<u>Low Level Design (LLD)</u>

E-Commerce Application Clone

Revision No: 01
Last Date of Revision: 19/07/24

Yalla Meghana

Contents

1. Document Version Control

Date	Version No.
15/07/24	1
19/07/24	2

2. Introduction

2.1 Why this Low-Level Design Document?

A Low-Level Design (LLD) document is a detailed technical blueprint that expands on the high-level design specifications provided in the High-Level Design (HLD) document.

LLD helps developers understand how to implement each part of the system effectively.

LLD describes individual system components, including their purpose, functionality, interfaces, and dependencies.

2.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

iNeuron

3. Architecture

The success of an online store depends heavily on its architecture. The design, structure, and functionality of a website can impact user experience, conversion rates, and search engine optimization significantly.

It encompasses the layout, functionality, user interface, and the technologies used to develop the website.

This application is easy to navigate, products are easy to find, and can be added to cart .

It provides a smooth and user-friendly experience for online shoppers.

- Improved Performance
- Better Scalability
- Increased flexibility
- Easier maintenance
- Increased customer satisfaction

This is deployed locally on a computer.

```
Compiled successfully!

You can now view ineuron_project in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.43.180:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

4. Component Design

4.1 Product Listing:

Description: Displays a list of products with filtering and sorting options.

Interfaces: React components to fetch and display product data from static files.

4.2 Shopping Cart:

Description: Manages the shopping cart functionality, allowing users to add, remove, and update items.

Interfaces: React components to handle cart operations and state management.

4.3 Checkout:

Description: Facilitates the checkout process, including order summary and user information collection.

Interfaces: Integrates with the cart component to finalize the purchase.

5. Unit Test Cases

Verification of product details: Products are correctly stored and displayed.

Verification of Cart: Adding & Removal of items to & from the cart are verified. Ensured correct calculation of total price.

Verification of Checkout: Verified address, payment processing & order confirmation.