SYNOPSIS

ON

FOOD DELIVERY WEBSITE

Submitted By:

Anand Pratap Singh (2115000143)
Riya Saxena (2115000863)
Vaishnavi Sharma (2115001086)

Submitted To:

Mr. Sanjay Madaan
(CEA Department)
Technical Trainer

Title of the Project:

Foodalix - Simplifying Food Delivery

Objective:

The main objective of our project is to develop a user-friendly food delivery website that streamlines the process of ordering food online. By integrating the MERN stack, we aim to provide a seamless experience for users to browse menus, place orders, and track their order history. Additionally, the platform will include a checkout option to clear the cart once the order is delivered.

Scope

Our project will encompass the creation of a robust food delivery platform with user authentication (login and sign up), address retrieval from an open API, a cart portal powered by useContext and useReducer for efficient state management, and an order history section. The checkout option will be implemented to clear the cart once the user had added all its required food items.

Methodology:

We plan to utilize the MERN stack, consisting of MongoDB for database management, Express.js for server-side logic, React.js for frontend development, and Node.js for backend development. User authentication will be implemented using JWT tokens, and we'll integrate an open API for address retrieval. The cart portal will leverage the useContext and useReducer hooks provided by React for state management.

Proposed System:

The proposed system will offer a user-friendly interface where customers can create accounts, browse restaurant menus, add items to their cart, and place orders. Upon login, users' addresses will be automatically retrieved using an open API, simplifying the ordering process. The cart portal will efficiently manage the state of the user's cart using useContext and useReducer hooks, ensuring a smooth shopping experience. Additionally, users will have access to an order history section where they can view details of their previous orders. The checkout option will clear the cart once the order is delivered successfully.

Features:

- User authentication (login and sign up).
- Address retrieval from an open API.
- Cart portal with useContext and useReducer for state management.
- Order history section for users to view past orders.
- Checkout option to clear the cart after successful delivery.

Implementation Plan:

1. **Setting up the MERN stack:**
- First, we'll set up our database using MongoDB.
- Then, we'll create the backend of our website using Express.js.
- After that, we'll build the frontend using React.js.
- Lastly, we'll connect everything together using Node.js.
2. **Adding user authentication:**
- We'll make forms for users to sign up and log in.
- Next, we'll create a way for the server to check if the user's credentials are correct.
- Then, we'll use JWT tokens to keep track of whether a user is logged in or not.
3. **Integrating address retrieval:**
- We'll find an API that can give us addresses based on a user's location.
- We'll set up the server to talk to this API and get the addresses when needed.
- Finally, we'll display the addresses on the website for the user to see.
4. **Building the cart portal:**
- We'll design a cart where users can add items they want to buy.
- Using React's hooks, we'll keep track of what's in the cart.
- We'll also make sure that the cart works well with the backend so that users can see what they've added.
5. **Creating the order history section:**
- We'll make a section where users can see their past orders.
- We'll set up the server to send this information to the frontend when asked.
- Then, we'll display the orders in an easy-to-read format for the user.
6. **Testing and fixing any issues:**
- We'll test each part of the website to make sure it works as expected.
- If we find any problems, we'll fix them to make the website run smoothly.

Future Expansion:

In our continuous effort to enhance user experience and provide more convenience, we plan to integrate a payment gateway API following the checkout process. This expansion will streamline the payment process for our customers, allowing for secure and seamless transactions directly on our platform.

Team Members:

Anand Pratap Singh: Backend developer

Riya Saxena: Database Administrator

Vaishnavi Sharma: Frontend developer

Resources Required:

- MongoDB for database management.
- Express.js for server-side logic.
- React.js for frontend development.
- Node.js for backend development.
- Open API for address retrieval.

References:

We'll use online tutorials and guides to help us with coding and design. Some of the websites we might use include:

- MongoDB Documentation: https://docs.mongodb.com/
- React Documentation: https://reactjs.org/docs/getting-started.html
- Node.js Documentation: https://nodejs.org/en/docs/
- Express.js Documentation: https://expressjs.com/
- useContext and useReducer Hooks: https://reactjs.org/docs/hooks-reference.html#usecontext
- JWT Authentication: https://jwt.io/introduction/
- Open API Documentation (for address retrieval)

Expected Outcomes:

By the end of our project, we hope to have a fully functional website where people can easily order food online. Users should be able to sign up, log in, add items to their cart, place orders, and view their past orders and the cart should automatically clear after checkout once the order is delivered.

Project Supervisor:

Mr. Sanjay Madaan

Conclusion:

Our project aims to provide a seamless and efficient food ordering experience for users by leveraging the capabilities of the MERN stack. Through user authentication, address retrieval, and an intuitive cart portal, we strive to create a platform that simplifies the process of ordering food online while ensuring a smooth and enjoyable user experience.