Food Ordering Platform Documentation

By Aditya Prashant Inamdar.

Overview

This project is a simple food ordering platform that allows users to register, browse available food items, add them to a cart, and place an order. The platform is built using **Node.js** for the backend, **MongoDB** for database management, and **EJS** for rendering dynamic HTML templates. Users can browse items by categories, view their cart, and check out with an order.

System Design

1. Application Architecture

- **Frontend**: HTML, CSS, JavaScript (with EJS for templating)
- **Backend**: Node.js with Express for routing and business logic
- **Database**: MongoDB, using Mongoose for object modelling.
- **Authentication**: Managed via JWT's for persisting authentication info across sessions. .

2. Features

1. **User Registration and Authentication**
   - Users can register using an email and password.
   - User authentication is handled with sessions; once logged in, the session is persisted across pages.
2. **Browsing Food Items**
   - Items are categorized into types such as Fruits, Vegetables, Non-veg, Breads, etc.
   - Users can browse and filter items by these categories.
3. **Shopping Cart**
   - Users can add multiple items to their cart with specified quantities.
   - Items in the cart are persisted, and users can see their cart history across sessions.
   - Users can adjust quantities of items in their cart.
4. **Checkout**
   - Users can view a summary of their selected items before proceeding to checkout.
   - Upon successful checkout, the stock is updated, and users receive an order confirmation.
5. **Order Management**
   - Users can view their order history.
   - Order delivery status is tracked and visible.

3. Database Schema

- **User Schema**

```
 User = ({
  name: String,
  email: String,
  password: String,
  cart: [{
    item: Item
    quantity: Number
  }],
  orders: [{
    items: [Item]
      quantity: Number
    }],
    orderDate: { type: Date, default: Date.now },
    status: { type: String, default: 'Pending' }
  }]
});
```

- **Item Schema**

```
Item = ({
  name: String,
  category: String,
  stock: Number,
  price: Number
});
```

4. Routes

Authentication Routes

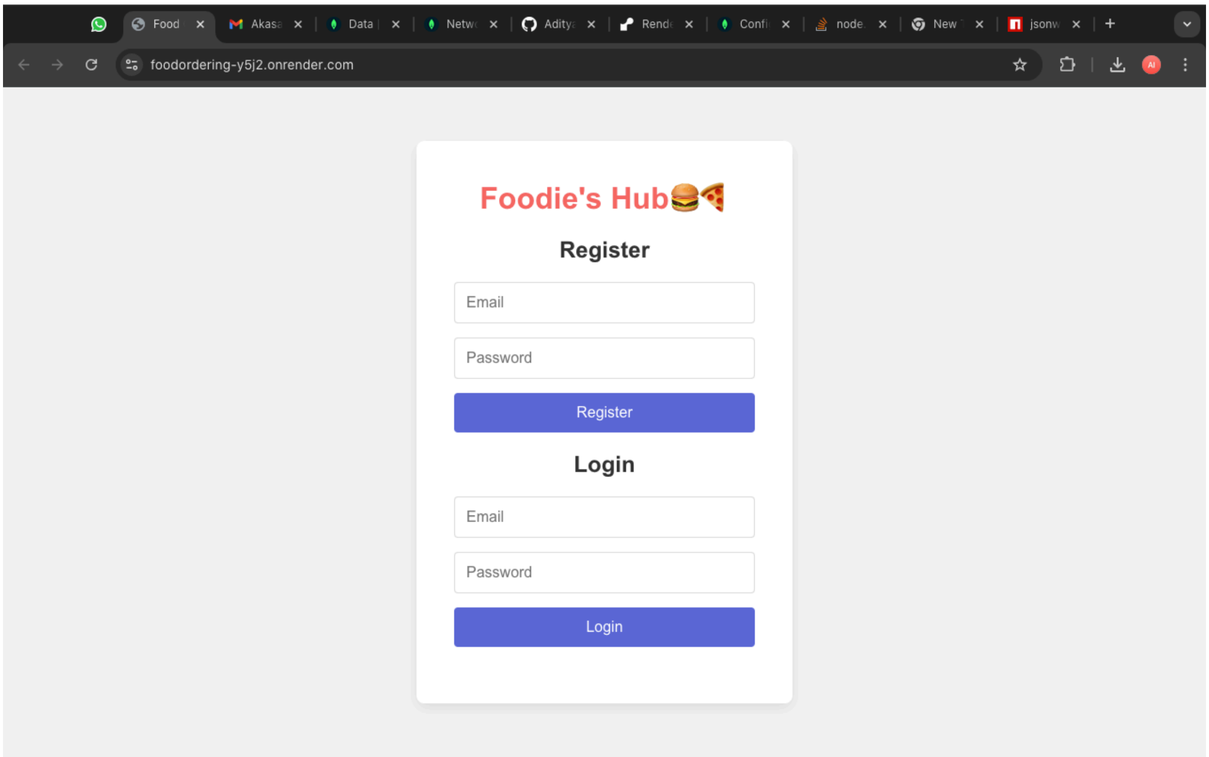
- /register (GET/POST): Registers new users.
- /login (GET/POST): Authenticates users and starts a session.

Browse & Cart Routes

- /items/:category (GET): Displays items based on category selection.
- /cart (GET): Displays items in the user’s cart.
- /cart (POST): Adds an item to the user’s cart.
- /checkout (GET/POST): Displays the checkout page and processes the order.

Conclusion

This food ordering platform serves as a basic example of a shopping cart system with user authentication, cart management, and checkout functionality using Node.js and MongoDB. The modular structure and use of Mongoose for database interaction make it easy to extend this application with more features in the future.



foodordering-y5j2.onrender.com

Foodie's Hub🍔🍕

Register

Email

Password

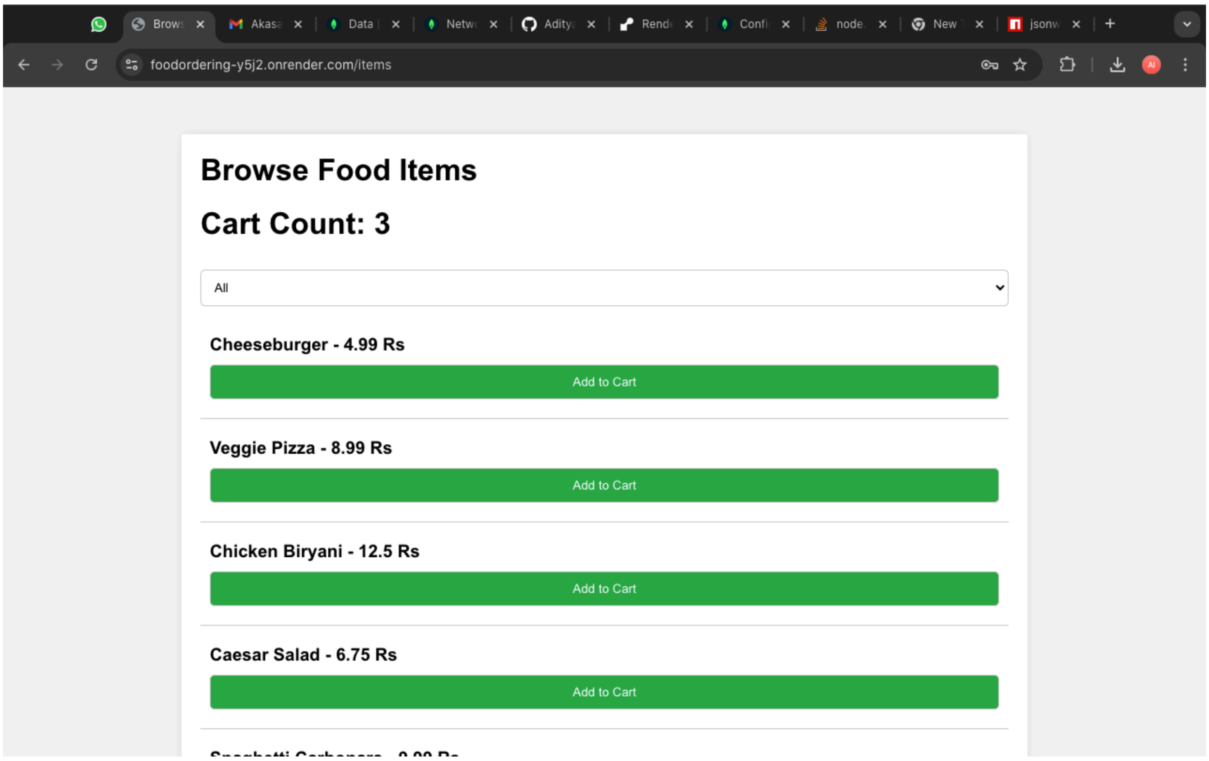
Register

Login

Email

Password

Login



foodordering-y5j2.onrender.com/items

Browse Food Items

Cart Count: 3

All

Cheeseburger - 4.99 Rs

Add to Cart

Veggie Pizza - 8.99 Rs

Add to Cart

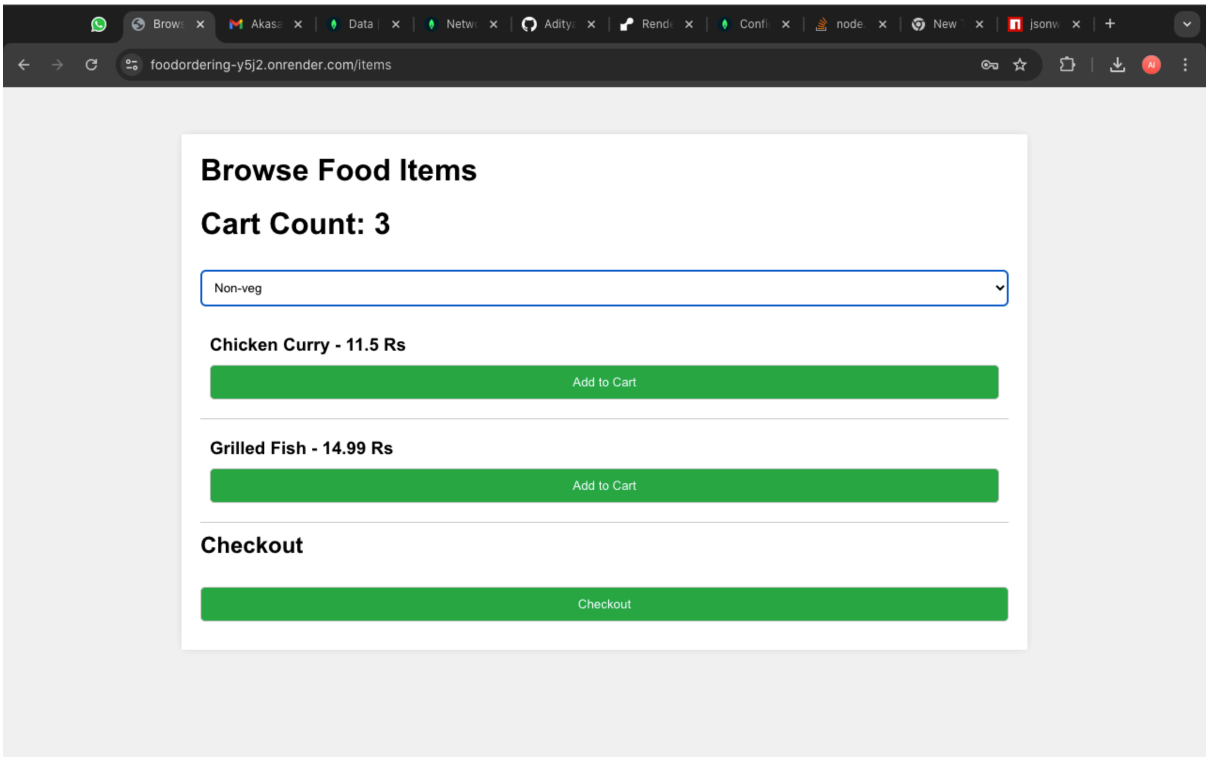
Chicken Biryani - 12.5 Rs

Add to Cart

Caesar Salad - 6.75 Rs

Add to Cart

Spaghetti Carbonara - 9.99 Rs



Browse Food Items

Cart Count: 3

Non-veg

Chicken Curry - 11.5 Rs

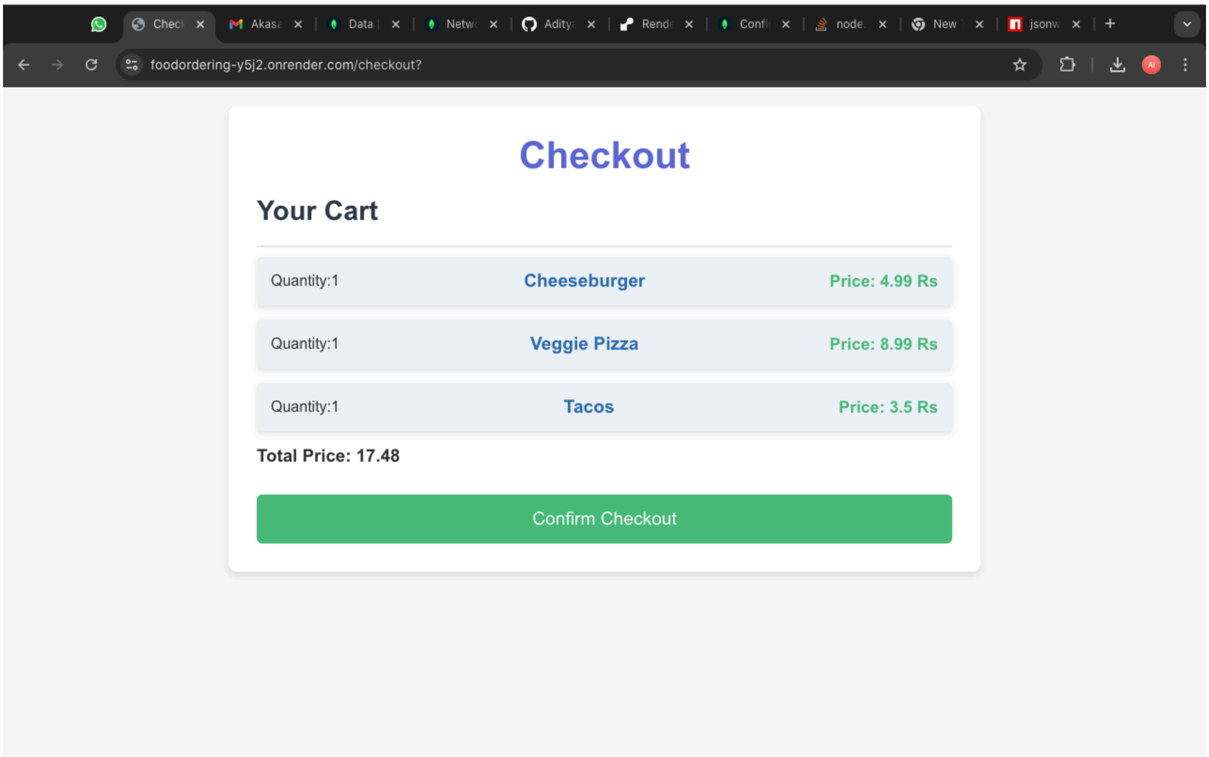
Add to Cart

Grilled Fish - 14.99 Rs

Add to Cart

Checkout

Checkout



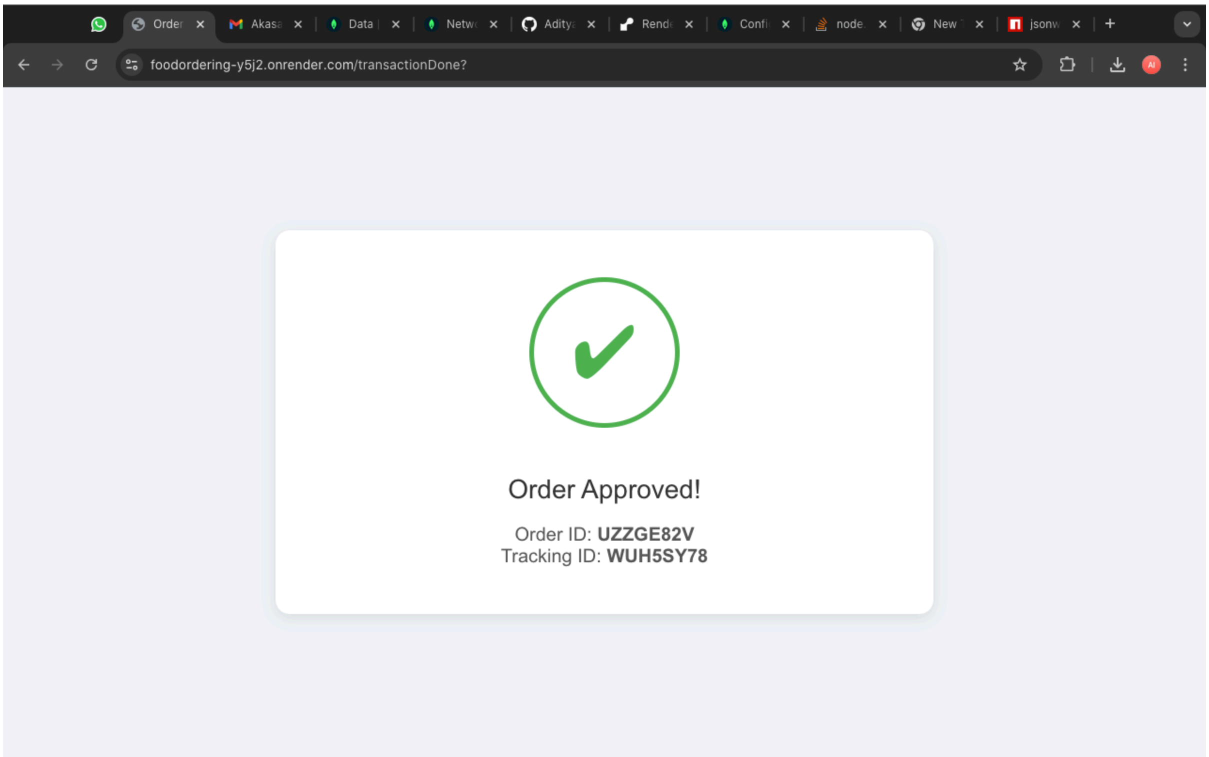
Checkout

Your Cart

Quantity:1	Cheeseburger	Price: 4.99 Rs
Quantity:1	Veggie Pizza	Price: 8.99 Rs
Quantity:1	Tacos	Price: 3.5 Rs

Total Price: 17.48

Confirm Checkout



Order Approved!

Order ID: **UZZGE82V**
Tracking ID: **WUH5SY78**