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

Одеса – 2023

АНОТАЦІЯ

УДК 004.9
Ф-51

Філатова О. С. 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.

Ця робота є дослідженням зосередженим на розробці і впровадженні інформаційної системи з метою покращення управління селекційним контентом на прикладі торговельних взаємовідносин між користувачами. Окрім цього система містить такі функциональності як: авторизація і автентифікація, механізми обмеження доступу і також можливість обмежувати канали зв'язку між клієнтом та компанією-продавцем.

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 критевіїв 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

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 винекненню 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 проєкту. 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:
Реєстрація 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 контроля 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 СЕРВЕРНОЇ ЧАСТИНИ

Серверна частина інформаційної системи реалізована з використанням наступних технологій:

- Конфігурація програмного забезпечення: віртуальний сервер Amazon Linux 2023 AMI, наданий веб-сервісом хмарних обчислень Amazon Elastic Compute Cloud.
- Засоби розроблення: середовище Visual Studio Code; оточення Node.js; мова програмування TypeScript.
- Додаткове ПЗ та засоби: платформа для створення масштабованих серверних додатків NestJS, фреймворк HTTP-сервера Express.js у складі NestJS; модуль Swagger для генерації документа, який відповідає специфікації OpenAPI; СУБД (система управління базами данных) MongoDB.

У складі серверної частини (backend) передбачено та реалізовано наступні функції:

- Автентифікація користувача, реєстрація нових користувачів.
- Перелік контенту (закладів, їх категорій та товарів).
- Онлайн-кошик для покупок, де зберігається облік товарів.
- Оформлення замовлення та можливість поставити відгук наданій послузі.

Робота із додатком починається з автентифікації користувача, зокрема реєстрації його як учасника інформаційної системи. Для реєстрації користувачу необхідно надати відомості про номер телефону, ім’я та пароль. Перш ніж створити обліковий запис і додати користувача до бази даних, усі наданні дані мають пройти попередню валідацію, яка реалізована за допомогою декораторів властивостей NestJs та регулярних виразів :

```
@IsString()
@IsNotEmpty()
@Matches(/^\+?3?8?(0\d{2}\d{3}\d{2}\d{2})$/)
@ApiProperty()
readonly phoneNumber: string;

@IsString()
@IsNotEmpty()
@MinLength(8)
@Matches(/^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9]).{8,}$/)
@ApiProperty()
readonly password: string;
```

Номер телефону має бути у форматі типового номеру для України : перші три символи - код країни, наступні два - код оператора, решта сім цифр - номер клієнта. Пароль має містити мінімум вісім символів : принаймні одну велику латинську літеру, принаймні одну малу латинську літеру, принаймні одну цифру. Після успішної валідації пароль шифрується за допомогою алгоритму Argon2 для захисту інформації.

Авторизація користувача здійснюється введенням номера телефону та паролю, які збігаються із зазначеними при створенні облікового запису. Клієнт отримує access та refresh токени, створені за JSON Web Token стандартом. Токени містять ідентифікатор користувача та його номер телефону.

```
        async signIn(authDto: AuthDto) {
          const { phoneNumber, password } = authDto;

          const user = await
this.usersService.getOneByPhoneNumber(phoneNumber);
          if (!user)
      throw new BadRequestException('User does not exist');
          if (user.isArchived)
      throw new BadRequestException('User is deleted');

          if (!await argon2.verify(user.password, password))
      throw new BadRequestException('Password is incorrect');

          const tokens = await this.getTokens(user._id, user.phoneNumber);
          await this.updateRefreshToken(user._id, tokens.refreshToken);
          return tokens;
        }
```

Архітектура серверної частини проєкту “клієнт-сервер” складається з 3 рівнів.

1. Рівень доступу до даних: цей рівень забезпечує доступ до даних, що зберігаються в базі даних MongoDB, і реалізує логіку доступу. Рівень розроблено за допомогою модулю “mongoose” і використанням декораторів NestJs @Schema.
2. Контролери - цей рівень забезпечує обмін даними із клієнтською частиною і реалізує основну частину взаємодії із додатком. Контролери відповідають за обробку вхідних запитів і повернення відповідей клієнту у форматі JSON. Механізм маршрутизації встановлює, який контролер отримує запити. Рівень розроблено із використанням декораторів NestJs @Controller і декораторів маршрутизації @Post, @Get, @Put, @Delete. Декоратори пов’язують класи з необхідними метаданими та дозволяють створювати карту маршрутизації.

Приклад методу POST для створення нової категорії :

```
 @Post()
  async create(@Res() response, @Body() createCategoryDto:
CreateCategoryDto) {
    try {
      const newCategory = await
this.categoryService.create(createCategoryDto);

      return response.status(HttpStatus.CREATED).json({
        message: 'Category has been created successfully',
        newCategory: newCategory
      });
    } catch (err) {
      return response.status(HttpStatus.BAD_REQUEST).json({
        statusCode: 400,
        message: 'Error: Category not created!',
        error: 'Bad Request'
      });
    }
  }
```

3. Сервісний рівень: цей рівень містить лише бізнес-логіку - усі операції та методи CRUD, щоб визначити, як можна створювати, зберігати й оновлювати дані.
   Наприклад, пошук закладів по назві :

```
async search(query : string): Promise<IShop[]> {
    const data = await this.shopModel.find({
      $text:
      {
        $search: query,
      }
    });

    if (!data || data.length == 0) {
      throw new NotFoundException('Shops data not found!');
    }
    return data;
  }
```

З повним кодом серверної частини проєкту можна ознайомитись у репозиторії https://github.com/hermes-delivery-app/hermes-backend-api.git

РОЗДІЛ 3
РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Результатом серверної частини проєкту являється JSON-файл для кожного HTTP-методу, який містить інформаційні дані, що передаються клієнтській частині, що може бути веб-застосунком (сайт) або мобільним застосунком. Для головних випробувань, звіт з яких надається далі, програмний код було встановлено та скомпільовано на віртуальному приватному сервері з «Amazon Linux», що є операційною системою «Linux» від Amazon Web Services (AWS).

Для покращення процесу тестування і користування серверною частиною встановлено модуль Node.js «Swagger», який генерує інтерактивну специфікацію для RESTful API з повним описом методів : моделі, параметри і тіла запитів, коди відповідей тощо. Розглянемо різні шляхи - кінцеві точки(ресурси), які надає API. Для кожного шляху визначено операції - методи HTTP, які можна використовувати для доступу до цього шляху, наприклад GET, POST або DELETE.

Шлях /auth відповідає за операції автентифікації, такі як реєстрація, авторизація, вихід з облікового запису, та оновлення токенів доступу. Реєстрація являє собою POST метод, який приймає в тілі запиту ім'я, номер телефону, пароль для створення нового облікового запису, а також стан запису (що його не заархівовано). Після пройденої валідації та успішного запиту користувач додається в СУБД «MongoDB» до колекції «users», а його пароль шифрується перед додаванням. Авторизація - це метод, який приймає номер телефону та пароль, при успішному введенні даних користувача, що збігаються з даними, введеними при реєстрації, видається access і refresh JWT-токени. Вихід є GET методом, на якому стоїть захист: якщо в заголовку запиту передається access token - виконується вихід (токен користувача обнуляється) та повертається статус 200(ok), якщо у запиті немає токена - повертає помилку 401 (unauthorized).

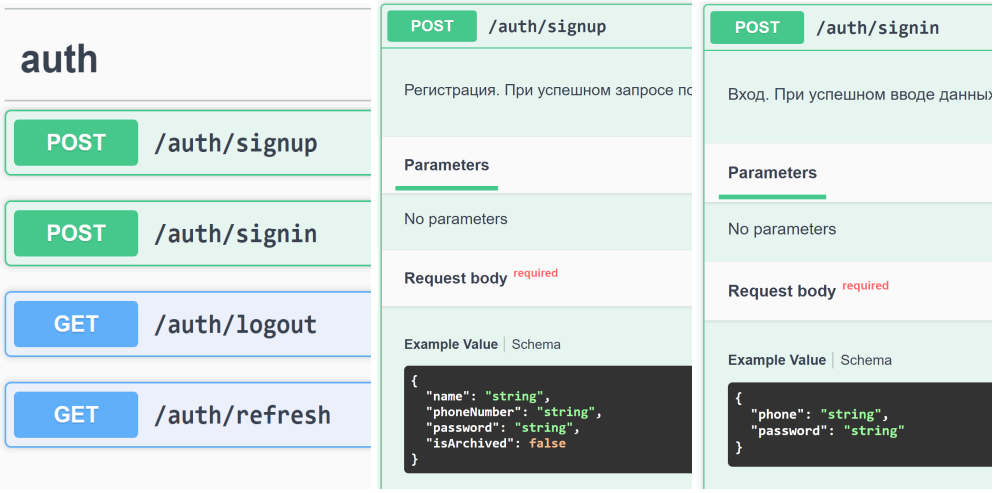

Рисунок 1 - Операції шляху /auth

Шляхи /shops, /categories, /items містять основний селекційний контент для додатка. На прикладі шляху /shops реалізовані такі операції, як додавання до колекції, вивід усіх закладів, вивід інформації про заклад по id, редагування інформації про заклад по id, видалення - перенесення до архіву, пошук закладів, можливість поставити позитивну чи негативну оцінку.

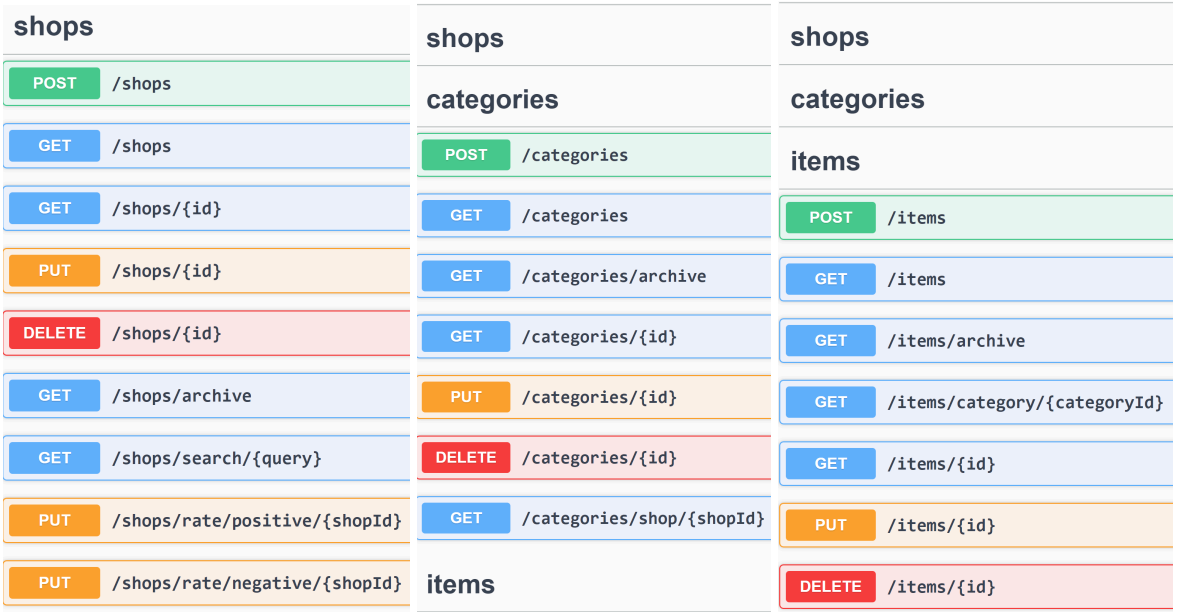

Рисунок 2 - Шляхи /shops, /categories, /items

Шлях /cart реалізує операції з онлайн-кошиком для покупок, які надають можливість створити або закрити кошик, додати або видалити товар із кошика з перерахунком вартості, відобразити всі кошики для конкретного користувача або заклада по наданому id.

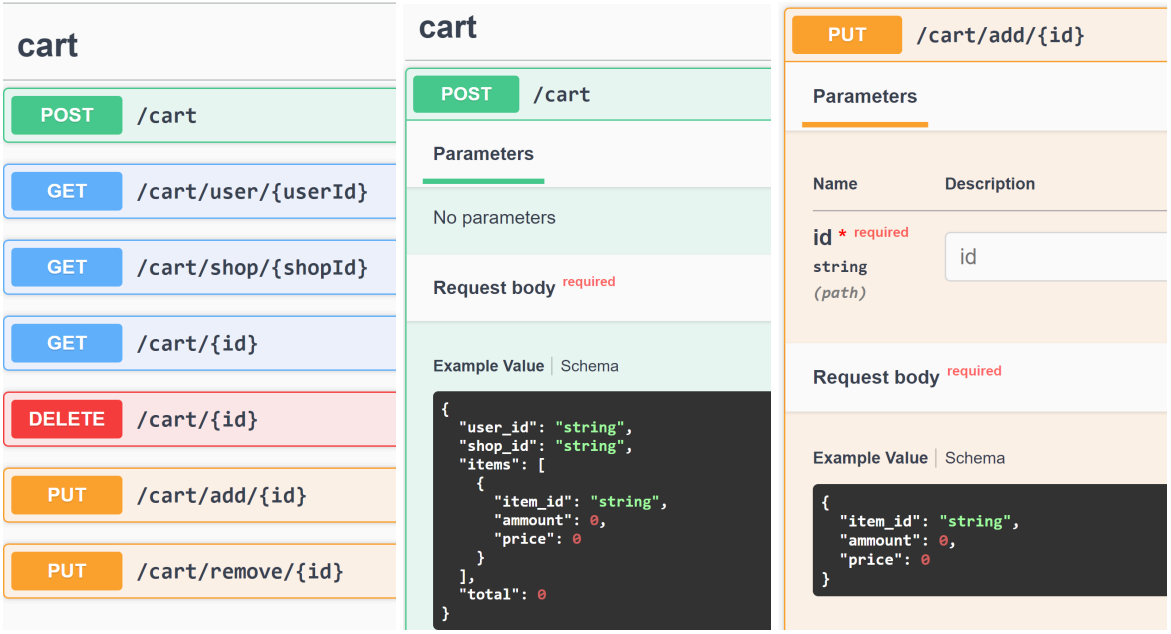

Рисунок 3 - Операції шляху /cart

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=R7JnLO2P93lMPhHZ-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

1. MongoDB Documentation. URL: https://www.mongodb.com/docs/
2. Documentation | NestJS - A progressive Node.js framework. URL: https://docs.nestjs.com/
3. TypeScript: Documentation - TypeScript for JavaScript Programmers. URL: https://www.typescriptlang.org/docs/handbook/
4. OpenAPI Specification v3.1.0 | Introduction, Definitions, & More. URL: https://spec.openapis.org/oas/v3.1.0
5. Swagger Documentation. URL: https://swagger.io/docs/
6. О. В. Грицунов. Інформаційні системи та технології : Навчальний посібник / О. В. Грицунов; 2010. – 222 с. ISBN 978-966-695-195-6
7. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп’ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. − Київ. – 2017. – 110 с.
8. Величко О.М. Інтелектуальні інформаційні системи: структура і застосування. Підручник / Величко О.М., Гордієнко Т.Б. - Херсон. - 2022. - 728 с. ISBN: 978-966-289-552-0
9. Чемакіна О.В. Дизайн системи візуальної інформації. Навчальний посібник / Чемакіна О.В., Рубцов А.Л., Свірко В.О. - Херсон. - 2019. - 200 с. ISBN: 978-966-289-216-1
10. Чайковська М.П. Концептуально-методологічні засади управління маркетинговими ІТ-проєктами в умовах цифрових трансформацій. Монографія / Чайковська М.П. - Херсон. - 2021. - 370 с.ISBN: 978-966-289-537-7