова впоратися з високими навантаженнями та забезпечити стабільну роботу. Незалежно від кількості користувачів і замовлень, система надійно обробляє запити та гарантує збереження інформації.

Ми прагнемо до постійного поліпшення та оновлення системи, щоб забезпечити її ефективність і надійність. Ми здійснюємо регулярні резервні копії даних і застосовуємо передові методи захисту інформації, щоб забезпечити безпеку всіх замовлень і конфіденційність користувачів. Усі ці заходи вживаються для того, щоб гарантувати, що наша система працює бездоганно і є надійною платформою для покупок.



HERMES DELIVERY Shops

Enter Name

Enter Phone

Enter Password

Repeat Password

Register

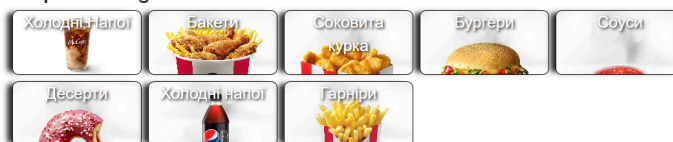
Cancel

Already have an account? [Login](#)

HERMES DELIVERY

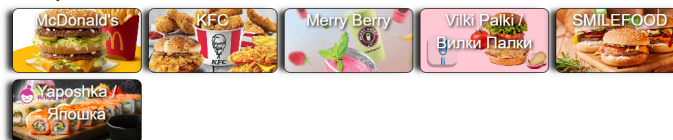
© Copyrights Reserved to Hermes Delivery LLC

Popular Categories



Discounts

Shops



HERMES DELIVERY Shops Categories

X

Холодні Напої

Бакети

Соковита курка

Бургери

Соуси

Десерти

Холодні напої

Гарніри

Гарячі напої

КАВОВІ НАПОЇ

ЛИМОНАДИ

АЙС КАВА

СМУЗІ

ДЕСЕРТИ

СЕТИ

Бакети

Хот Бакет 26 *

Хот Бакет 18 *

Бакет 8 ніжок

Бакет 8 ніжок гострий*

Дует Бакет гострий *

Дует Бакет ориг.

Хот Бакет 12 *

Сандерс Бакет OR

HERMES DELIVERY
© Copyrights Reserved to Hermes Delivery LLC

HERMES DELIVERY Shops Categories

Бакет 8 ніжок

-

+

6

1884₴

8 ніжок обсмажених в хрусткій оригінальний па

13

ВИСНОВКИ

Було проведено аналіз інформаційних систем, спрямований на визначення характерних особливостей і областей для вдосконалення в інформаційних системах управління селекційним контентом, з фокусом на оптимізації функцій, пов'язаних з управлінням контентом. У ході цього аналізу було встановлено категорії типових користувачів системи управління селекційним контентом та їх потреб у роботі з контентом:

- Автори створюють контент, який публікується в системі, вони можуть потребувати інструментів для створення та редагування контенту, налаштування параметрів публікації, додавання медіа-елементів, а також можливість відстежувати популярність свого контенту;
- Користувачі системи відвідують застосунок або сайт, переглядають та споживають контент авторів, вони мають потребу в зручному та легкому способі навігації по контенту, можливості пошуку, фільтрації та сортування, коментування та оцінювання контенту.
- Адміністратори займаються управлінням, налаштуванням прав доступу та контролем за безпекою системи, потребують інструментів для управління обліковими записами, призначення ролей та дозволів, виявлення можливих порушень.

Оцінка програмних засобів та інтерфейсів, які забезпечують ефективну співпрацю між користувачами в інформаційній системі, вимагала оцінки декількох ключових аспектів, таких як зручність використання, функціональність і сумісність. Після загального огляду основних аспектів інформаційних систем управління селекційним контентом було виявлено функціональні можливості, як здатність додавати, редагувати, видаляти та категоризувати контент різних типів, можливість створювати категорії, рубрики та сторінки для організації контенту, функції пошуку, фільтрації та сортування. На продуктивність таких систем сприяє швидка обробка операцій з контентом, мінімальний час завантаження сторінок та швидкість відгуку системи на дії користувачів. Іншим важливим аспектом є розширюваність - можливість розширення функціональності та внесення змін без проблем, гнучкість та масштабованість системи для збільшення обсягу контенту.

Було визначено основні проблеми та виявлено проблемні області, які потребують вдосконалення в системі. Це недоліки, які впливають на продуктивність та задоволення користувачів: швидкість роботи з великим обсягом контенту, складність використання інтерфейсу, неефективний процес редагування тощо. Для вдосконалення системи управління контентом можна задіяти ряд нових технологій, методів та практик, що поліпшать функціональність, продуктивність та зручність використання. Наприклад, розробка системи на основі мікросервісної архітектури, що дозволить легко масштабувати та розширювати функціональність; впровадження гнучкої структури контенту, яка дозволить розділити клієнтську та серверну частини,

забезпечуючи більшу гнучкість в розробці застосунків для різних пристроїв; застосування динамічних шаблонів та компонентів, що дозволяють швидко реорганізувати вміст; впровадження пошукової оптимізації (SEO) та аналітики; використання підходу “Mobile First” при проектуванні та розробці, зосереджуючи увагу на мобільних користувачах; розробка адаптивного дизайну, що забезпечує оптимальне відображення та функціональність на різних пристроях; використання шифрування та протоколів HTTPS для захисту передачі даних між клієнтом та сервером.

Впроваджена система управління селекційним контентом містить адаптивний дизайн для інтеграції з різними пристроями, гнучку мікросервісну архітектуру, використання шифрування для захисту конфіденційної інформації, підтримку API для інтеграції з іншими системами та додатками, та можливість розширювати функціональність системи. Відповідно до розподіленої клієнт-серверної архітектури система складається з трьох окремих компонентів, таких як «Mockup», «Backend» та «Frontend», розташованих за адресами:

- «Mockup»

<https://www.figma.com/file/9xXtbafRd6wiTVc0YZaZEK/HERMES-Delivery?type=design&node-id=0-1&mode=design&t=R7JnLO2P93IMPhHZ-0>

- «Backend»

<https://github.com/hermes-delivery-app/hermes-backend-api.git>.

- «Frontend»

https://github.com/hermes-delivery-app/hermes_delivery_android_version

Система пройшла тестування, включаючи встановлення на фізичних пристроях і численні сеанси зв'язку. Результати цих випробувань підтвердили загальну ефективність системи та продемонстрували її придатність для сприяння безперебійному обміну контентом між усіма категоріями типових користувачів системи управління селекційним контентом.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

Книги:

1. "JavaScript: The Good Parts" - Дуглас Крокфорд
2. "Eloquent JavaScript" - Марейн Хавербеке
3. "HTML and CSS: Design and Build Websites" - Джон Дакетт
4. "CSS Secrets: Better Solutions to Everyday Web Design Problems" - Леа Веру

Сайти та блоги:

5. MDN Web Docs (developer.mozilla.org) - Підтримується Mozilla Foundation і надає велику документацію з HTML, CSS і JavaScript.
6. CSS-Tricks (css-tricks.com) - Ресурс, де ви знайдете статті, керівництва та демонстрації з CSS і фронт-енд розробки загалом. Автор - Кріс Койєр.
7. Smashing Magazine (smashingmagazine.com) - Видання з фокусом на веб-дизайн і розробку, що пропонує статті, посібники та інструменти. Різні автори.
8. A List Apart (alistapart.com) - Журнал про веб-дизайн і розробку, з акцентом на стандарти та передові технології. Різні автори.
9. Scotch.io (scotch.io) - Сайт із підручниками, посібниками та прикладами з веб-розробки. Різні автори.

YouTube-канали:

10. Traversy Media - Канал, ведений Бредом Треверсі, пропонує відеоуроки з різних аспектів веб-розробки, включно з front-end технологіями.
11. The Net Ninja - Канал, який очолює Шон Піннок, пропонує практичні відеоуроки з різних тем, включно з HTML, CSS, JavaScript і фреймворками.

Навчальні платформи:

12. Codecademy (codecademy.com) - Платформа з інтерактивними курсами з різних мов програмування і технологій, включно з front-end розробкою.
13. FreeCodeCamp (freecodecamp.org) - Безкоштовна онлайн-платформа, що пропонує навчальні матеріали та практичні проекти з front-end та інших аспектів веб-розробки.

