Nancy Kibunga

Food Ordering Web Application Approach Method

Tools and Languages:

- React JS
- Node JS
- Express js
- Mongo DB

Planning and Requirements Gathering:

Identifying the Target Audience and Needs

- Identified the target audience and their requirements.

Determining Key Features

- Determined the key features of the application, including user registration, food browsing, ordering, payment processing, and order tracking.

Setting Up the Development Environment

- Installed necessary tools and dependencies, including Node.js, npm, and a code editor.

- Initialized a new React project using Create React App.

- Set up a backend server using Node.js and Express.

- Created a MongoDB database for storing user information, food items, orders, etc.

Frontend Development

Designed the application's UI/UX, which involved creating wireframes and mockups.

Implemented various components and pages using React, including Home, Login/Register, Food Listing, Food Details, Cart, Checkout, Profile, and Orders.

Utilized React Router to enable seamless navigation between different pages.

Integrated Axios to facilitate HTTP requests to the backend server.

Implemented authentication and authorization features, covering user registration, login/logout functionality, and secure access to protected routes.

Backend Development

- Defined RESTful APIs to manage CRUD operations related to user accounts, food items, and orders.

- Implemented the APIs using Express routes.

- Established a connection to the MongoDB database using Mongoose to execute database operations.

- Integrated features such as authentication and authorization using JWT (JSON Web Tokens).

- Ensured robust error handling and validation of incoming data for enhanced reliability and security.

Database Setup

- Defined schemas for user accounts, food items, orders, etc., using Mongoose.

- Set up database seeding scripts to populate initial data if required.

Payment Integration

- Integrated a payment gateway (e.g., PayPal, Stripe) for processing orders securely.

- Implemented the necessary backend APIs for handling payment requests and callbacks.

Order Tracking

- Implemented features to allow users to track the status of their orders.

- Used real-time updates or polling mechanisms to keep users informed about order status changes.

Testing

- Performed unit tests and integration tests for both frontend and backend components.

- Conducted user acceptance testing (UAT) to ensure that the application meets the requirements and works as expected.

Deployment (Ongoing)

- Deploying the frontend and backend applications to a hosting provider (e.g., Heroku, AWS, DigitalOcean).

- Configuring domain settings and setting up SSL certificates for secure communication.

Maintenance and Support (Ongoing)

- Regularly updating the application with new features and bug fixes based on user feedback.

- Monitoring server logs, database performance, and security vulnerabilities.

- Providing customer support and addressing user inquiries and complaints promptly.