Full Stack Development with MERN

Project Documentation

1. Introduction

- **Project Title:** OrderOnTheGo: Your On-Demand Food Ordering Solution
- **Team ID :** LTVIP2025TMID55780
- **Team Members:**
    - Madhu Kanth – Full Stack Developer (React, Node.js, MongoDB, Express.js)

2. Project Overview

- **Purpose:**
  FoodMunch is a fully functional and responsive food ordering website that enables users to register, log in, and browse various food menu items. The goal is to provide a seamless experience for both desktop and mobile users with secure authentication and dynamic data handling.
- **Features:**
    - User Registration & Login with JWT
    - Responsive and modern UI using Bootstrap
    - Reusable components in React
    - Dynamic menu items from JSON
    - MongoDB integration for persistent storage
    - Password hashing and validation for security

3. Architecture

- **Frontend (React):**
    - Built using React 18
    - Component-based architecture with Hooks
    - React Router for client-side navigation
    - Axios for API calls
    - Responsive layout using Bootstrap 4
- **Backend (Node.js & Express.js):**
    - RESTful API with Express.js
    - JWT for authentication
    - bcryptjs for password encryption
    - Input validation with express-validator
- **Database (MongoDB):**

- Uses Mongoose ODM
- One primary collection: Users
- Fields include: username, email, password, createdAt

4. Setup Instructions

- **Prerequisites:**
  - Node.js (v14 or higher)
  - MongoDB (v4.4 or higher)
  - npm (comes with Node.js)
- **Installation:**

1. **Clone the Repository**

```
git clone https://github.com/madhukodali23/FoodMunch
```

Install Backend Dependencies

```
cd foodmunch/server
npm install
```

2. **Install Frontend Dependencies**

```
cd ../client
npm install
```

3. **Environment Variables (Server/.env)**

```
PORT=5000
JWT_SECRET=your-secret-key-here
MONGODB_URI=mongodb://localhost:27017/foodmunch
```

5. Folder Structure

- **Client (React Frontend):**

```
client/
 ├── public/
 │   └── index.html
 ├── src/
 │   ├── components/
 │   │   ├── Navbar.js
 │   │   ├── Home.js
 │   │   ├── Login.js
 │   │   ├── Register.js
 │   │   ├── MenuItem.js
 │   │   └── WcuCard.js
 │   ├── data/
 │   │   └── menuItems.js
 │   ├── App.js
 │   ├── index.js
 │   └── index.css
 └── package.json
```

- **Server (Node.js Backend):**

  pgsql

  CopyEdit

  ```
  server/
  ├── server.js
  └── package.json
  ```

6. Running the Application

- **Start MongoDB**

  ```
  mongod
  ```

- **Start Backend Server**

  ```
  cd server
  npm start
  ```

- **Start Frontend React App**

  ```
  cd client
  npm start
  ```

- **Localhost Ports:**
  - **Backend: http://localhost:5000**
  - **Frontend: http://localhost:3000**

7. API Documentation

- **Authentication Routes:**

Method	**Endpoint**	**Description**
POST	**/api/register**	**Register a new user**

Method	Endpoint	Description
POST	**/api/login**	**Login existing user**
GET	**/api/profile**	**Get user profile (JWT required)**

- **Example:**

```
// POST /api/login Request Body
{
  "email": "user@example.com",
  "password": "mypassword"
}

// Response
{
  "token": "jwt_token_here"
}
```

8. Authentication

- **Password Hashing:** Using bcryptjs before storing in MongoDB.
- **JWT Tokens:** Generated during login and sent to the client.
- **Protected Routes:** Middleware verifies token and grants access to private endpoints.
- **Auth State:** Stored in localStorage on the frontend for session management.

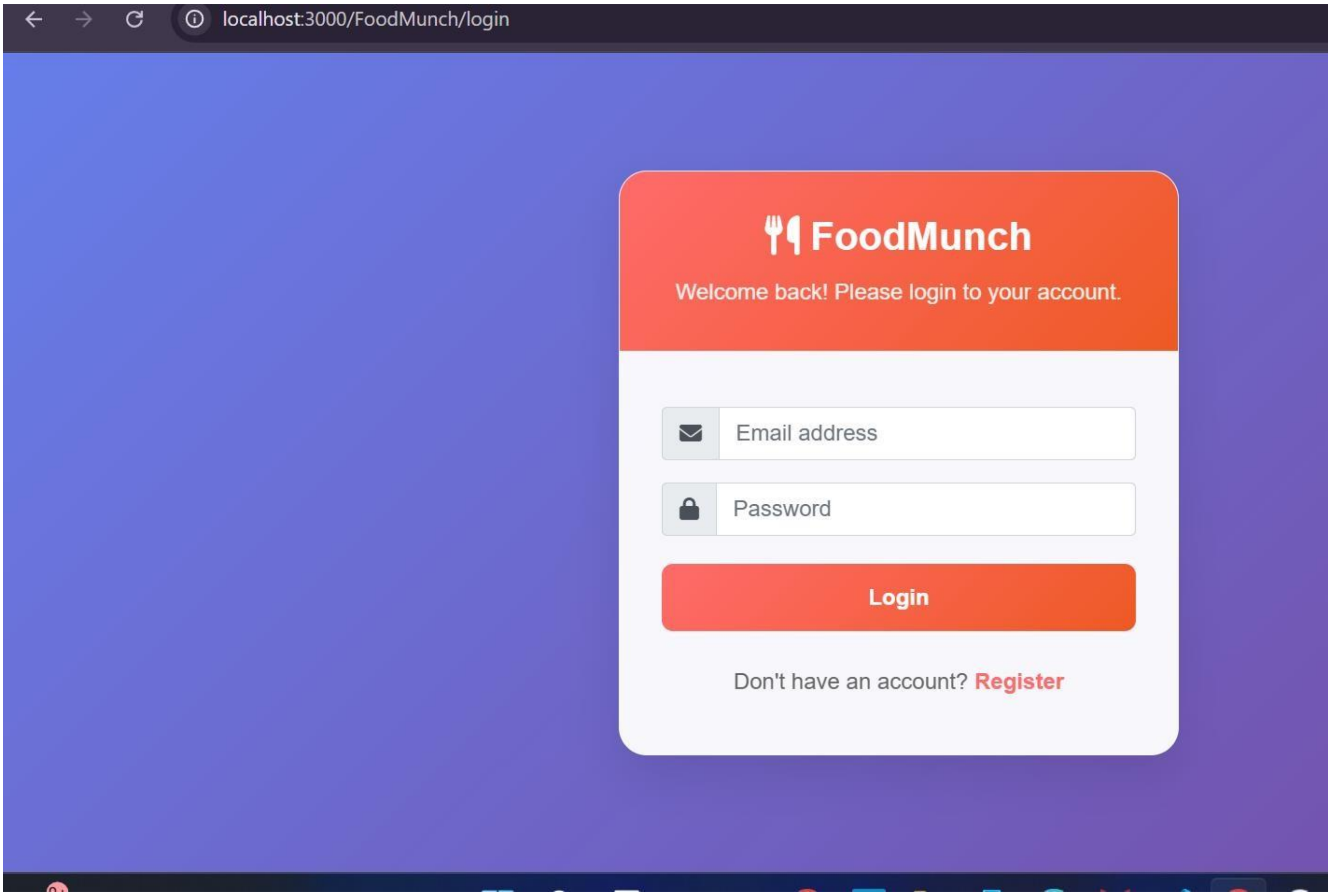
9. User Interface

- **Built With:**
  - Bootstrap 4
  - Font Awesome Icons
  - CSS Animations
- **Key Pages:**
  - **Home Page:** Menu categories, hero section, and call-to-action.
  - **Login/Register:** Forms with validation and API connection.
  - **Navbar:** Changes based on auth state.
  - **Menu Items:** Rendered dynamically using menuItems.js

10. Testing

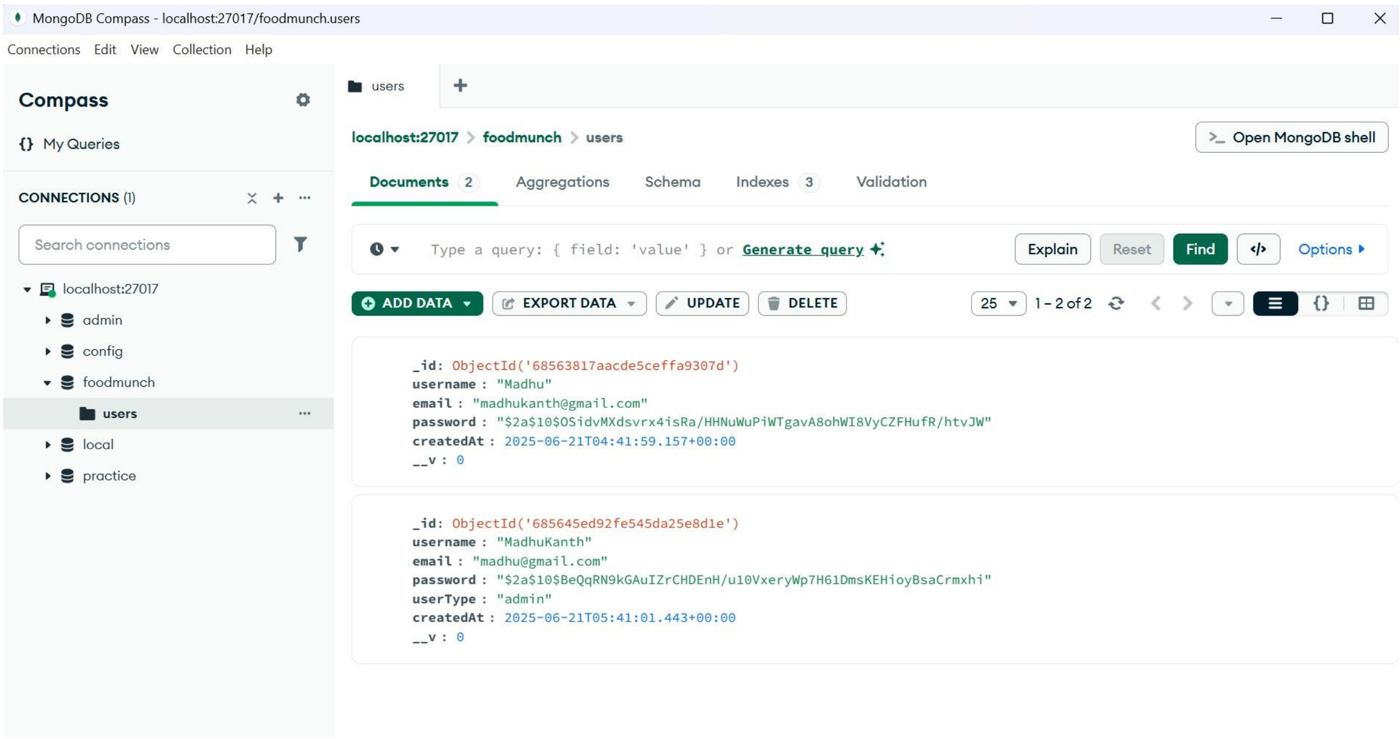
- **Manual Testing:** Basic form validation, login/logout flow, and MongoDB integration.
- **Tools:** Postman for API testing, React Developer Tools for component debugging.

11. Screenshots or Demo



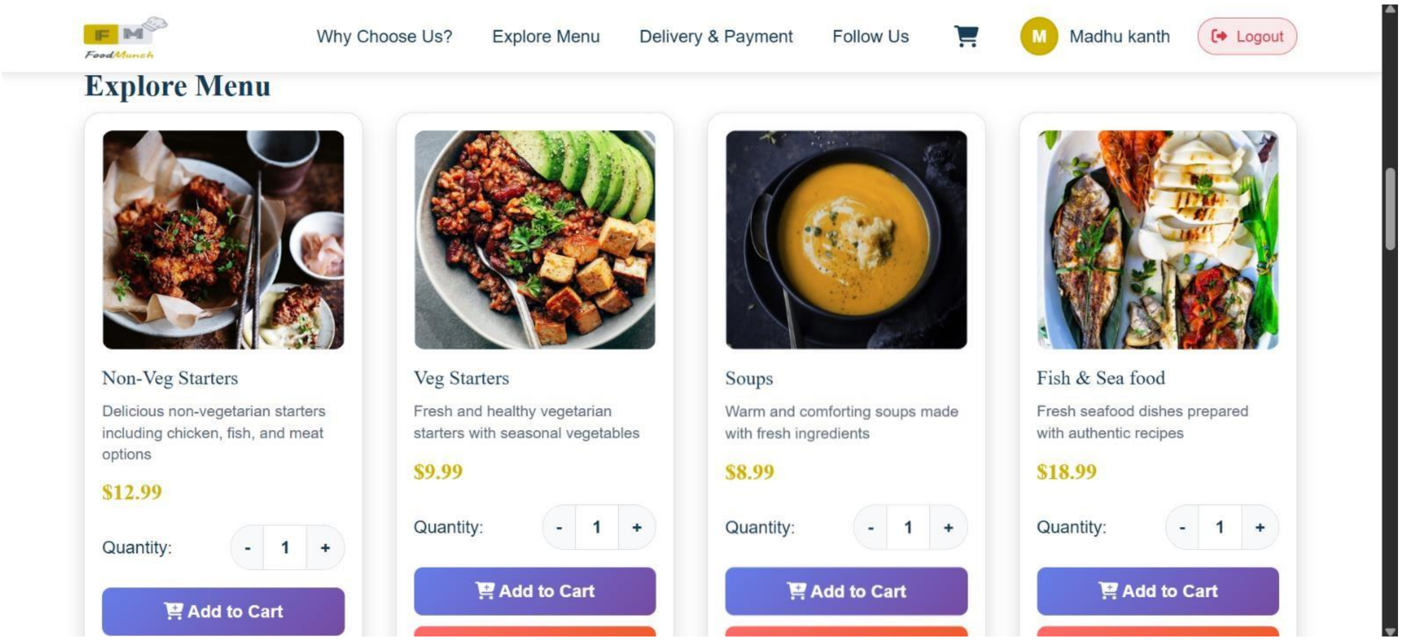
Welcome back! Please login to your account.

Don't have an account? Register



```
_id: ObjectId('68563817aacde5ceffa9307d')
username : "Madhu"
email : "madhukanth@gmail.com"
password : "$2a$10$OSidvMXdsvrx4isRa/HHNuWuPiWTgavA8ohWI8VyCZFHufR/htvJW"
createdAt : 2025-06-21T04:41:59.157+00:00
__v : 0
```

```
_id: ObjectId('685645ed92fe545da25e8d1e')
username : "MadhuKanth"
email : "madhu@gmail.com"
password : "$2a$10$BeQqRN9kGAuIZrCHDEnH/u10VxeryWp7H61DmsKEHioyBsaCrmxhi"
userType : "admin"
createdAt : 2025-06-21T05:41:01.443+00:00
__v : 0
```



Explore Menu

Non-Veg Starters

Delicious non-vegetarian starters including chicken, fish, and meat options

$12.99

Quantity: - 1 +

Add to Cart

Veg Starters

Fresh and healthy vegetarian starters with seasonal vegetables

$9.99

Quantity: - 1 +

Add to Cart

Soups

Warm and comforting soups made with fresh ingredients

$8.99

Quantity: - 1 +

Add to Cart

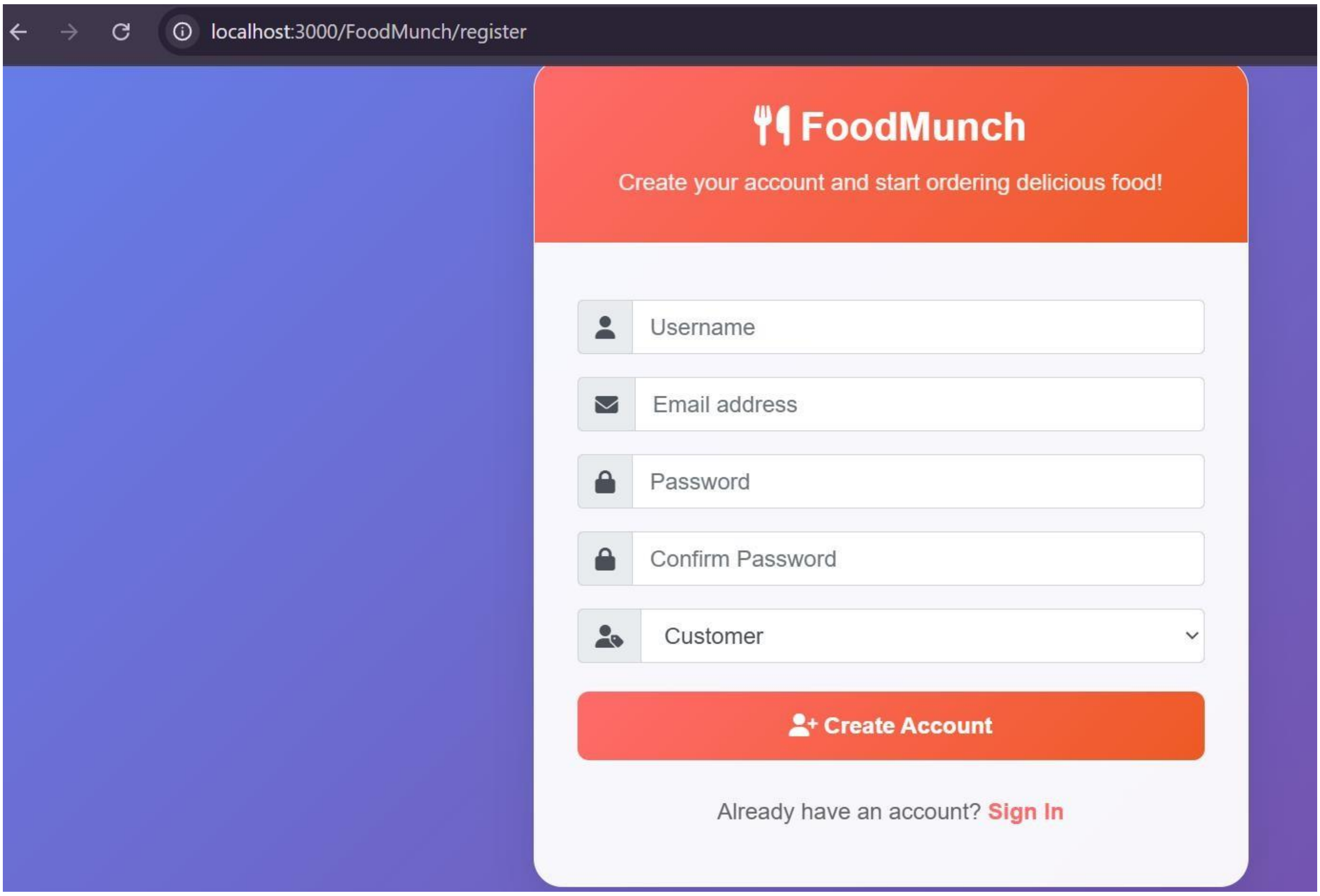
Fish & Sea food

Fresh seafood dishes prepared with authentic recipes

$18.99

Quantity: - 1 +

Add to Cart



localhost:3000/FoodMunch/register

FoodMunch

Create your account and start ordering delicious food!

Username

Email address

Password

Confirm Password

Customer

Create Account

Already have an account? Sign In

12. Known Issues

- No admin panel for adding/updating menu items
- No role-based access control (admin vs. user)
- Limited form validation on frontend
- Minimal error feedback on failed API calls

13. Future Enhancements

- Admin dashboard to manage menu items
- Integration with payment gateways (Razorpay/Stripe)
- Improved error handling and user feedback
- Add unit and integration tests (Jest, Supertest)
- Deployment on cloud platforms (Vercel + Render/Heroku)