**Маршрути (Routes):**

* **Що це таке?**

Маршрути в контексті веб-розробки визначають, як ваш веб-додаток відповідає на різні типи запитів, що надходять від користувачів або інших сервісів.

* **Навіщо?**

Вони допомагають структурувати ваш додаток, розділяючи його функціональність на різні частини. Кожен маршрут може вказувати на конкретні дії або функціональність вашого додатку.

**Контролер (Controller):**

* **Що це таке?**

Контролер - це місце в вашому веб-додатку, де знаходиться логіка обробки даних та виклику функцій для відповіді на конкретні запити.

* **Навіщо?**

Контролери допомагають утримувати ваш код організованим і чистим. Вони відокремлюють логіку обробки даних від маршрутів та інших частин вашого додатку.

**Мій код:**

* **Маршрут (Routes):**

Файл `auth.routes.js` визначає маршрут для аутентифікації користувачів, зокрема для обробки запитів, пов'язаних із входом в систему (логін).

* **Контролер (Controller):**

Файл `auth.controller.js` містить функцію `authUser`, яка відповідає за обробку логіки аутентифікації користувача.

Отже, коли користувач робить HTTP-запит POST на маршрут `/login`, маршрут вказує, що потрібно викликати функцію `authUser` з контролера `auth.controller.js`. Ця функція відповідає за обробку логіки аутентифікації користувача.

**Елементи які використовуються:**

* **prisma/client (^5.8.1):**

***Опис:*** `@prisma/client` - це автоматично згенерована бібліотека клієнта для Prisma, яка дозволяє легко та безпечно взаємодіяти з базами даних у JavaScript та TypeScript.

***Використання:*** Використовується для створення ефективних запитів до баз даних та отримання та зберігання даних.

* **argon2 (^0.31.2):**

***Опис:*** Argon2 - це криптографічний алгоритм хешування паролів, призначений для забезпечення високого рівня безпеки та опору до атак хешування паролів.

***Використання:*** Використовується для безпечного зберігання паролів у хешованому вигляді.

* **colors (^1.4.0):**

***Опис:*** Colors - це бібліотека, яка додає кольори до виводу в консолі, покращуючи його зручність та читабельність.

***Використання:*** Використовується для створення кольорових та виділених повідомлень в консолі, що полегшує відлагодження.

* dotenv (^16.3.1):

***Опис***: Dotenv - це модуль, який завантажує змінні середовища з файлу `.env`, дозволяючи зручно управляти конфігурацією середовища під час розробки.

***Використання:*** Використовується для зберігання конфіденційної інформації, такої як ключі API чи параметри конфігурації, у змінних середовища.

* **express (^4.18.2):**

***Опис:*** Express - це мінімалістичний та гнучкий веб-фреймворк для Node.js, спрощує створення веб-серверів та маршрутів для обробки HTTP-запитів.

***Використання:*** Використовується для розробки серверної частини веб-додатків та API, забезпечуючи швидку реалізацію.

* **express-async-handler (^1.2.0):**

***Опис:***  Express Async Handler - це middleware для обробки асинхронних помилок у виразах Express.

***Використання:*** Використовується для спрощення обробки асинхронних помилок в контролерах Express, полегшуючи код.

* **jsonwebtoken (^9.0.2):**

***Опис:*** JSON Web Token (JWT) - це стандарт для створення токенів, які містять підписану інформацію та використовуються для аутентифікації та авторизації.

***Використання:*** Використовується для безпечної передачі інформації між сторонами та перевірки ідентифікації користувачів.

* **morgan (^1.10.0):**

***Опис:*** Morgan - це middleware для Express, яке надає журналізацію HTTP-запитів на сервер.

***Використання:*** Допомагає відстежувати та аналізувати HTTP-запити, полегшуючи відлагодження та моніторинг.

* **node (^21.2.0):**

***Опис:***  Node.js - це середовище виконання JavaScript на сервері, яке дозволяє використовувати JavaScript для розробки серверних застосунків.

***Використання:***  Використовується для створення серверів, API та інших серверних додатків.

* **nodemon (^3.0.3):**

***Опис:***  Nodemon - це інструмент, який автоматично перезапускає сервер Node.js при змінах у вихідних файлах.

***Використання:***  Забезпечує автоматичний перезапуск сервера для швидкої зміни та відлагодження коду під час розробки.

* **pg (^8.11.3):**

***Опис:***  pg - це драйвер PostgreSQL для Node.js, що дозволяє взаємодіяти з базою даних PostgreSQL.

***Використання:***  Використовується для виконання SQL-запитів та роботи з базою даних PostgreSQL.

* **prettier (^3.2.4):**

***Опис:***  Prettier - це інструмент для форматування коду, який допомагає забезпечити єдинообразний стиль коду у проекті.

***Використання:***  Використовується для автоматичного форматування коду, що полегшує спільну роботу та збереження читабельності.

* **prisma (^5.8.1):**

***Опис:***  Prisma - це набір інструментів для роботи з базами даних, включаючи `@prisma/client` для доступу до даних та Prisma CLI для управління міграціями та моделями даних.

***Використання:***  Спрощує роботу з базами даних, дозволяючи використовувати мови програмування для взаємодії з базовими системами даних.