**Backend/student\_assignment Setup instructions**

**Overview**

This project is an Assignment Management System built using Node.js, Express.js, MySQL, and JWT for authentication. The system provides endpoints for user authentication, assignment creation, retrieval, update, deletion, and submission. The project follows the MVC (Model-View-Controller) architecture, ensuring a clear separation of concerns and maintainability.

**Project Structure**

root/

├── controllers/

│ ├── authController.js

│ ├── assignmentController.js

├── database/

│ ├── db.js

├── middlewares/

│ ├── auth.js

├── models/

│ ├── assignmentModel.js

├── routes/

│ ├── routes.js

├── .env

├── Dockerfile

├── initializeDatabaseAndTables.sql

├── index.js

├── package.json

**Setup Instructions**

**Prerequisites**

Ensure you have the following software installed:

Node.js (v12.x or later)

MySQL (v5.7 or later)

**Step 1: Clone the Repository**

**Clone the project repository to your local machine using the following command:**

git clone https://github.com/your-username/assignment-management-system.git

cd assignment-management-system

**Step 2: Install Dependencies**

Install the required Node.js dependencies by running:

npm install

**Step 3: Configure Environment Variables**

Create a .env file in the root directory of the project and add the following environment variables:

PORT=3000

DB\_HOST=localhost

DB\_USER=root

DB\_PASSWORD=root

DB\_NAME=student\_assignments

SECRET\_KEY=djo98t\*tu4\*%&jfdlj$jdlgkjdfg8u9485udjgkdjgljkg4578438\*dfhjkgsfh784eyt4\*@^$tgyufty

Note: These variables configure the server port, database connection details, and the secret key for JWT authentication.

**Step 4: Initialize the Database**

Start your MySQL server and create the necessary database and tables by executing the SQL script provided. You can use a MySQL client or the command line:

mysql -u root -p < database/initializeDatabaseAndTables.sql

This command will prompt you for the MySQL root password and then execute the SQL script to set up the database and tables.

**Step 5: Start the Server**

Start the Node.js server by running:

node index.js or npm run dev

The server will start on the port specified in the .env file (default is 3000).

**Step 6: Test the API**

You can now test the API endpoints using tools like Postman or cURL. Ensure that you include the JWT token in the Authorization header for all authenticated routes except of /login route. For testing all enpoint we have postman file that can executes directly and will open all the routes in the postman. For understating how wll api endpoints works please refer to endpoint documentation file.

**Available Scripts**

In the project directory, you can run:

**npm start:** Starts the server in production mode.

**npm run dev:** Starts the server in development mode using nodemon for automatic restarts on file changes.

Contact

For any questions or issues regarding the setup or usage of the project, please contact the project developer at [ataul.computer786@gmail.com](mailto:ataul.computer786@gmail.com) or phone: 7983732026