Full Stack Development with MERN

API Development and Integration Report

Date	10th July 2024
Team ID	SWTID1720126010
Project Name	SBFoods
Maximum Marks	6 Marks

Project Title: SB Foods
Date: 10th July 2024
Prepared by: Yash Tekwani, Sarthak Ray, Kripa Verma, Pavan Gunupati
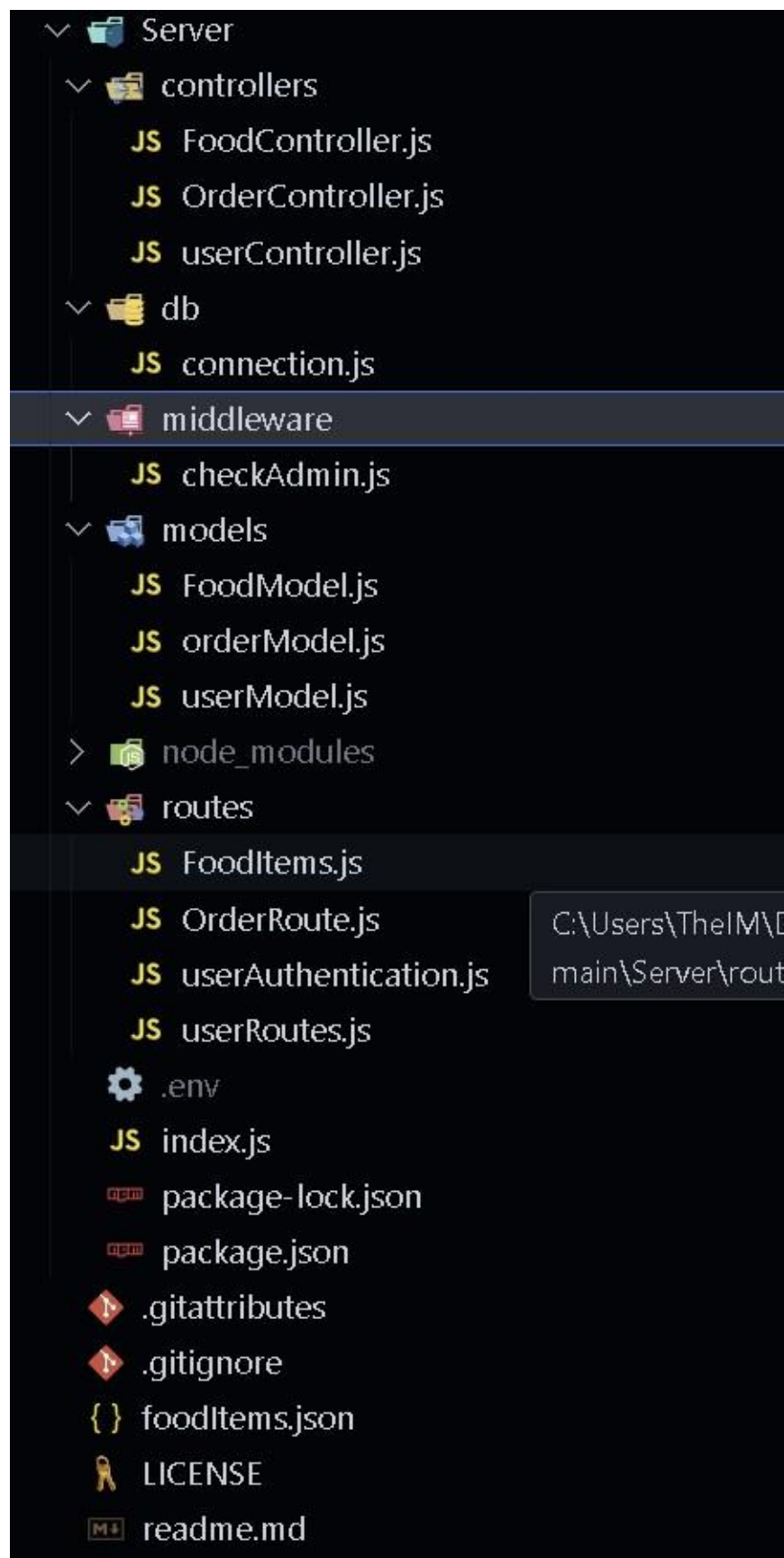
Objective

The objective of this report is to document the API development progress and key aspects of the backend services implementation for the SB Foods - FoodDelivery project.

Technologies Used

- **Backend Framework:** Node.js with Express.js
- **Database:** MongoDB • **Authentication:** [JWT]

Project Structure

Provide a screenshot of the backend project structure with explanations for key directories and files.

```
∨ 🗃 Server
  ∨ 🗃 controllers
      JS FoodController.js
      JS OrderController.js
      JS userController.js
  ∨ 🗃 db
      JS connection.js
  ∨ 🗃 middleware
      JS checkAdmin.js
  ∨ 🗃 models
      JS FoodModel.js
      JS orderModel.js
      JS userModel.js
  > 🗃 node_modules
  ∨ 🗃 routes
      JS FoodItems.js
      JS OrderRoute.js                C:\Users\TheIM\[
      JS userAuthentication.js        main\Server\rout
      JS userRoutes.js
    ⚙ .env
    JS index.js
    📦 package-lock.json
    📦 package.json
  ◈ .gitattributes
  ◈ .gitignore
  { } foodItems.json
  🔑 LICENSE
  Ⓜ readme.md
```

Key Directories and Files

1. **/controllers** ○ Contains functions to handle requests and responses.
2. **/models** ○ Includes Mongoose schemas and models for MongoDB collections.
3. **/routes** ○ Defines the API endpoints and links them to controller functions.
4. **/middlewares** ○ Custom middleware functions for request processing.

API Endpoints
A summary of the main API endpoints and their purposes:

User Authentication

• **POST /api/UserAuth/SignUp** - Registers a new user.
• **POST /api/UserAuth/SignIn** - Authenticates a user and returns a token.

User Management

• **GET /api/checkAdmin –** Checks if the current user is Admin via their JWT Token •
PUT /api/findUserByToken – Finds the Details of the User by their JWT Token.

Food Management

• **GET /api/Food/foodItems** – Retrieves the list of all the available food item
• **PATCH /api/Food/foodItems/:id** – Checks if the user is Admin if so then updates the data provided in the response
• **POST /api/Food/foodItems/addItem** – Checks if the user is Admin if so then add the provided Food Item to the menu

Order Management

• **POST /api/Order/AddOrder** – Adds the order to the orderList that's visible on admin panel
• **POST /api/Order/GetOrder** – Gets the list of all the pending Orders
• **POST /api/Order/GetOrderById** – Get the specific Order Details via it's Order Id •
POST /api/Order/deleteOrder – Deletes the order details once finished or Cancelled.

Integration with Frontend
The backend communicates with the frontend via RESTful APIs. Key points of integration include:

- **User Authentication:** Tokens are passed between frontend and backend to handle authentication.
- **Data Fetching:** Frontend components make API calls to fetch necessary data for display and interaction.

Error Handling and Validation

Describe the error handling strategy and validation mechanisms:

- **Error Handling:** Centralized error handling using middleware.

Security Considerations

Outline the security measures implemented:

- **Authentication:** Secure token-based authentication.
- **Data Encryption:** Encrypt sensitive data (passwords) at rest and in transit.