

# 1. Introduction

The **purpose of this document** is to describe the overall behaviour of the "SmartStore website". This SRS defines and describes the operations, performance and quality assurance requirements, which is to be developed. Provides description of the function & non-functional requirements, design constraints and assumptions. Requirements described in this document will be used as guidelines to develop the application.

## Project Purpose

The purpose of the SmartStore website is to provide a user-friendly ecommerce retail management application that enables customers to purchase products online. With the growing popularity of online shopping, the SmartStore website aims to offer a hassle-free and convenient shopping experience to customers who prefer shopping from the comfort of their homes.

- The website will feature a wide variety of products across multiple categories, such as clothing, electronics, home decor, and more, catering to the diverse needs and preferences of the customers.
- The platform will be user-friendly and provide an easy navigation system to allow customers to easily browse and search for their desired products.
- Customers can access their shopping history to keep track of their previous purchases which helps them to monitor their spending
- They can add products to their Wishlist and keep track of their favourite products, and make more informed purchasing decisions in the future.
- The website offers a convenient and centralized platform for customers to access a wide range of products in one place.
- The project's goal is to make online shopping a more accessible and enjoyable experience for all customers.

## Project Scope:

The SmartStore website provides a range of features to enhance the online shopping experience for customers. Users can easily browse and search for products by filtering categories and can add items to their cart or Wishlist for later. The platform also allows users to checkout using cash on delivery as a payment method, making the process convenient and hassle-free. Customers can also keep track of their orders and edit their profiles and address details as needed. The website also provides an admin portal, enabling authorized personnel to add new products and manage the inventory.

This SmartStore retail application will facilitate the following: -

- ✓ Allow users to add products to the cart, Wishlist
- ✓ Allow users to search for products, filter product categories.
- ✓ Allow user to checkout with cash on delivery payment method.
- ✓ Allow user to check user's orders.
- ✓ Allow user to edit their profile, edit their address details.
- ✓ Allow admin to add products.

## 2. Overall Description

### Product Perspective

SmartStore retail management Application will be a platform where Admin/seller will be able to add products in the website and users will be able to buy those products. Orders history and quantity of the products will be managed by the application to make shopping and selling easy for the customers and the seller.

### Product Features

Following Features will be provided by our website: -

- Login
  - ✓ Users will be able to log in to the app after signup
  - ✓ Website also allow users to login using google.
- view all the product available for purchase.
  - ✓ Users can view all the products and see the details of each product.
  - ✓ User can search of the product and filter the products with categories.
  - ✓ Users can view the products they have ordered from the past orders history. Users can view details of the products from Cart and Wishlist.
- Add to cart.
  - ✓ User can add product to the cart.

- ✓ While adding products to the cart, user cannot add the more than the quantity of the products available and after each order the product available quantity will be decreased by the order quantity
- ✓ Whenever user adds a product in the cart, he can see the number of items in the current cart on the top of cart icon in the navbar.
- Checkout
  - ✓ When user want to order the product, the user can checkout and edit the address of delivery.
  - ✓ User can view total summary of the order, total cost, and cost of each product.
  - ✓ He can chode the payment method and place order.
- Add to Wishlist
  - ✓ User can add the products to the Wishlist for the future purchases.
  - ✓ When whenever the product is in the Wishlist, user can see red heart on the products and if not a white heart.
- Orders
  - ✓ User can see the details of every past order and details of each product in the order.
  - ✓ User can view total cost, cost of product at the time of order and keep track of their purchase.
- A web-based Admin Portal will be implemented where Admin will be able to add new products.

## User Classes & Characteristics

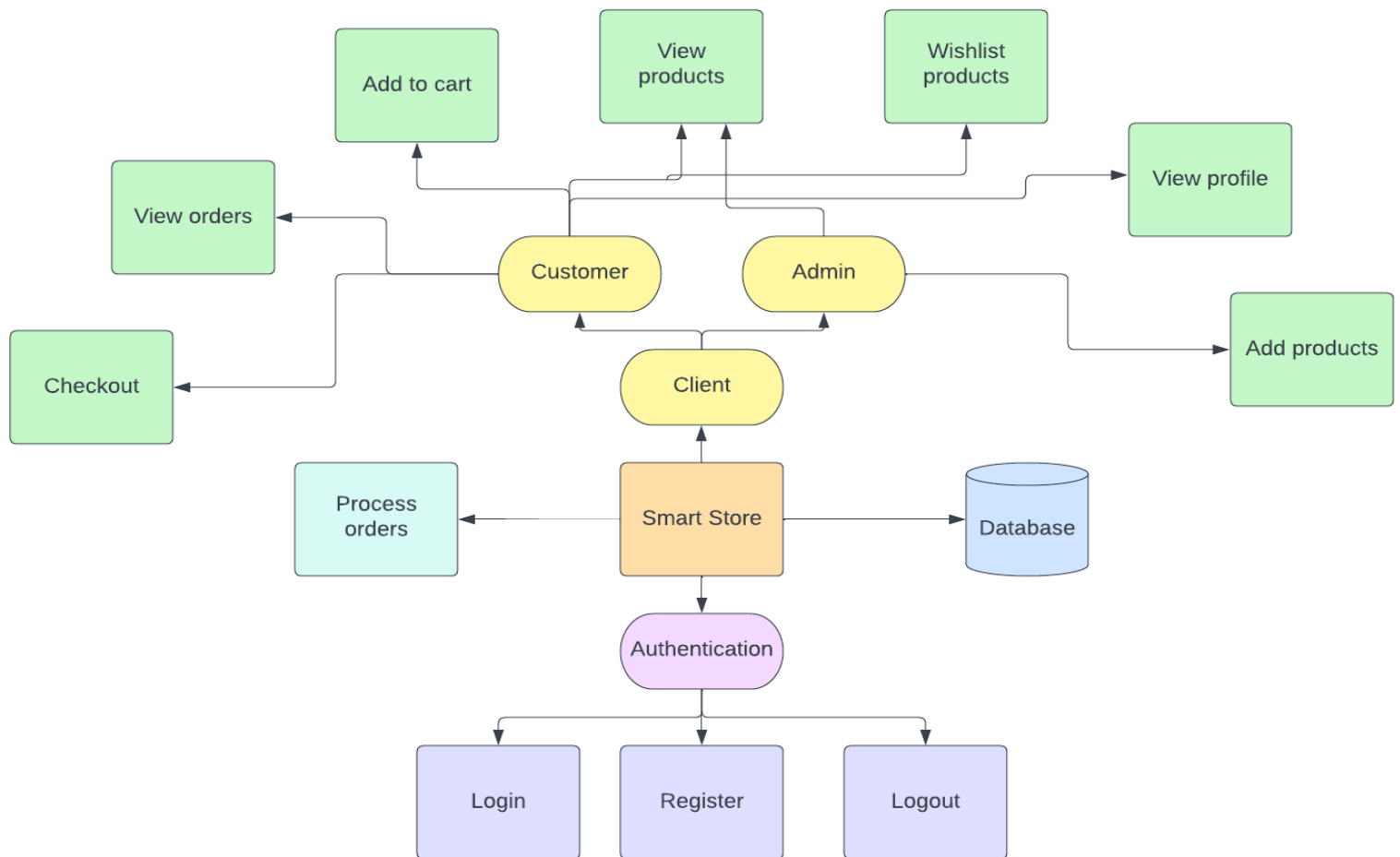
### Stack holders/ Physical actors:

- Customers: The target audience of the Smartstore application will be people who want to shop online for a variety of products. The customers may be of any age, gender, or location.
- Admin: The admin will manage the products, inventory, pricing, and other related aspects of the Smartstore application.

### System Actors:

- Operating Environment: Smartstore application shall operate with any smartphone or computer with a modern web browser and internet connectivity.

## 3. Block Diagram



## 4. Software Stack

- Web-based application: Frontend [React, HTML, CSS], Backend [Nodejs and Express]
- Database: PostgreSQL, pgAdmin
- Authentication: Google Auth API

# 5. Features & Requirements

## System/Hardware features

### 1. User registration:

description & priority

- to use the system, user first signup in the site by filling a simple form, the link of which is provided in the home page navbar.
- if he does not want to sign up, he can login use google login.
- priority: low

stimulus & response

precondition: user in homepage, clicked 'login' and then in login page clicked signup.

- user fills the form and submit
  - form validation checks to be passed for successful signup.
  - user redirected to login page. then he can login.
  - all the user passwords are encrypted
- post condition: the user get its account interface

### 2. User login:

description & priority

- the user login in the login window provided in the homepage navbar by filling the email id and password if he is already registered.
- the user can also login using google without signup.
- priority: high

stimulus & response

precondition: user is register if not logging in through google.

- user fill login form and submit it
  - user login credentials are validated from google auth or database
  - if login info is not correct or all fields are not filled, submission fails and resubmission is required
  - if successfully submitted redirect to home page of smartstore
- post condition: the user get logged in and see his personalised details.

### 3. Search engine, Filters, and View products:

description & priority

- user can search for products by name, string in the description, availability and by specific categories.
- priority: medium

stimulus & response

precondition: no preconditions, any user can view and search for products.

- user opt to search for products.
- typing search text or selecting category provides available options from database.

post condition: user gets the required list of search text is query is not empty and get to see all products if search text is empty.

### 4. Cart and Wishlist:

description & priority

- on clicking on add to cart and heart symbol on each product card customer can add products to the cart and wishlist. he can view products in cart and wishlist by clicking on symbols on navbar.
- priority: high

stimulus & response

precondition: user is logged in.

- user selects the he wants and click on add to cart button or heart button.
- the product gets added to the cart or wishlist depending on the users wish.
- user can increase or decrease the quantity of the product in the cart by pressing + or – symbols until the quantity of product available in stock.
- user can delete the product in wishlist or cart by clicking on delete button.
- compute total cost.

post condition: products get deleted, added in the cart or wishlist.

### 5. Place Order:

description & priority

- user can buy the products in the by clicking on checkout
- priority: high

stimulus & response

precondition: user is logged in and added the products in the cart

- user clicks on checkout and edit his delivery address products.
- able to see the summary and select the payment method.
- without selecting payment method user cannot place order.

post condition: if everything is correct the order will be placed and cart becomes empty and items in the cart inserted into orders. quantity of the products in the stock decreases.

## 6. Account:

description & priority

- user can edit and view profile.
- user can view
- priority: low

stimulus & response

precondition: user is logged in.

- view profile and edit profile.
- view past orders.

post condition: user details get saved in profile.

## 7. Admin portal:

description & priority

- admin can view and add products in the stock.
- priority: high

stimulus & response

precondition: admin has unique email id and encrypted password.

- admin add products in database
- admin can view all products in the database.

post condition: product gets added in the stock.

## Functional requirements

1. **User Registration:** The system shall provide a simple form for the user to fill out in order to register and create an account. The system shall also allow users to log in with their Google account. All user passwords shall be encrypted.

2. **User Login:** The system shall allow users to log in using their registered email and password, or with their Google account. The system shall validate user credentials from the database or Google authentication. If the login information is not correct or all fields are not filled, the submission shall fail and resubmission shall be required.
3. **Search Engine, Filters, and View Products:** The system shall allow users to search for products by name, string in the description, availability, and specific categories. The system shall provide available options from the database based on the search text or category. If the search text is not empty, the user shall receive a required list of products. If the search text is empty, the user shall receive all products.
4. **Cart and Wishlist:** The system shall allow users to add products to their cart and wishlist by clicking on the add to cart and heart symbol on each product card. The system shall allow users to view products in their cart and wishlist by clicking on the corresponding symbols in the navbar. Users shall be able to increase or decrease the quantity of products in the cart and delete products from the wishlist or cart. The system shall compute the total cost.
5. **Place Order:** The system shall allow users to buy products by clicking on the checkout button. The user shall be able to edit their delivery address and view a summary of their order before selecting a payment method. The user shall not be able to place an order without selecting a payment method. If everything is correct, the order shall be placed, the cart becomes empty, and the items in the cart shall be inserted into orders. The quantity of the products in the stock shall decrease.
6. **Account:** The system shall allow users to view and edit their profile. Users shall be able to view their past orders.
7. **Admin Portal:** The system shall provide a web-based Admin Portal where the admin shall manage the products, inventory, pricing, and other related aspects of the Smartstore application. The admin shall be able to add new products to the database and view all products in the database. The admin shall have unique email and encrypted password credentials.

## Interface requirements

### User Interfaces:

- All pages shall have a consistent layout, style, and color scheme for easy navigation and recognition.
- The user interfaces shall be intuitive, user-friendly, and accessible to all users.
- The forms and input fields shall have appropriate labels and placeholders for clear understanding and guidance.



- The error messages shall be displayed in a prominent location with clear instructions on how to fix them.

#### Hardware Interfaces:

- The application shall be compatible with desktops, laptops.
- The device used to open the software should have internet access.

#### Software Interfaces:

- The application shall use a secure protocol (HTTPS) for all communications between the server and client to ensure data privacy and protection.
- The application shall integrate with popular payment gateways to facilitate online payments.
- The application shall use a reliable database management system to ensure efficient storage, retrieval, and management of data.

#### Communication Interfaces:

- The HTTP protocol will be used for communication between server application and client.

## Software system attributes & Non-functional requirements

#### Smartstore Retail Management Application:

- **Security:** The Smartstore application shall provide secure access to user information by implementing secure user authentication and authorization mechanisms. User passwords shall be encrypted, and all sensitive data shall be encrypted during transmission. The application shall follow the best security practices to prevent hacking, data breaches, and other security-related issues.
- **Accuracy:** The Smartstore application shall accurately track inventory levels, product orders, and sales data. The system shall provide real-time updates to ensure accurate stock levels, pricing, and product information. The application shall provide robust data validation mechanisms to ensure data accuracy and consistency.
- **User-Friendly:** The Smartstore application shall provide a user-friendly interface for both customers and admin. The system shall be easy to navigate and shall provide clear instructions and feedback to users. The application shall

provide simple and intuitive interfaces for all key features, such as product search, ordering, payment, and customer management.

- **Robustness:** The Smartstore application shall provide robust and reliable performance, even under high traffic loads. The system shall be designed to handle large numbers of simultaneous users and transactions. The application shall provide error handling and recovery mechanisms to ensure smooth operation in the event of system failures or errors.
- **Reliability:** The Smartstore application shall provide reliable and consistent performance, even under adverse conditions. The system shall provide mechanisms for data backup and recovery, to ensure that data is not lost due to system failures or crashes. The application shall provide mechanisms for detecting and handling errors, such as network errors or database failures.
- **Availability:** The Smartstore application shall be available to users at all times, with minimal downtime for maintenance or upgrades. The application shall provide mechanisms for handling server maintenance or upgrades without interrupting user access. The system shall be designed to scale up or down as needed to ensure that it can handle varying levels of traffic and load.
- **Maintainability:** The Smartstore application shall be designed with maintainability in mind, to ensure that it can be easily modified or updated as needed. The application shall be modular and well-documented, with clear separation between front-end and back-end components. The system shall be designed to minimize dependencies and to allow for easy testing and debugging of individual components.
- **Portability:** The Smartstore application shall be designed for portability, to ensure that it can be easily adapted to different platforms and environments. The system shall be designed to be easily deployed on different hardware and software platforms, such as cloud-based services or on-premises servers. The application shall be written using industry-standard programming languages and frameworks to ensure compatibility with different platforms and environments

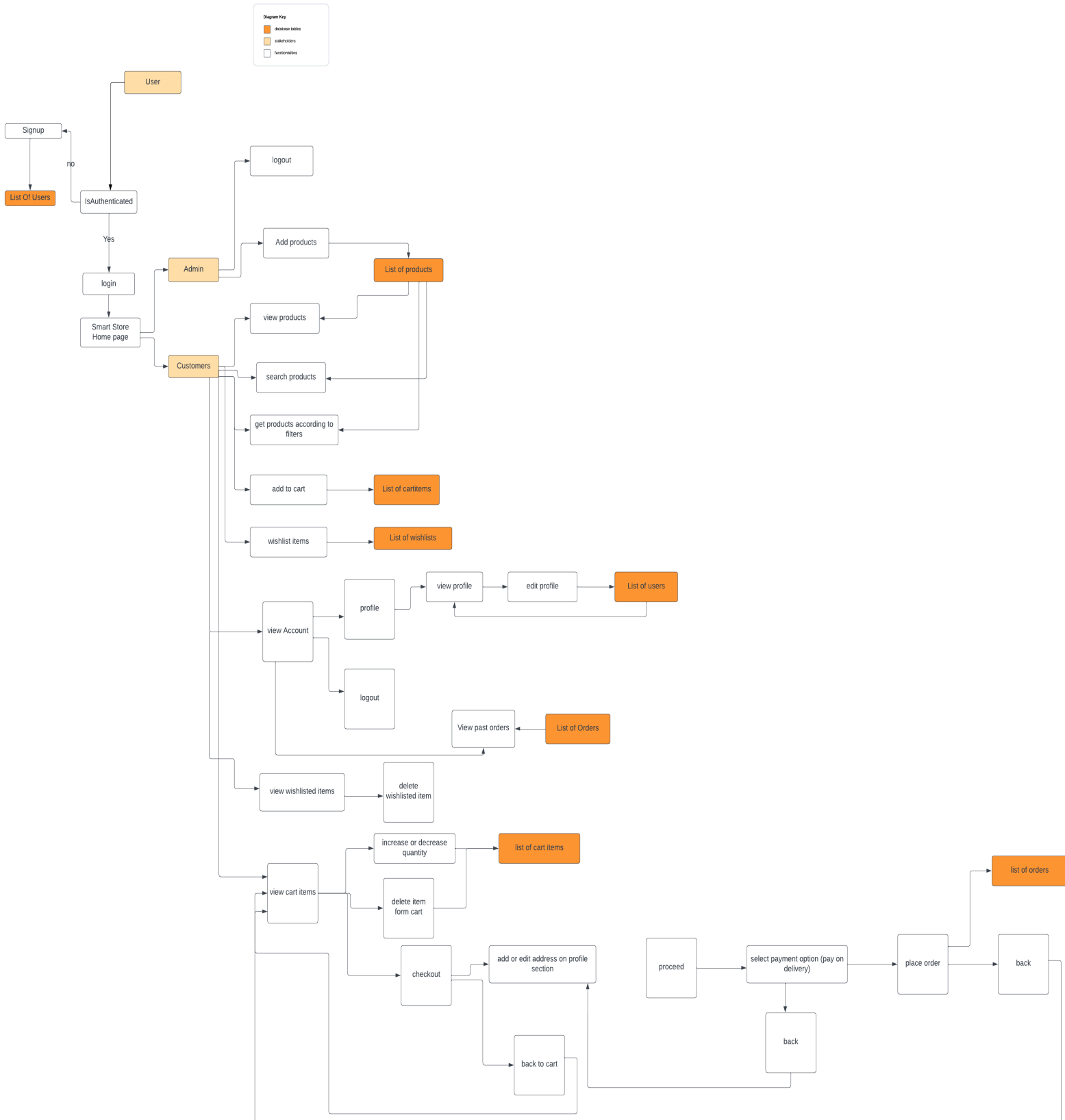
## 6. Implementation details

### Assumptions

- The email ID and password for the admin will be assumed before (username: [admin@iitrpr.ac.in](mailto:admin@iitrpr.ac.in) , password: admin123)
- Payment method will be only cash on delivery.
- We can only add one address in the profile for delivery.
- Admin can only add the products.

# Design and Implementation Constraints

This application is provisioned to be built using React, Express, NodeJS and PostgreSQL along with an admin to be built using the same tech stack.



## 7. User Description

As a user, you can access the Smartstore retail management application through a web browser, which means you do not need to download or install any additional software. Simply open the website on your preferred browser, and you will be able to start shopping immediately. You can browse through a wide range of products and categories, add items to your cart, and make purchases securely using a variety of payment options.

As a user, you can create an account on the Smartstore retail management application to enjoy additional features such as order tracking, personalized recommendations, and loyalty rewards. You can also easily manage your account details, including your shipping address and payment information, and view your order history.

The Smartstore retail management application also provides customer support through various channels, including live chat, email, and phone. You can get help with any questions or issues you may encounter while shopping, such as product inquiries, order status updates, or payment concerns. Overall, the user experience is designed to be easy, convenient, and enjoyable for all shoppers.

## 8. Tasks Workflow

1. Prepare SRS document
2. Prepare design documents.
3. Design Work breakdown structure
4. Setup basic application environment.
5. Setup login and signup activity
6. Database schema, creation, and user authentication
7. write all the functionalities to be implemented or Design website.
8. Create pages for all the functions or components.
9. Write all the function frontend and backend.
10. Preparation of test plans
11. Overall feature Integration
12. Testing of software and resolving issues/bugs

13. Continuous build and deployment of software.

14. Final deployment of the software.