E-Commerce Application Clone

High Level Design Documentation (HLD).

Project for:- iNeuron (E-Commerce Application Clone Internship) By:- Aman Negi

Content

1. Abstract:

This project entails the design and development of a frontend-only E-Commerce Application Clone aimed at providing users with a seamless online shopping experience. The application allows users to browse through a collection of products, explore individual items, and conveniently add desired products to their virtual shopping cart. Key functionalities include a dynamic homepage showcasing available products, category pages for organized browsing, and a user-friendly cart page for managing selected items. Leveraging modern frontend technologies such as React.js and React Router, the application ensures smooth navigation and responsive design across various devices. Data mocking techniques simulate backend API calls, enabling the presentation of sample product data, while local storage facilitates temporary cart item storage. Emphasizing user experience, the design integrates accessibility features, performance optimizations. Development tools such as ESLint, Prettier, and version control systems ensure code quality and collaboration efficiency. Testing strategies encompass unit testing, end-to-end testing, and accessibility testing to guarantee functionality and compliance with standards. Finally, deployment involves hosting the frontend application on static site hosting platforms with continuous integration and deployment pipelines for automated testing and deployment processes. This abstract outlines the comprehensive approach to building a robust frontend solution for an E-Commerce Application Clone, catering to modern development standards and user expectations.

2. Introduction.

2.1. Why this High level Design Documentation ?

The purpose of High Level Documentation is to (HLD) is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at the high level.

The HLD will:

- Present all of the design aspects and define them in detail.
- Describe the user interface being implemented.
- Describe the hardware and software interfaces.
- Describe the performance requirements.
- Include design features and the architecture of the project.
- List and Describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

2.2. Scope.

The HLD Documentation presents the structure of the system, such as the database, architecture, layers, application flow (Navigation), and the technology architecture. The HLD uses non-technical and mildly technical terms which should be understandable to the administrators of the system.

3. Description.

3.1. Problem Perspective and 3.2. Problem Statement

In response to the growing demand for intuitive online shopping experiences, this project aims to develop a frontend-only E-Commerce Application Clone. The challenge is to design a solution that showcases products, enables seamless navigation, and facilitates cart management without relying on a backend infrastructure. Leveraging frontend technologies and data mocking techniques, the objective is to create a dynamic and interactive platform that prioritizes user experience considerations such as accessibility, performance optimization, and responsive design. By addressing these requirements, the project seeks to fulfill the need for a modern and efficient E-Commerce application clone tailored for frontend development environments.

3.3. Proposed Solution.

Our solution involves building a frontend-only E-Commerce Application Clone using React.js and React Router. Key components include dynamic homepage, category pages, and a user-friendly cart page. Data mocking techniques will simulate backend API calls for product data, while local storage manages cart items temporarily. We'll prioritize user experience with accessibility features and performance optimizations. Deployment will be on static site hosting platforms with continuous integration for automated processes.

3.4. Solution Improvements.

- **Backend Integration:** Enhance with real-time data and functionalities like user authentication and order processing.
- **Advanced User Experience:** Implement features such as product recommendations and personalized profiles for enriched interaction.
- **Performance Optimization:** Continuously optimize for faster load times and smoother interactions.
- **Security and Localization**: Strengthen security measures and support multiple languages for enhanced user trust and accessibility.

4. Requirements

4.1 Hardware Requirements:-

A working computer to code with active internet connection.

4.2 Tools / Software Requirements:-

- Integrated Development Environment (IDE)
- Node.js and npm (Node Package Manager)
- React.js and React Router
- Git: Version control system for tracking changes in codebase.
- GitHub: Platforms for hosting and managing Git repositories.
- Create React App: Development environments for setting up a local server and building React applications.
- Code Quality Tools:
    - ESLint: JavaScript linter for identifying and fixing code errors and enforcing coding standards.
    - Prettier: Code formatter for maintaining consistent code style across the project.

5. Data Requirements.

Data requirements are essential for managing the core functionalities of your ecommerce application, including product listings, user interactions, cart management, order processing, and providing relevant information to users. Depending on the specific needs of your application, you may need to expand or customize these data requirements further.

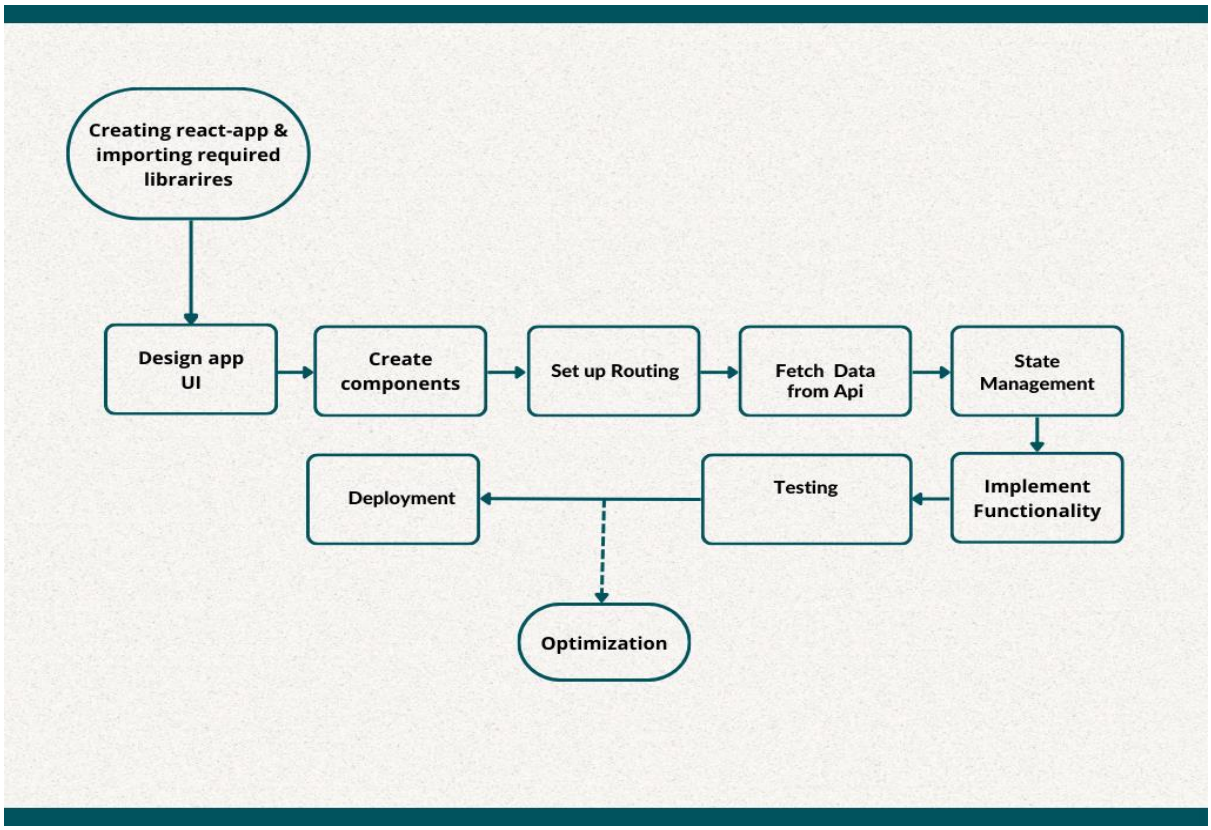
6. Constraints.

The web application should be user friendly so that without knowing any technical information user should be able to use our predictive system.

7. Assumptions.

The main objective of the project ecommerce app is to provide users with a seamless and enjoyable shopping experience, facilitating easy browsing, selection, and purchase of products while ensuring security, reliability, and convenience.

8. Design Flow.



9. Logging & Error Handling.

Every important step is logged within the system that runs internally, it basically shows us the data time of each process which is done with our system. It provides us with logging information for end to end web applications. The logging which we have done in the above process helps us to handle the error because the error is being logged in log files (every time we run code) so that the developer can rectify it.

10. Performance Evaluation.

10.1 Page Load Time:

Measure the time it takes for your app's pages to load. This includes both initial page load and subsequent interactions.

10.2 Response Time:

Evaluate the responsiveness of your app by measuring the time it takes for the server to respond to user actions, such as adding items to the cart or updating product filters.

10.3 Error Rate :

Monitor the occurrence of errors and exceptions in your app, including server errors, client-side errors, and network-related issues. Aim to minimize error rates to ensure a smooth user experience.

10.4. Mobile Performance:

Assess the performance of your app on mobile devices, including load times, responsiveness, and usability. Ensure that your app delivers a consistent experience across different screen sizes and devices.

11. Conclusion.

We have successfully built an e-commerce web application clone using ReactJs. Its user-friendly interface, advanced filtering options, and seamless checkout process, the app aims to meet the diverse needs of our customers and enhance their online shopping journey.