

SHOPPING CART

Contents

Document Version Control.....	2
Abstract.....	4
1 Introduction.....	5
1.1 Why this High-Level Design Document?	5
1.2 Scope.....	5
1.3 Definitions.....	5
2 General Description.....	6
2.1 Product Perspective	6
2.2 Problem statement	6
2.3 PROPOSED SOLUTION.....	6
2.4 FURTHER IMPROVEMENTS	6
2.5 Technical Requirements.....	6
2.6 Tools used	7
2.7 Constraints	7
2.8 Assumptions.....	7
3 Design Details	7
3.1 Process Flow.....	7
3.2 Error Handling	7
4 Performance.....	8
4.1 Reusability.....	8
4.2 Application Compatibility.....	8
4.3 Resource Utilization.....	8
4.4 Deployment.....	8
5 Dashboards	9
5.1 KPIs (Key Performance Indicators).....	10
6 Conclusion.....	10

Abstract

This project introduces an advanced web-based shopping cart application built with React, showcasing its prowess in revolutionizing the landscape of online shopping. By harnessing React's modular architecture and robust state management, the application seamlessly integrates dynamic cart updates, ensuring real-time synchronization with user actions. The interface's fluidity enhances user interaction, while the intuitive design simplifies the management of selected products. This project not only underscores React's capabilities in crafting modern and responsive e-commerce solutions but also emphasizes its role in elevating user engagement through its component-driven approach. In an era where online shopping is ubiquitous, this React-based cart application stands as a testament to the platform's ability to create compelling, user-centric digital experiences that align with the ever-evolving expectations of contemporary consumers.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document 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 a 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:
 - o Security
 - o Reliability
 - o Maintainability
 - o Portability
 - o Reusability
 - o Application compatibility
 - o Resource utilization
 - o Serviceability

1.2 Scope

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

1.3 Definitions

<i>Term</i>	<i>Description</i>
<i>UI</i>	User Interface. The visual elements and interactions through which users interact with a system.
<i>React</i>	An open-source JavaScript library for building user interfaces

2 General Description

2.1 Product Perspective

This section unfurls the application's role within the larger digital landscape. It highlights the interactions with external systems, emphasizing how the application interfaces with users, products, and third-party services.

2.2 Problem statement

Delving into the root challenges, this part outlines the difficulties encountered by users when shopping online, underscoring the need for an efficient and user-friendly shopping cart application.

2.3 PROPOSED SOLUTION

A spotlight is cast on the overarching solution adopted for addressing the identified problems. The blueprint revolves around the strategic utilization of React's component-based architecture to forge an interface that ensures fluidity, real-time updates, and intuitive product management.

2.4 FURTHER IMPROVEMENTS

This section invites speculation on potential augmentations that could be woven into the application's fabric in the future. Ideas may include social media integration, personalized product recommendations, and optimizing the checkout process.

2.5 Technical Requirements

The React-based shopping cart application is meticulously designed, prioritizing user experience through cross-device compatibility, responsive design, and real-time updates. Security is ensured via encryption and trusted payment gateways. Streamlined navigation, browser compatibility, accessibility, and scalability enhance efficiency and inclusivity. Rigorous testing, comprehensive documentation, and high-quality code underpin reliability and maintainability, facilitating seamless growth and future enhancements.

2.6 Tools Used

A comprehensive inventory of tools employed during the development lifecycle is outlined. For instance, a code editor such as Visual Studio Code, Git for version control, and ROS (Robotic Operating System) for specific functionalities are among the tools used.



- Front end development is done using HTML/CSS
- GitHub is used as version control system.

2.7 Constraints

The limitations influencing the project's design are laid bare, encapsulating aspects such as time constraints, resource availability, and compatibility considerations.

2.8 Assumptions

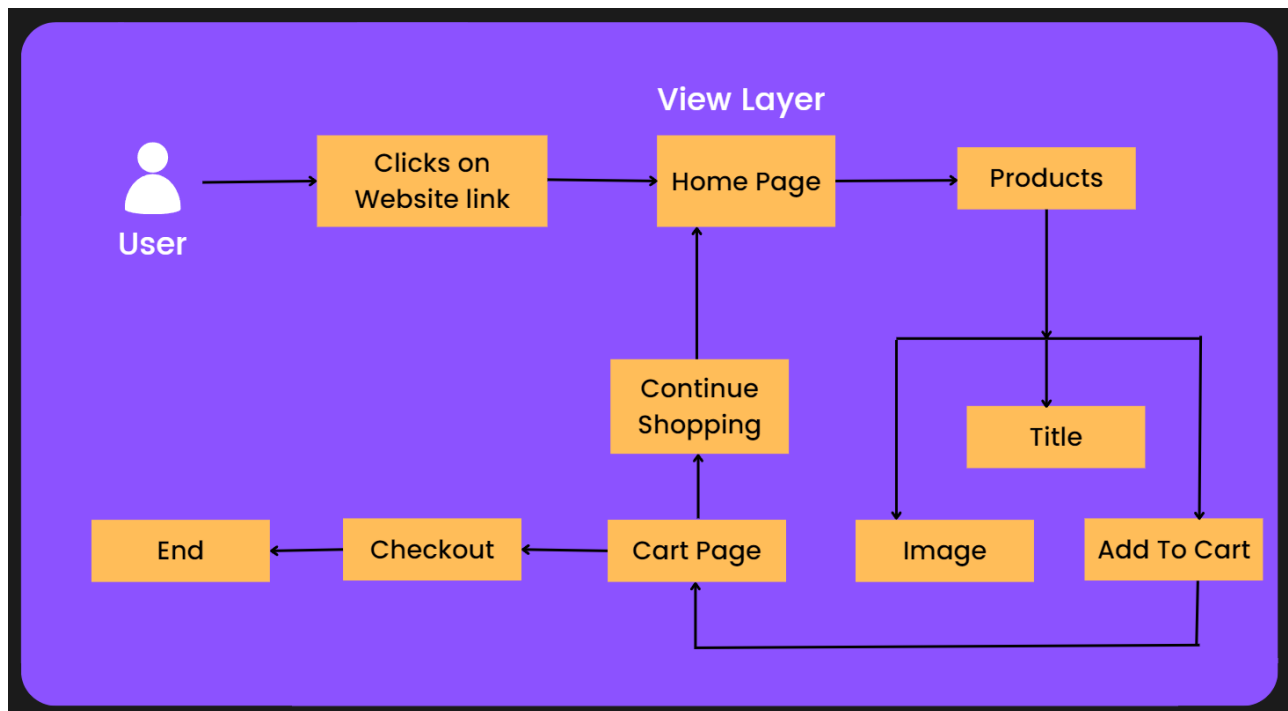
This segment encapsulates assumptions made during the design phase, including users' familiarity with e-commerce concepts and the application's primary interaction with modern web browsers..

3 Design Details

3.1 Process Flow

The application's process flow encompasses two pivotal stages. The first involves model training and evaluation, where machine learning models are fed with historical data to facilitate personalized product recommendations. Subsequently, the deployment process integrates these models into the application's backend, paving the way for seamless user experiences.

Proposed methodology



3.2 Error Handling

The application is fortified with robust error-handling mechanisms. When errors occur, users are presented with clear and user-friendly error messages that guide them on corrective actions. Behind the scenes, errors are logged and analyzed to enable effective troubleshooting and continuous refinement of the user experience.

4 Performance

Performance optimization forms a cornerstone of the application's design. Techniques such as lazy loading of assets, server-side rendering, and asynchronous loading are employed to ensure swift page load times and responsive interactions, enhancing overall user satisfaction.

4.1 Reusability

A hallmark of the design approach is the creation of modular and reusable components. These building blocks can be easily repurposed across the application, promoting efficient development, consistency, and easier maintenance.

4.2 Application Compatibility

The application is thoughtfully crafted to ensure compatibility across a wide range of devices, browsers, and platforms. The responsive design ensures a seamless experience on various screen sizes and orientations.

4.3 Resource Utilization

Efficient resource management is a guiding principle. The application optimizes resource usage, striking a balance between memory consumption, processing power, and bandwidth utilization to ensure a responsive and smooth user experience.

4.4 Deployment



5 Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the unveiled problems that if not addressed in time could cause catastrophes of unimaginable impact.



As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

5.1 KPIs (Key Performance Indicators)

The application's dashboards are equipped with a set of vital Key Performance Indicators (KPIs) that provide insights into its health and effectiveness. Metrics such as user engagement, conversion rates, and average order values are monitored to guide data-driven decision-making and continuous improvement.

6 Conclusion

In summary, this High-Level Design document offers a comprehensive blueprint for the React-based shopping cart application. It underscores the meticulous considerations that shape the architectural decisions, fostering an environment of seamless user interaction, real-time updates, and efficient data management. This document serves as a guiding beacon as the development journey unfolds, culminating in a sophisticated and user-centric e-commerce solution.

