**Голова комісії представляє наступного студента.**

Доброго ранку, шановна комісія!

**Слайд 1:**

До вашої уваги курсова робота на тему «Реалізація багатошарової клієнт-серверної архітектури на прикладі створення системи онлайн продаж»

**Слайд 2:**

Метою курсової роботи є створення сервісу онлайн продаж

**Слайд 3:**

Сервіс може створювати, видаляти та змінювати користувачів з різним рівнем адміністрування, категорії та типи продуктів, поставників продуктів, продукти, відгуки до продуктів та замовлення

**Слайд 4:**

На екрані представлена схема бази даних

**Слайд 5:**

Курсова робота була виконана на мові програмування TypeScript, яка базується на JavaScript.

Для розробки сервісу було використано Node.js – кросплатформове внутрішнє середовище виконання JavaScript.

В якості головного фреймворку було використано Express - мінімалістичний та гнучкий фреймворк для веб-застосунків, який забезпечує тонкий прошарок базової функціональності для веб-застосунків, Express має в свому розпорядженні безліч допоміжних HTTP-методів та проміжних обробників, тому створювати надійні API легко і швидко.

В якості бази даних було обрано MongoDB, та пакет mongoose для зручної взаємодії з нею.

Swagger та OpenAPI – технології, які дозволяють в коді відразу описувати методи сервісу та в подальшому автоматично генерувати документацію

Cors, multer та body-parser допоміжні пакети, які дозволяють розширити функціонал Express та обробляти більшу кількість різних запитів.

Nodemon, concurrently, eslint та prettier – пакети які пришвидшують розробку сервісу надаючи автоматичний запуск, перевірку та форматування коду

**Слайд 6:**

Середовищем функціонування системи є Node.js

**Слайд 7:**

**Демонстрація сервісу**

В сервісі представлено 9 блоків, кожен з яких має від 3-х до 6 операцій.

Всього операцій в сервісі представлено 44 різних операцій.

Всі запити на витягування даних доступні без авторизації, наприклад, запит на всі категорії. Як можна бачити на екрані цей запит може повернути або 200 код та масив з категоріями, або 500 та повернути помилку. Виконаємо запит щоб подивитись, чи є вже збережені категорії в базі. Як можна бачити в сервісі вже є збережені 6 категорій які можна використовувати.

Для повноцінної роботи з сервісом треба створити адміністратора, але перед цим варто подивитись чи є вже якісь користувачі в системі. Виконуємо запит /user/all. Як можна бачити на екрані в системі вже є 3 користувачі з різними рівнями адміністрування, де 0 – звичайний користувач.

Створемо свого користувача з другим рівнем адміністратора, для цього треба авторизуватися за допомогою root-паролю.

Виконаємо додавання нового користувача, вкажемо дані. В поле accountAdminLevel потрібно передати унікальний ключ рівня адміністратора.

Додаємо користувача та заходимо в систему вже через нього.

Ціллю сервісу є створення замовлень, а для них треба створити продукт.

Як ми можемо бачити в продукті є два приклади, один з усіма полями, а другий з мінімально можливими

Як ми бачимо з відповіді система автоматично створила productAllInfo, ми можем за допомогою унікального ідентифікатора product змінити ці поля.

Змінюємо, показуємо

Створюємо відгук, додаємо його через /productsAllInfo/addReview

Тепер у нас є готовий продукт та готовий користувач, час створити замовлення

Створюємо замовлення

В кожному блоці є багато допоміжних функцій, показати на прикладі категорій.

Показати зміну паролю користувача

**На цьому демонстрацію сервісу завершено**

**Слайд 8:**

Висновки прочитати зі слайду. Всі функції сервісу, які були поставлені в завдані були виконані.

**Доповідь закінчено. Дякую за увагу.**

**Питання комісії, студент відповідає.**

**Голова говорить: Захист закінчено. Тільки тоді можна відключатися.**