

1. Food Delivery App

Technologies to Use:

- Frontend: React.js, Tailwind CSS/Bootstrap
- Backend: Node.js with Express.js
- Database: MongoDB
- Deployment: Netlify (Frontend), Render/Heroku (Backend)

Core Features:

1. User Authentication (JWT-based)
2. Menu Display
3. Add to Cart
4. Checkout Page
5. Order Status Tracking

Steps:

1. Setup React Project.
2. Design UI with Tailwind CSS.
3. Create Backend APIs with Express.js.
4. Connect MongoDB for data storage.
5. Integrate Frontend and Backend using Axios.
6. Deploy the app on Netlify and Heroku.

2. Booking App (Hotel/Flight)

Technologies to Use:

- Frontend: React.js, Material UI
- Backend: Node.js with Express.js
- Database: PostgreSQL
- Deployment: Vercel (Frontend), Railway/Heroku (Backend)

Core Features:

1. Search and Filters
2. User Authentication
3. Booking System
4. Payment Integration (Stripe API)
5. Booking History

Steps:

1. Setup React and Material UI.
2. Build UI components for search and booking.
3. Create backend routes and connect PostgreSQL.
4. Integrate Stripe for payments.
5. Deploy frontend on Vercel and backend on Railway.

3. Real-Time Chat App

Technologies to Use:

- Frontend: React.js, Tailwind CSS
- Backend: Node.js with Express.js
- Real-Time Communication: Socket.io
- Database: MongoDB
- Deployment: Netlify (Frontend), Render/Heroku (Backend)

Core Features:

1. User Authentication
2. Real-Time Messaging using Socket.io
3. Group Chat
4. User Presence (Online/Offline)
5. Chat History

Steps:

1. Setup React and Socket.io.

2. Build Chat UI components.
3. Create backend routes for saving messages.
4. Integrate Socket.io for real-time messaging.
5. Deploy the app.

4. Social Media App

Technologies to Use:

- Frontend: React.js, Tailwind CSS/Material UI
- Backend: Node.js with Express.js
- Database: MongoDB
- File Upload: Cloudinary
- Deployment: Vercel (Frontend), Render (Backend)

Core Features:

1. User Authentication
2. Create Posts (Image and Text)
3. Like and Comment System
4. Follow/Unfollow Users
5. User Profiles

Steps:

1. Setup React project and UI components.
2. Create backend APIs for posts and users.
3. Integrate Cloudinary for image uploads.
4. Connect MongoDB for data management.
5. Deploy on Vercel and Render.