Politechnika Częstochowska

Wydział Inżynierii Mechanicznej i Informatyki

Sprawozdanie 1 z Web 2.0 – wersja z samym API
MEAN Stack po stronie serwera (back-end)

Kamil Rzeźniczek

Informatyka studia II stopnia niestacjonarne

Aplikacja magazyn książek w Express JS z Mongo DB

1. Uruchomienie aplikacji

Aplikację można otworzyć w programie Microsoft Visual Studio Code, zainstalować potrzebne moduły poleceniem npm install i uruchomić z panelu Run and Debug w Visual Studio Code. Umożliwia to plik launch.json znajdujący się w folderze vscode. Drugą opcją jest użycie komendy nodemon Main.js w konsoli. Nodemon zostanie zainstalowany razem z innymi pakietami. Do projektu dołączone są dwie kolekcje do lokalnej bazy Mongo Db w formacie JSON. Przed uruchomieniem projektu należy utworzyć lokalną bazę danych o nazwie „expressdb" i załadować obie dołączone kolekcje tzn. „Users" i „Books". Oprócz tego dołączony został plik z wyeksportowaną kolekcją programu „Postman" z testowanymi endpointami aplikacji oraz plik o rozszerzeniu .env ze zmiennymi środowiskowymi potrzebnymi do działania aplikacji.

2. Zakres pracy

Aplikacja została przygotowana spełniając wymagania z zadań od 1-6 to znaczy łączy się z bazą danych Mongo DB za pomocą pakietu Mongoose, posiada pełną obsługę CRUD dostępną jedynie dla uwierzytelnionych za użytkowników. Wykorzystuje do tego JSON Web Token. Obiekty w ciałach żądań są globalnie parsowane do formatu JSON z wykorzystaniem pakietu bodyParser.

Aplikacja oprócz logowania umożliwia także rejestrację użytkowników z wysłaniem maila aktywacyjnego na podany adres mailowy. Wykorzystywany klient poczty to Gmail a pakiet wykorzystany do obsługi do to nodemailer. Do jego uwierzytelnienia i działania potrzebne jest specjalnie wygenerowane hasło dostępu dla aplikacji znajdujące się z pliku .env. Odczytywanie zmiennych z tego pliku umożliwia wykorzystanie pakietu dotenv.

Aplikacja umożliwia również odzyskanie hasła i ustawienie nowego z weryfikacją poprawności obu wprowadzonych haseł. Hasła użytkowników zapisywany są do bazy danych zahaszowane za pomocą pakietu bcrypt.

3.Opis działania

Aplikacja uruchamia się lokalnie na porcie 3000 => http://localhost:3000.

- **/register => endpoint ten umożliwia rejestrację użytkownika poprzez przesłanie w ciele żądania obiektu User czyli nowego użytkownika,który zostaje dodany do bazy danych a na podany adres zostaje wysłany email aktywacyjny z tokenem dostępowy ważnym 24h.**

- **/registerconfirm => endpoint umożliwający aktywowanie zarejestrowanego konta użytkownika,link do żądania razem z tokenem dostępowym ważnym 24h został wcześniej wysłany na adres użytkownika. Po weryfikacji tokenu użytkownik otrzymuje status aktywny w aplikacji i może się zalogować.**
- **/resetpassword => endpoint umożliwiający odzyskania hasła użytkownika po podaniu jego adres email. Zakładam że w takim wypadku po stronie front-end znajdował by się komponent który po pozytywnej weryfikacji, że email istnieje od razu pokazałby komponent do zmiany hasła. Ten endpoint zwraca token który te operację umożliwia.**
- **/resetconfirm => endpoint zarówno dla zalogowanego jak i odzyskujące hasło użytkownika który po weryfikacji poprawności tokenu przyjmuje nowe hasło i jego potwierdzenie. Sprawdza ich identyczność i jeżeli są takie same generuje hash hasła i zapisuje go do bazy danych.**
- **/login => endpoint umożliwiający logowanie użytkownikom po podaniu loginu i hasła. Jeżeli weryfikacja przebiegła pomyślnie (hash podanego hasła i hash w bazie danych się zgadzają) zwracany jest token dostępowy na okres 1h który posłuży do wykonywania operacji CRUD za pomocą innych zabezpieczonych endpointów.**
- **/get/books => endpoint typu GET pobiera listę książek przefiltrowaną z użyciem opcjonalnych filtrów przekazanych w parametrach żądania. Zabezpieczony jest poprzez middleware weryfikujący najpierw poprawność przesłanego tokenu dostępowego podobnie jak inne endpointy wykonujące zapytania CRUD.**
- **/insert/books => endpoint typu POST, umożliwia dodanie nowej książki do bazy danych expressdb do kolekcji Books. Przesłany obiekt książki znajduje się w ciele żądania.**
- **/update/book/:id => endpoint typu PUT umożliwiający update wybranych właściwości w obiekcie wybranej książki (Book). W zapytaniu należy podać nr identyfikacyjny książki w bazie danych a w ciele żądania obiekt z listą zmian jakich należy dokonać w wybranym obiekcie.**
- **/delete/book/:id => endpoint typu DELETE, umożliwia usunięcie wybranego obiektu Book w bazie danych. W żądaniu należy podać nr identyfikacyjny obiektu z bazy danych.**