E-commerce Platform Development Schedule with React Frontend

Day 1: Core Functionality Implementation

Hour 1-2: Project Setup

- Set up React Frontend:
  - Create a new React project using `npx create-react-app my-app`.
  - Install Redux for state management: `npm install redux react-redux`.
  - Configure a basic Redux store in your React project.
  - Set up routing with `react-router-dom`: `npm install react-router-dom`.

- Backend Setup:
  - Initialize the Node.js project if not already done.
  - Install necessary backend dependencies (`express`, `mongoose`, `dotenv`, `jsonwebtoken`).

Hour 3-4: User Authentication

- Backend:
  - Create a user model using Mongoose.
  - Implement user registration and login endpoints.
  - Secure routes with JWT middleware.

- Frontend:
  - Create registration and login forms using React.

- Implement authentication flow:
  - On successful login, save JWT token in local storage.
  - Use Redux to manage authentication state.
- Implement private routes in React that require authentication.

Hour 5-6: Product Listings

- Backend:
  - Create a product model in MongoDB.
  - Implement CRUD operations for products via API endpoints.

- Frontend:
  - Create a product listing page using React.
  - Fetch product data from the backend and display it.
  - Use Redux to manage the state of products.

Hour 7-8: Shopping Cart

- Backend:
  - Create endpoints to add, update, and remove items from the cart.

- Frontend:
  - Implement a shopping cart using Redux.
  - Allow users to add products to the cart from the product listing page.
  - Store cart data in the user's session or MongoDB.

Day 2: Additional Features and Enhancements

Hour 1-2: Order Tracking

- Backend:
  - Create an order model and corresponding API endpoints.

- Frontend:
  - Implement order history and tracking using React components.
  - Display order status and details.

Hour 3-4: Payment Integration

- Backend:
  - Integrate Stripe and PayPal APIs.
  - Create payment processing endpoints.

- Frontend:
  - Create payment forms using React.
  - Handle payment flow and update order status after payment.

Hour 5-6: Admin Panel

- Backend:
  - Create admin-specific routes and functionalities.

- Frontend:
  - Build a React-based admin panel.
  - Implement CRUD operations for product management.
  - Add authentication checks to secure admin routes.

Hour 7: Real-Time Product Recommendations

- Backend:
  - Use MongoDB queries or third-party services to generate recommendations.

- Frontend:
  - Display product recommendations on the product detail page.

Hour 8: Final Testing and Deployment

- Testing:
  - Perform end-to-end testing for both frontend and backend.
  - Identify and fix bugs.
  - Optimize code for performance.

- Deployment:
  - Deploy the backend to a service like Heroku.
  - Deploy the frontend to Netlify or Vercel.