Food Delivery Application (MERN Stack) - Project Overview

Step 1: Planning Stage

We decided to build a Food Delivery Application using the MERN stack: MongoDB, Express.js, React.js, and Node.js.

Step 2: Setting up the Development Environment

Installed Node.js and Visual Studio Code. Created a main project folder named 'food-delivery-application'.

Step 3: Building the Frontend (User Interface)

Created a 'frontend' folder. Used 'npx create-react-app frontend'. Installed web-vitals, axios, and react-router-dom. Designed the user interface, created components and pages, and uploaded the frontend code to GitHub.

Step 4: Building the Backend (Server & APIs)

Created a 'backend' folder and a 'server.js' file. Installed express, mongoose, cors, dotenv, bcryptjs, jsonwebtoken, and nodemon. Set up the server to run and created a .env file for environment variables.

Step 5: Setting up the Database (MongoDB Atlas)

Created a project and cluster in MongoDB Atlas. Allowed all IP addresses for network access. Stored the database URI securely in the .env file.

Step 6: Testing the API

Used API testing tools like Postman or Thunder Client to test user registration, login, and menu fetching APIs.

Step 7: Deploying the Project (Going Live!)

Backend deployed on Render by connecting GitHub repository. Frontend deployed on Vercel. Both live and ready for users.

Step 8: Final Live Application Flow

Users interact with the frontend (Vercel), frontend talks to backend APIs (Render), backend fetches/stores data in MongoDB Atlas.

Food Delivery Application (MERN Stack) - Project Overview

Why this Project is Important

Covers real-world web app development. Full-stack project cycle experience. Helps in teamwork, problem-solving, and deployment practices.