Full Stack Development with MERN

API Development and Integration Report

Date	7 July 2024
Team ID	SWTID1720171853
Project Name	Project - Food Ordering System
Maximum Marks	

Project Title: Food Ordering System
Date: 7 July 2024
Prepared by: Buvaneswaran A S

Objective

The objective of this report is to document the API development progress and key aspects of the backend services implementation for the [Your Project Title] project.

Technologies Used

- **Backend Framework:** Node.js with Express.js .
- **Database:** MongoDB .
- **Authentication:** bcrpet ,body-parser ,JWT.

Project Structure

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

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.
5. **/config**
   - Configuration files for database connections, environment variables, etc.

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

User Authentication

POST /api/user/register - Registers a new user.

POST /api/user/login - Authenticates a user and returns a token.

User Management

GET /api/user/- Retrieves user information by ID.

PUT /api/user/- Updates user information by ID.

Restaurant Management

POST /api/update-promote-list - Updates the promoted restaurant list.

POST /api/approve-user - Approves a restaurant user.

POST /api/reject-user - Rejects a restaurant user.

GET /api/fetch-restaurants - Fetches all restaurants.

GET /api/fetch- restaurant-details/- Fetches restaurant details by owner ID.

GET /api/fetch-restaurant/- Fetches restaurant details by restaurant ID.

Order Management

GET /api/fetch-orders - Fetches all orders.

PUT /api/cancel-order - Cancels an order.

PUT /api/update-order-status - Updates the status of an order.

POST /api/place-cart-order - Places an order from the cart.

Food Item Management

GET /api/fetch-items - Fetches all food items.

GET /api/fetch-item-details/- Fetches details of a food item by ID.

POST /api/add-new-product - Adds a new product (food item).

PUT /api/update-product/- Updates a product (food item) by ID.

Cart Management

GET /api/fetch-cart - Fetches all cart items.

POST /api/add-to-cart - Adds an item to the cart.

PUT /api/remove-item - Removes an item from the cart.

Administrative Management

GET /api/fetch-categories - Fetches all categories.

GET /api/fetch-promoted-list - Fetches the promoted restaurant list.

User Details

GET /api/fetch-user-details/- Fetches details of a user by ID.

GET /api/fetch-users - Fetches all users.

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:** Error handling using Try catch blocks.
- **Validation:** Input validation using libraries like JWT web token or express-validator.

Security Considerations

Outline the security measures implemented:

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