Software Design Specification Document (CS360)

StoreX

Group Number: 6

Mati Ur Rehman
Muhammad Hamza Sajjad
Muhammad Hamza Shahzad
Muhammad Harris
Muhammad Raahem Asghar

Course: Software Engineering CS360

Instructor: Suleman Shahid

University: Lahore University of Management Sciences (LUMS)

Version: 1.0
Date: (21/03/2021)
Number of hours spent on this document: 250

Contents

1 Change Log

1.1 Project Scope

The function of StoreX is to provide a wide array of items to be bought by a cash on delivery method or for viewing different items. The users browse through the items and if they want to purchase something, they will have to register an account to do so and carry on with the purchase. This will help the store manager increase his sales by connecting to a larger customer base. This would especially remain useful for people living at a distance. This project will streamline the process of buying items and will have the added convenience of having them delivered straight at your door. It would be useful for any local store/outlet/market etc. who do not have an online service yet or their online service is not up to the mark. As compared to the status quo, we will be using the latest technologies which will make our application scalable and will make it faster and efficient. This web application can be accessed through any computer or mobile phone or tablet having an active internet connection. Other than that, this web application can run on any modern web browser applications such as Google Chrome, Mozilla Firefox etc.

1.2 Change Log

- We have removed the use case regarding giving admin privileges to different users as we thought that having a single admin would make more sense.
- We did not add options for signing up or logging in through Google or Facebook. The user can log in only using the email with which they created an account on StoreX.

2 Introduction

2.1 Document Purpose

The purpose of this SDS is to explain the requirements, designs, theoretical implementations and plan of an e-commerce store, StoreX. This ecommerce store intends to increase its revenue through sales via its web application. This document focuses on the first release of the product and explains the requirements of both user and admin side of the application. It contains diagrams explaining how the use cases within the application will be met, their execution sequence, and their output. It also explains the different components of the system and the different classes through diagrams. It explains about how things are organized in our database and shows how the screens will look like. It also explains the different requirements of the system.

2.2 Product Scope

The function of StoreX is to provide a wide array of items to be bought by a cash on delivery method or for viewing different items. The users browse through the items and if they want to purchase something, they will have to register an account to do so and carry on with the purchase. This will help the store manager increase his sales by connecting to a larger customer base. This would especially remain useful for people living at a distance. This project will streamline the process of buying items and will have the added convenience of having them delivered straight at your door. It would be useful for any local store/outlet/market etc. who do not have an online service yet or their online service is not up to the mark. As compared to the status quo, we will be using the latest technologies which will make our application scalable and will make it faster and efficient. This web application can be accessed through any computer or mobile phone or tablet having an active internet connection. Other than that, this web application can run on any modern web browser applications such as Google Chrome, Mozilla Firefox etc.

2.3 Intended Audience and Document Overview

This SRS document is intended for the teaching assistants and the instructor. The SRS covers our product, StoreX. It contains an overall description of the product, its functionalities, requirements, assumptions made during development and non-functional requirements. This document will help us plan the design of the final product while also making sure the expectations of the users are met.
The SRS is organised into three main parts, overall description, functional requirements and non-functional requirements, in that order. Overall description covers the uses of the web application. Functional requirements cover the tasks the web application should be able to perform for its users. Non-functional requirements other than the functions it should be able to perform e.g. performance requirements, software quality, security etc.

This document should be read by the readers starting from section 2.0, the introduction to get to know more about the product, its design, its purpose, its context, and the important definitions within the document. The reader should then proceed forward towards reading about the overall description, to get to know more about the normal control flow of the system, its constraint and architectural design. Then comes the system architecture. The system architecture should be viewed by both developers and designers. It contains the component diagram, the class diagram, the sequence diagrams, and the activity diagrams. The component diagram shows all the components of the system and how they are connected with eachother. Its important for designers to view this so that they know the components that are interconnected and they can design screens accordingly to ensure that such interconnectivity is maintained within the screens as well. Developers should read it so that they know how they will have to logically connect all those components specified in the component diagram, together. The sequence diagram is for the top 3 use cases of the system and shows the sequence of instructions that are carried out along the path to achieve the desired output result of the specific use case. This is important for developers so that they know how to logically integrate the functionalities specified by the sequence diagrams into the system. The activity diagram of a particular use case shows the flow of control for that use case. Its important for designers to know this to design screens that can exhibit such a flow of control and its important for developers to know this so that they can know how to logically connect the system for that particular use case. The class diagram is important for developers to know how many classes should be there in the system.

Then within section 4 comes the DB schema and its explanation which is important for both developers and designers to know. Its important for designers so that they know how different tables are related to eachother and they can design the screens right way to be connected accordingly. Its important for developers so that they know what is expected from them while connecting and integrating the system, which different components are related to eachother and so on. In section 5, we have discussed the screen designs, their logics and their visual representations. By looking at the screen design, and seeing the rationale behind the choices made, and the expected functionality of the screens, front end engineers can develop the system accordingly to fulfill the design and functioning expectations. In section 6, we have explained the non-functional requirements which are important for developers and testers to test the system against them and ensure that there are no points of failures.

2.4 Definitions, Acronyms and Abbreviations

Admin	The person who has all the control of web application, who is in charge of adding items and controlling their prices.
Analytics	Provide data regarding products. For example, how many products are in stock, which are the most popular items etc.
Concurrent users	Users that are using the web application at the same time
CSRF	Cross-Site Request Forgery
DoS	Denial of service attack
Inventory	The storage of the items to sold
Item	An individual article or unit with an ID, title, price
Response time	Time taken for the user to get their answer
User	A person who visits the web application.
Wishlist	List of products that the user likes and would like to buy later either because they are unavailable or because of some other reason.
XSS	Cross-site scripting

2.5 References and Acknowledgments

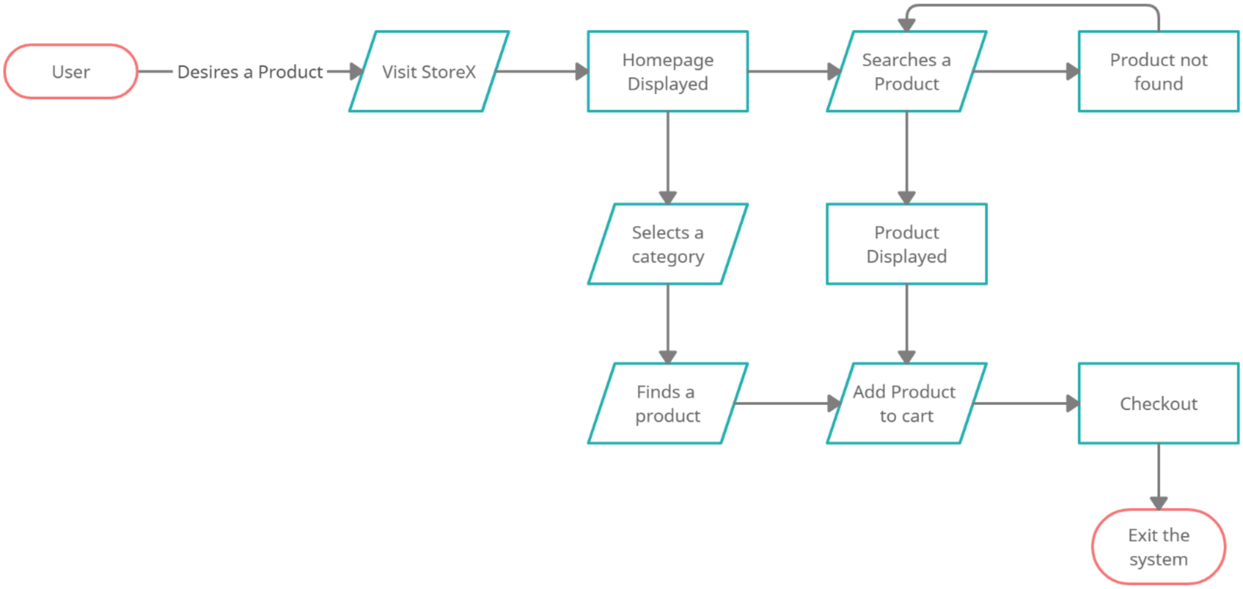
1) Our SRS document

2) Material Design (https://material.io/)

3 Overall Description

3.1 System overview

The system is designed for customers who prefer the online mode of shopping and includes a wide variety of categories from which the customers can select different products. The user visits the web application and is directed to the Homepage of the system, from where they can navigate to different pages. From the homepage, they can search for products using the search bar within the header, or add products to their "*favourites*" list from within the header. They can also check out the products on sale by clicking on the "sale button" in the header. From within the header, they can also visit their shopping cart page and see what items are currently there in their shopping cart. The users can also use the navigation bar on left to either create an account or login to their account. From the navigation bar, they can also visit different category pages for different category of products that the store offers, or send a complaint or a suggestion to the admin. They can use the footer to see the location and contact details of the store, subscribe to the store's newsletter or go back to important links like shopping cart, Home, Sale etc. The following diagram details how the system interacts with the environment:



3.2 System constraints

3.2.1 Hardware or software environment

The user needs a system with at least 2GB RAM and 1.2Ghz dual-core processor for smooth performance. Secondly the user needs a modern web browser e.g. chrome, Firefox, Safari, IE etc.

3.2.2 Interface/Protocol Requirements

The user interface should be easy to use such that any new user should get accustomed to it within 10 minutes. The navigation should be informative such that at any given time the user can see what part of the application it is using or at what page of the application it is on.

3.2.3 Data repository and distribution requirements

The application needs a distributed cloud storage service for storing user data and products information.

3.2.4 Safety requirements (or other such regulations)

1. The web application should safely store user's data ensuring that there is no data breach. It should prevent attackers from being able to get access to user.
2. There should be an application gateway between the system and the rest of the network so that all the network traffic is received at this gateway and the gateway only allows authorized packets to reach the system**.**
3. Initially the application will need 0.1GB of storage space but it should be easy to scale as the number of users and storage demand increases.

3.2.5 Security requirements (or other such regulations)

1. The web application should be secure from DoS attacks
2. Alongside using cookies, the web application should have a unique token for every round trip for every user to prevent CSRF attacks
3. The web application should follow the Same Origin Policy.
4. Ensure Validation of cookies and query strings against a rigorous specification of what should and should not be allowed. This will ensure prevention of XSS attacks.
5. Query structure should be specified independent of the user to prevent SQL injection attacks
6. Frame-busting must be implemented to ensure that frames can not be included in other websites.

3.2.6 Performance requirements

1. The response time for the web application shall not exceed 0.5s.
2. The web application should be able to handle 100,000 concurrent users at a time.

3. The system must implement standard web security practices to prevent any chance of attack.
4. The downtime for the web application shall not exceed 30 minutes on any day in-order to prevent significant loss in revenue for the owner.
5. Web application maintenance should be easy and cost-effective such that the maintenance cost shall not exceed 10% of the total cost each year.
6. The UI should be user friendly such that it should not take more than 5 minutes for any new user to understand the interface.

3.2.7 Network communications

It should use HTTPS instead of HTTP because data transfer over HTTP is not encrypted. Data is transferred between systems as plain text if HTTP is used as a protocol for communication. HTTPS on the other hand ensures end-to-end encryption of data when transferring it between systems. Hence many users don’t even use e-commerce platforms that use HTTP for end to end communication.

3.3 Architectural strategies

We will be using ReactJS to implement the frontend of our web application. Making user interfaces on React is efficient, convenient and time saving. It allows the components of the interface to change in real time depending upon the state of the web application. Other than that, most of our team members were familiar with React. This would allow us to spend less time on learning the framework and more on implementation. Since react is the most popular front-end framework, it has a large community which will make it easier to debug it if we run into a problem.

For the implementation of our backend, we will use Firebase as it has a noSQL database. Firebase provides serverless hosting service and we will not need to worry about making our own server. Instead, we can focus more on application design and making the UI/UX of the application better. This was one of the major reasons for going with firebase. We plan on adding a website analytics ssection on the admin side of our application to allow track of real time analytics and firebase helps do that because it has Google Analytics integrated into it. The firebase testlab can help us in the testing by providing virtual and physical device access.

React bootstrap is another library which we will be using during our implementation. This library would allow us to include various components like for slideshow of our products, we can use a carousel component or to display an alert in a particular design, we use the alerts component of this library. This would help us to save time as these features have already been made, so we will simply reuse them. Other than that, if time permits, all functionalities would be implemented by ourselves. If there is a shortage of time, we would resort to reusing functionalities.

The user interface is a Graphical User Interface whereby the user interacts with the web application through icons, tabs, images, cards and so on. We have aimed to design a clean and appealing interface which is easy to learn and follow. The user will learn through the interactive web application on how to go about the system to purchase, browse or review products. The design decisions were made keeping the user in mind. The overall information architecture of the system was kept broad and shallow so that all the options the user can access are available throughout and the user will not have to make an effort in locating these options. The sidebar will be fixed at the left side of the screen so that he/she could access variety of options. The navbar also is kept fixed so that the user can always go back to a familiar place e.g. the homepage. The same rules have been followed for the admin side as well with the exception of the navbar. The sidebar will help the admin to locate himself/herself on the particular page that he/she is on.

The web application will be deployed on a hosting service. The hosting service we decided upon was Firebase hosting service. The main reason we chose this service was that it provides fast and secure hosting. Zero-configuration SSL is built into Firebase Hosting, so content is always delivered securely. We can host static and dynamic content as Firebase allows this. Other than that, resources for how to deploy a web application on Firebase are available to help guide us and save time.

4 System Architecture

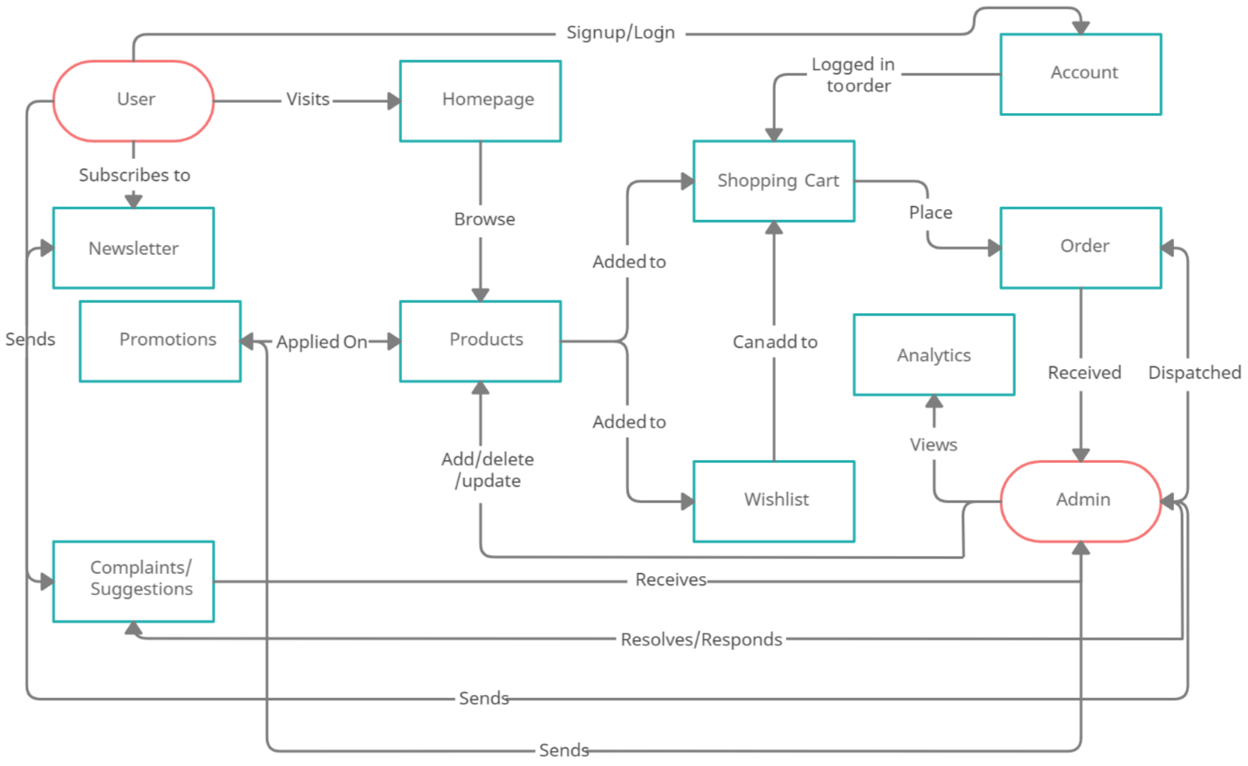
In our SRS document, we mentioned our top 5 use cases. Since we have to make different diagrams with respect to our top 3 use cases, at this point we would like to shortlist our use cases further and state our top 3 use cases. They are:

1) Search product (User Side)
2) Add to Cart (User Side)
3) Add Products (Admin Side)

The diagrams shown below are with respect to these above-mentioned use cases.

4.1 System Architecture

4.1.1 Component Diagram



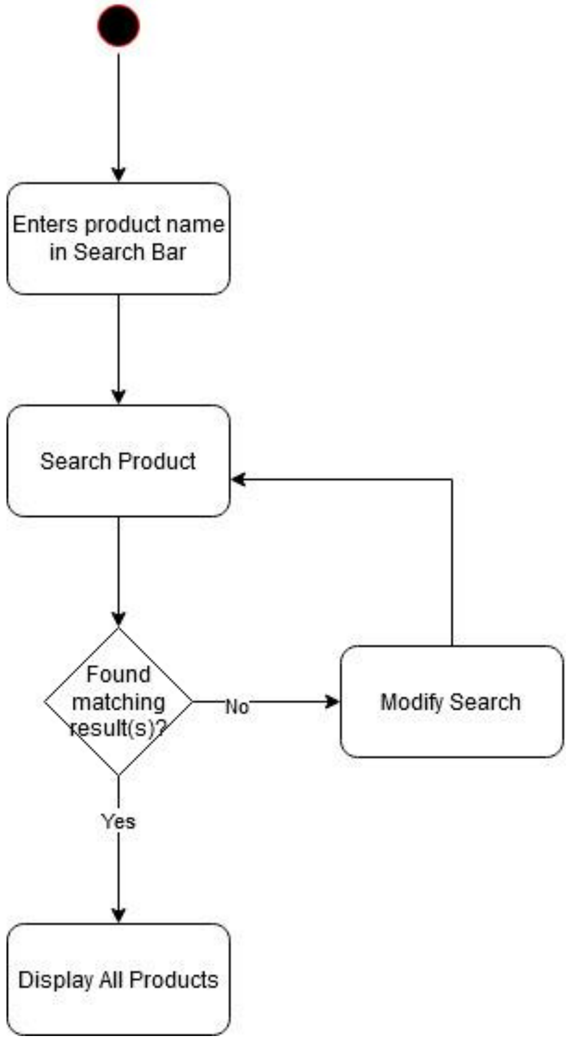
The above component diagram describes the organization and wiring of the physical components of the web application i.e. StoreX. There are two primary actors i.e. the user and the admin.

The **homepage** is the main page of StoreX where the user will be directed to once he/she opens the application in the web browser. This page consists of other components such as **products** which the user can browse. Various **promotions** are applied on different products. If he/she wishes to purchase something, the user adds the product(s) to the **shopping cart**. In order to complete the transaction and proceed to checkout, the user must be logged in. If the user, already has an **account**, he/she logs in by entering the relevant details in the log in page. If the user does not have an account, he/she creates one by following the instructions on the create account page. Once the user has created the account, he/she can log in to complete the transaction. The user, now can place his/her **order** and this order will be received by the admin. In case the user wants to purchase a product at a later date, then he/she can add it to his/her **wishlist**. If the user wants to purchase something from his/her wishlist, he/she is able to add products from the wishlist to the shopping cart. If the user wishes to subscribe to the store's **newsletter**, he/she is able to do this. If the user feels that something is off regarding the service or experience of browsing through our store, he/she will leave **complaints** or **suggestions**. These complaints and suggestions will be sent to the admin for review and the user will be notified once they have been resolved or considered respectively.

The admin side also has various components. The admin can **add/update** products in case there is a new arrival or some price has changed or new stock has arrived. He/she can also **delete** products in case they are discontinued. The admin can add **promotions** to existing product so that the user is more interested in purchasing that particular product. The **orders** received by the admin from the user will be reviewed and then dispatched. He/she will also notify the user that their order has been dispatched. The admin can view the overall **analytics** of the website i.e. how many users are currently navigating the store, total number of users that have created an account etc. The sales analytics will help the admin to keep track of all the transactions that have happened and will also help him/her to monitor the monthly or quarterly sales. The admin will review the **complaints** and **suggestions** received from the users and will resolve those complaints or consider the suggestions and will notify the users that the complaint has been resolved or the suggestion has been considered by him/her or not. Finally, the admin can send **newsletter** to all the subscribed users, luring them into visiting the website.

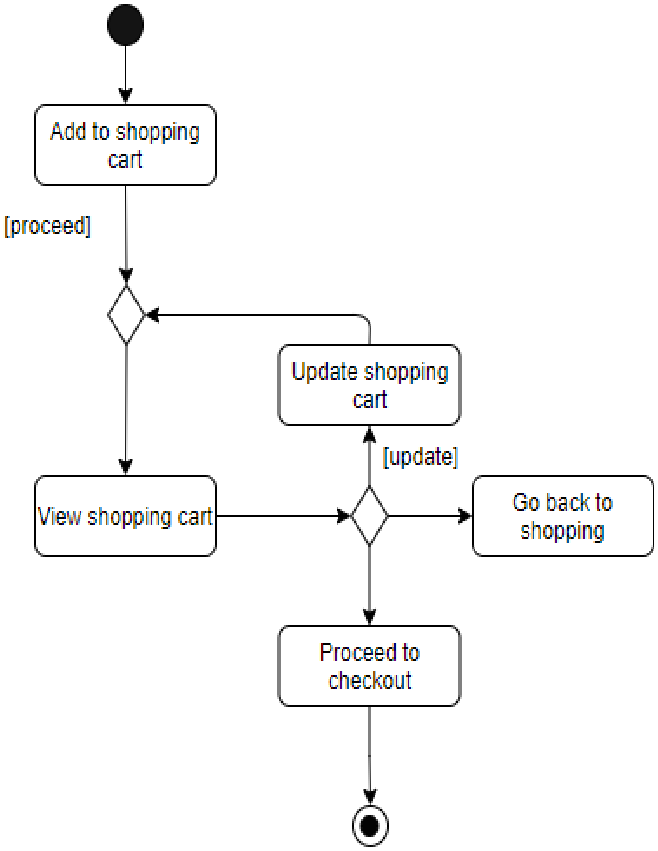
4.1.2 Activity Diagrams

4.1.2.1 Search Product



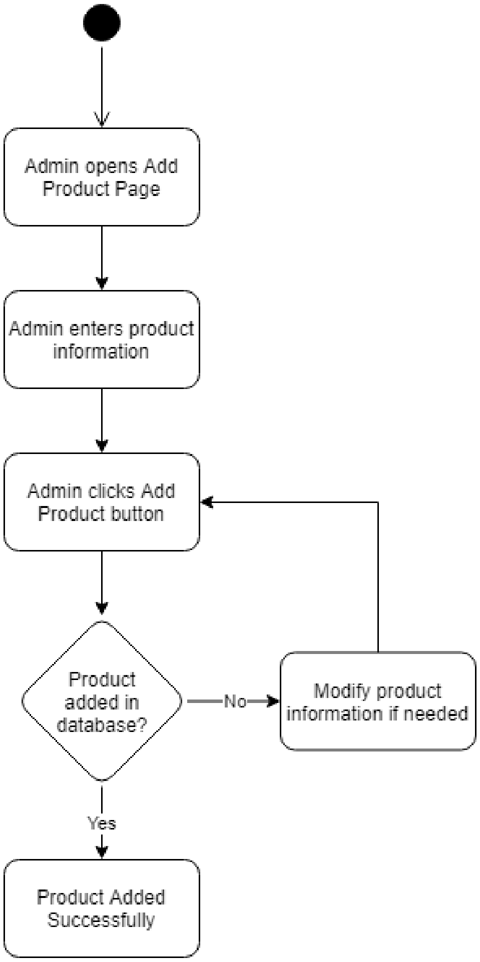
The user enters the name of the product they want to search for in the Search Bar, and clicks the search icon next to it. The web app will search for all products which have that name either in their category or product names, and will return matching results. If no matching results are found, the user can modify their search and then click search again. If there are matching results found, they will all be displayed to the user.

4.1.2.2 Shopping Cart



The user can add specific product(s) to their shopping cart as they are browsing through StoreX. They can view their shopping cart at any given time by clicking on the cart icon on the top right. They can update the quantity of items inside the cart too by clicking + or – next to their respective quantities. From there, they can either go back to shopping and browsing other products, or they can proceed to checkout and consequently place their order.

4.1.2.3 Add Products - Admin

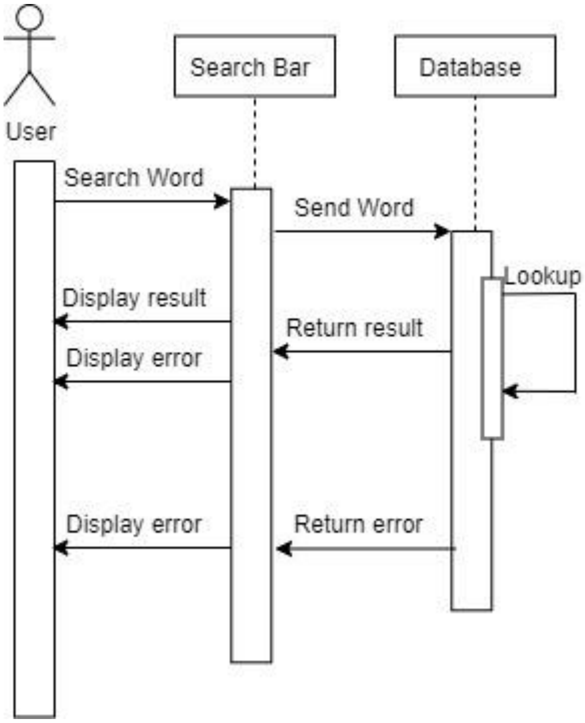

Admin opens Add Product Page
Admin enters product information
Admin clicks Add Product button
Product added in database?
No
Modify product information if needed
Yes
Product Added Successfully

The admin can add new products in StoreX by opening the Add Product page from their admin navigation bar. The admin can then add the product information and details in the given fields, and then click the Add Product button. If the product is successfully added into the database, it shows a dialog box indicating ‘Product Added Successfully’. If not, the dialog box indicates an error in adding the new product, that is, it could not be added successfully. The admin can modify the product information if needed, and can then try to add the product again by clicking on the Add Product button again.

4.2 Subsystem Architecture

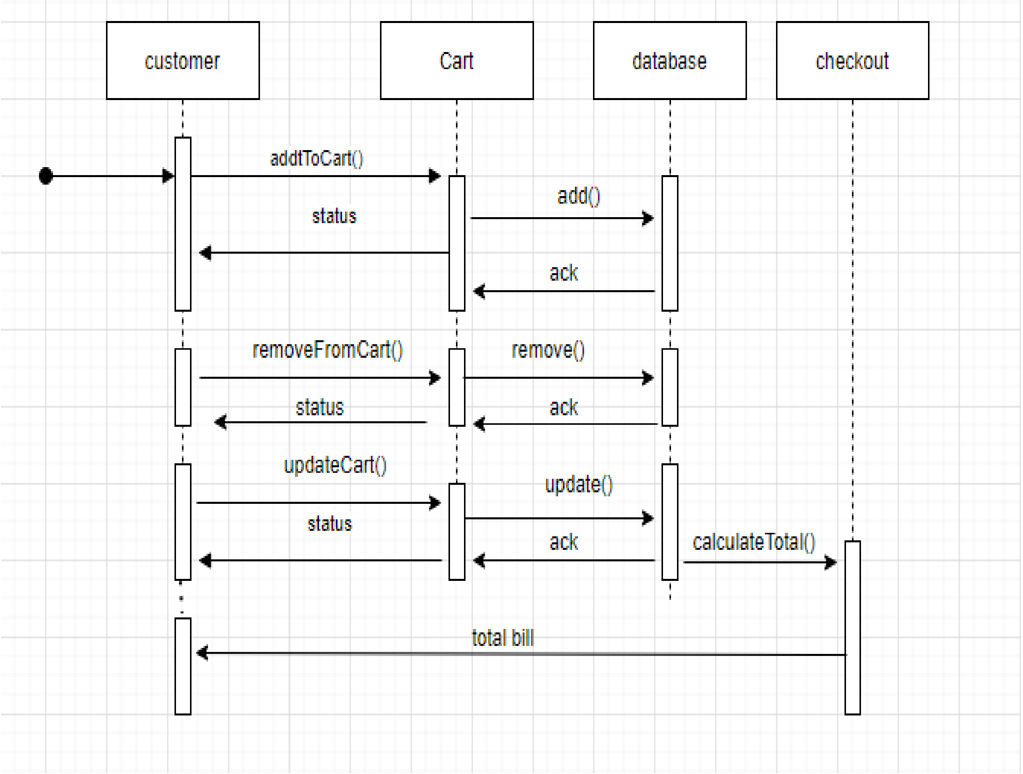
4.2.1 Sequence Diagrams

4.2.1.1 Search Product



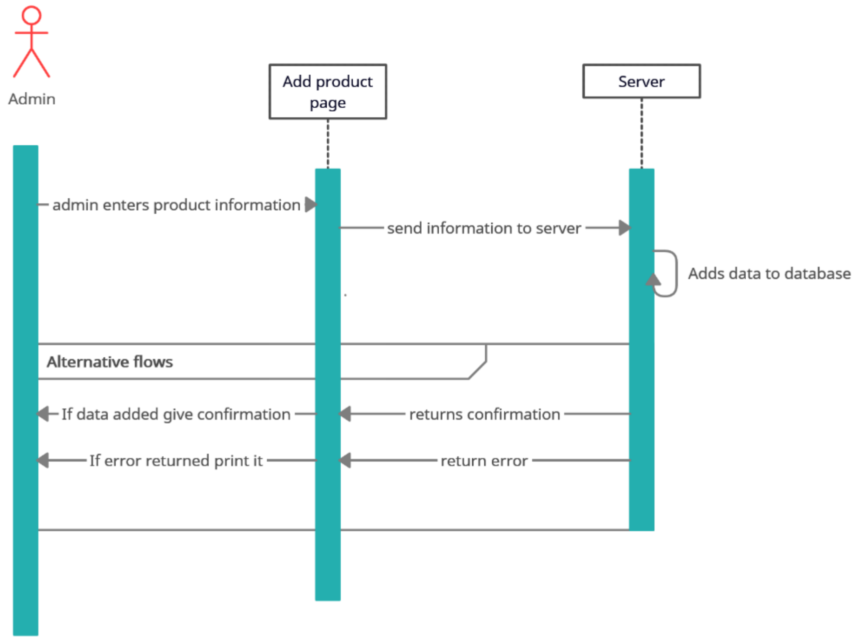
The user searches for a particular product by writing word(s) in the search bar. The entered word is sent to the database to lookup if any product or category contains that word. If there is, the results are returned to the front end, where they are displayed to the user. If no results are found, it is displayed that no matching results were found. If there is any other error while looking up the word in the database, that error is returned and displayed to the user too.

4.2.1.2 Shopping Cart



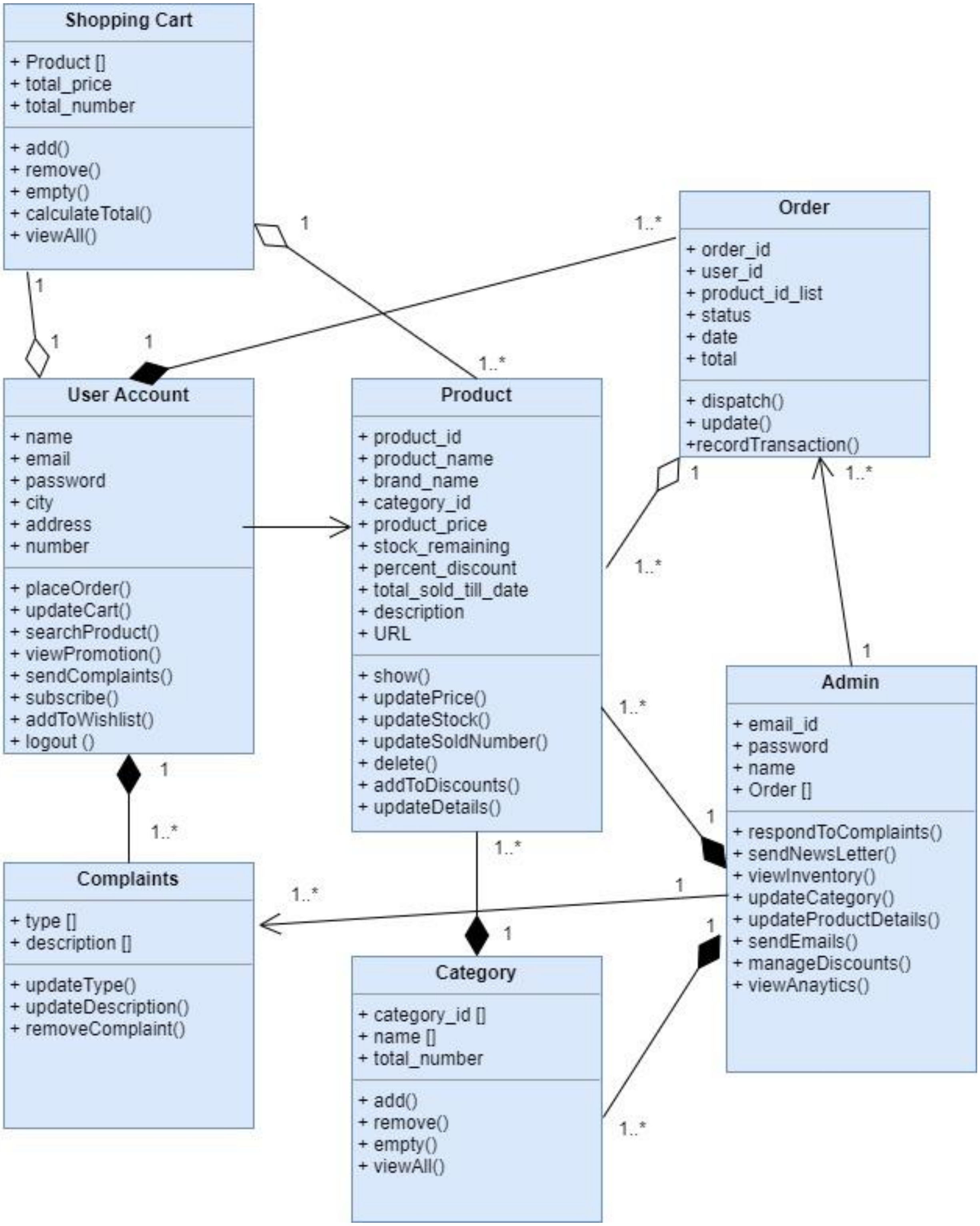
The user can click on Add to Cart button for a particular product to add it to their shopping cart. The product is then added to the cart and the status is returned to the user in the form a number indicator on top of the cart icon (which increments when a product is added to the cart). The added product is also added to the database to store which products have been added into the cart by a specific user. Also, the user can remove products from their cart, upon which it is also removed from the database (where the products in cart are stored for that user), and this is then returned to the user by visually removing the product from the displayed items in the cart. The cart can be updated to by changing the quantities of the products added in the cart, upon which the database is updated with the new quantities too, and the user can visually see the quantities change too, along with the total bill updating too. On checkout, the final total bill, which is calculated by summing up the added products’ prices multiplied by their respective quantities, is displayed to the user too.

4.2.1.3 Add Products - Admin


Admin
Add product page
Server
admin enters product information
send information to server
Adds data to database
Alternative flows
If data added give confirmation
returns confirmation
If error returned print it
return error

The admin can add product information for a new product by adding the details in the respective fields in the Add Product page. This entered information is sent to the backend server, which adds this data to the database where product information is stored. If it is added successfully, the server returns a confirmation, which is displayed to the admin in the form of a Product Added Successfully dialog box. If there is an error in adding the details of the new product to the database, the server returns an error, which is also displayed to the admin in the form of a dialog box.

4.2.2 Class Diagram



The above diagram shows how different classes of the system will be linked with each other, with each class representing a single system item/component/use case. The product class is the central most class of the system and is connected to almost every other class. It has a one-to-one aggregation relationship with the shopping cart i.e. a product within the system (which is representing an individual item within our store) can exist without a shopping cart but is added to the shopping cart in order to complete the functionality of shopping cart i.e. when its added to shopping cart, it loses its own entity and the product and the cart become a single entity i.e. a shopping cart (This is coherent with the definition of aggregation). The product class also has an aggregation relationship with the order class as they become a single entity when combined and yet, they are independent and stand alone when they are separate from each other. The product class also has a composition relationship with the category class as a product cannot survive standalone if it hasn't been assigned to a category. It also has a composition with the admin as the admin is the only one who can add and delete the product. Without the admin's approval, no product can exist within the system.

Now, the relationships of other classes will be discussed. The shopping cart has a 1 to 1 aggregation relationship with the user account i.e., you can add items into the shopping cart without even being a user. Shopping cart still exists independently. At that time, it's a random shopping cart not owned by anyone. But as soon as the user logs in, the cart is owned by the user. The complaints class has an association with the user account because it cannot exist without a user. If there is no user, there are no complaints. The category class like the product class also has a composition relationship with the admin class. These classes and their relationships overall demonstrate how our system is linked together.

4.3 Data Structure

4.3.1 Internal software data structure

Data structures that are passed among components of the software are described. The data between the software components transferred in the form of JSON files. The server and client communication is through the JSON files so these are used to communicate between different components.

4.3.2 Global data structure

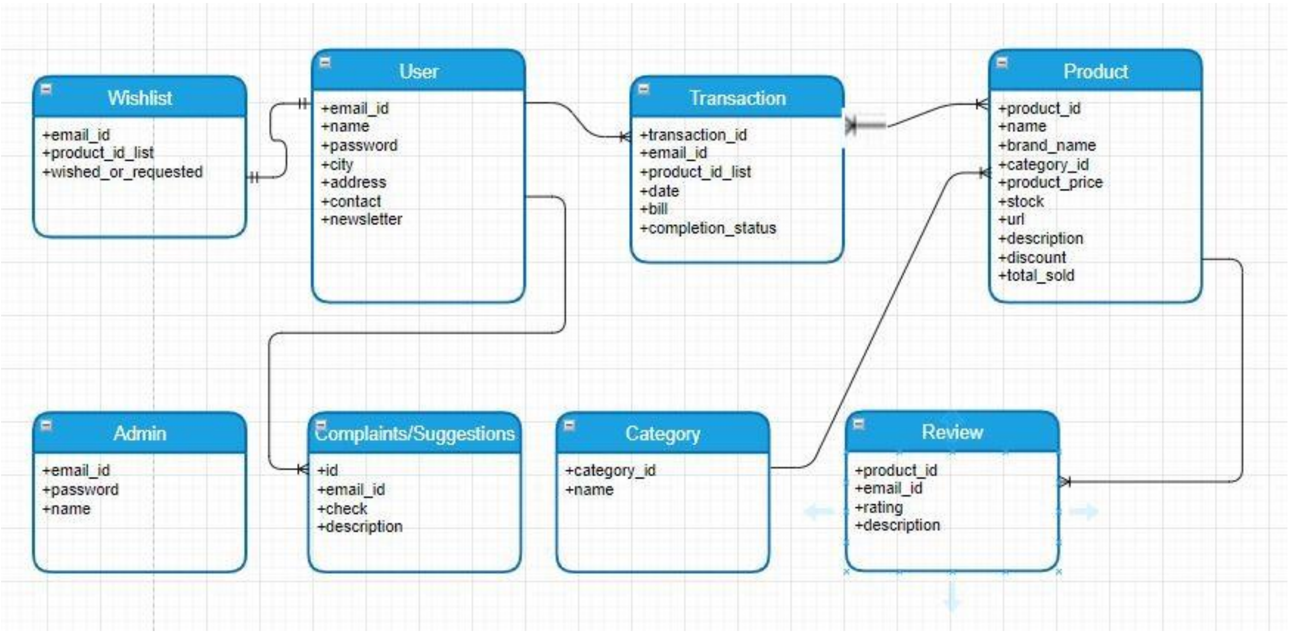
Data structured that are available to major portions of the architecture are described. The internal data is organized as NoSQL using Firestore. All the data in the store is contained in documents which are similar to JSON trees. This allows for scalable database with fast queries.

4.3.3 Temporary data structure

Files created for interim use are described. The temporary data structures are the JSON files used to send information from the web browser to the server. These files contain the information for the queries that are to be performed by the server.

4.4 Database Model

4.4.1 Database scheme and detailed description



Description: We have a total of 8 tables in our database. They are described below

User: The user table has 7 different attributes. The email_id is the email address with which the user created their account on StoreX, the name is the name with which they registered on our platform, the password is their account's password, the city is the name of the city where they currently reside. The city name is being asked from the user to be stored for the purposes of delivering them the product that they order. The address is the address where they would want the product that they ordered to be delivered, contact is their contact number and newsletter is a Boolean variable that describes whether they have subscribed to our newsletter or not. In this table, email_id is the unique identifier of every user (Primary Key). The user can change these settings as well in the account settings page.

Wishlist: The Wishlist table has 3 attributes. The email_id is the user's email id. The product_id_list is the list of all the products that the user has added to their wishlist. The wished_or_requested is a Boolean list corresponding to the product_id_list. It has the same size as the product_id_list and each corresponding entry within the wished_or_requested list tells whether that particular product has been wished by the user or requested by the user. If the product in the wishlist is out of stock, it is assigned the label "requested", otherwise it is assigned the label "wished". The email_id is both a primary key and a foreign key here.

Transaction: The transaction table has 6 different attributes. The transaction_id is the unique identifier used to distinguish transactions from eachother, email_id is the user's email id, product_id_list is a list of tuples, with the first element of the tuple referring to the unique id of the product ordered, and the second element of the tuple referring to its quantity. The date refers to the date on which the transaction took place, bill is the total amount of the transaction and completion status is a Boolean variable that becomes true when this particular order is dispatched by the admin, otherwise remains false. transaction_id is the primary key.

Product: The product_id refers to the unique identifier of a product, name refers to the name of the product, brand_name refers to the brand of the product and if the product has no brand, by default, this attribute is set to NULL. category_id refers to the category to which the product belongs to, product_price refers to the price of the product, stock refers to the total remaining products with product_id in the inventory, url is the site where the image for a product is located, description is the description of the product, discount is a Boolean variable that tells whether a discount is applicable on the product or not, and total_sold is the total number of that product sold till date, which we will use to find out the top selling products. product_id is the primary key.

Complaints/Suggestions: The complaints/suggestions table has 4 attributes. id is the unique identifier of that particular complaint/suggestion. email_id is the email address of the user sending that complaint/suggestion to the admin, check is a Boolean variable to check if the type of feedback received was a complaint or a suggestion, and description are the actual comments left by the user for the admin. id is the primary key.

Category: Category has 2 attributes. category_id refers to the unique identifier of a particular category and name is the name of that category. Category_id is the primary key.

Review: Review table has 4 different attributes. product_id and email_id combined form a unique identifier (Primary key) in this table. rating is the rating given by the user having email_id to the product having product_id, out of 5. Description are any comments that the user wants to leave for a particular product.

Admin: The admin table is used to store the email_id, password, and name of the admin. It will only have one record in it.

Relationships: There is a one-one relationship between a user and their Wishlist because a single user can have only one wishlist. There is a one-many relationship between the User table and the Complaints/Suggestions table because a single user can submit multiple complaints/suggestions. There is a one-many relationship between the User table and the Transaction table because a single user can carry out multiple transactions. There is a many-many relationship between the transaction table and the product table as one transaction can have multiple products and one product can be a part of multiple transactions too. There is a one-many relationship between the product table and the review table because one product can get multiple reviews. There is a one-many relationship between product and category because a single category can have multiple products. The admin table is a standalone table and does not directly have a relationship with these other tables because there is only one record in it to identify one admin. However, the admin has the authority to add/delete products, categories, view suggestions/complaints and view transactions etc.

4.4.2 Database

We will be using firebase database due to the following reasons:

1) It’s a Nosql database. This gives us the advantage of avoiding the long and complicated SQL queries and hence this approach has 3 advantages:

a) We can avoid the human error of accidentally typing the wrong queries and making our database do actions that we did not intend or perform no action at all in case of a logically incorrect query.

b) We can avoid having to manually secure our system against SQL injection attacks and hence we don't need to worry about doing that.

c) We can avoid wasting our time trying to optimize queries as we will no longer have to write the queries our self.

2) Firebase provides serverless hosting service and we will not need to worry about making our own server. Instead, we can focus more on application design and making the UI/UX of the application better. This was one of the major reasons for going with firebase.

3) The firebase testlab can help us in the testing by providing virtual and physical device access. It can provide us with screenshots as test results or we can even use their auto crawler bot for crawling the application, finding bugs etc.

4) We plan on adding a website analytics section on the admin side of our application to allow track of real time analytics and firebase helps do that because it has Google Analytics integrated into it.

Keeping in mind these advantages of Firebase, we decided to use it for our hosting.

4.5 External Interface Requirements

4.5.1 User Interfaces

The application is going to be implemented as a graphical user interface. The interface on the main home page will be very easy to use and user friendly such that all the categories of products will be displayed as a list with an icon next to the category of each product for the user to be able to better understand the types of items that they will be able to find in that particular category. There will also be different buttons throughout the homepage as well as other pages that will lead the user from one page to another when clicked. These buttons will be clearly visible so that the user can click them when required and reach their required page. The remaining left over items will also be displayed in front of each product's details. For giving feedback the user can use the input fields and text boxes. For sections involving multiple options, the application uses dropdown menus. There is a fixed navbar that the user can use to navigate to different sections of the webpage. The navbar also highlights the section on which user currently is.

4.5.2 Hardware Interfaces

The user side of the application will follow a responsive design and will be able to run on desktop, tablet and mobile devices provided that they have a modern web browser and at least 2GB ram for ensuring smooth performance. The system can run on multiple operating systems including windows, macOS, Linux etc. However, the admin component of the system will not be responsive and therefore will only be supported on a desktop computer. The systems must support peripherals such as mouse, keyboard, touchpads or touch screen for interacting with the web application. The main software libraries that the application will be using are nodejs firebase client, express.js and react.js.

5 User Interface Design

5.1 Description of the user interface

Our team decided to use React as the tool to make the front end. We came to this decision because of our familiarity with this framework. This would allow us to spend less time on learning the framework and more on implementation. Since react is the most popular front-end framework, it has a large community which will make it easier to debug it if we run into a problem.
Another reason for picking react was its performance. React provided good performance for small relatively static web applications, which will be similar to our product. Performance is important since it will play an important role in user experience. Performance includes the small download size and faster webpage loading.

5.2 Information architecture

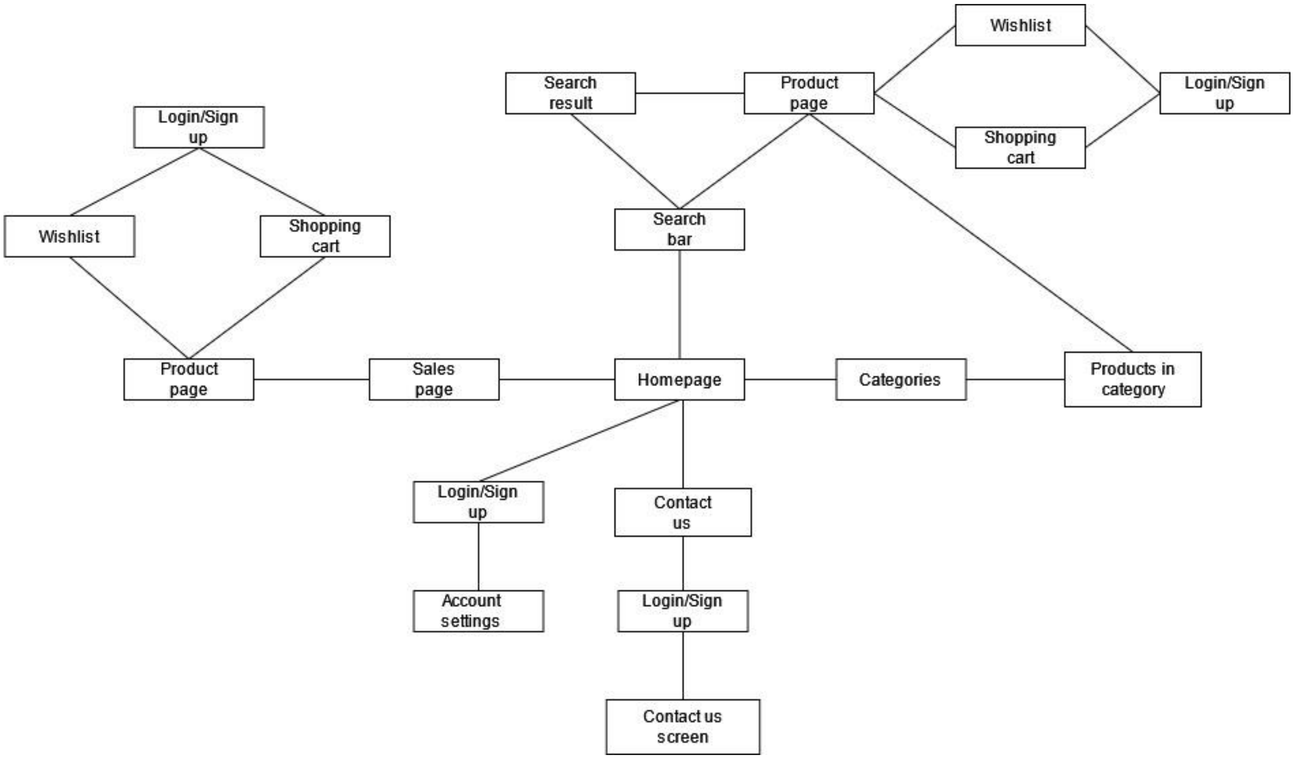


Fig. 5.2.1 User side Information Architecture

When the user accesses the web app, he/she sees the homepage. From the homepage the user has many options. The user can search an item using the search bar, which will return a list of items. Clicking on an item will lead to the product page for that item. From there the item can be added to Wishlist or cart. On the homepage the user can also see the sales, which list the items currently on discount. Homepage also has the button to contact the admin of the webapp. They will be asked to login/sign up and then will be able to send a suggestion or complaint to the admin. There are also categories listed on the sidebar. The user can click any of the categories and he/she will be given the list of products in that category. Lastly adding item to cart, Wishlist and checking out requires an account so before that action is completed the users are asked to sign up/login.

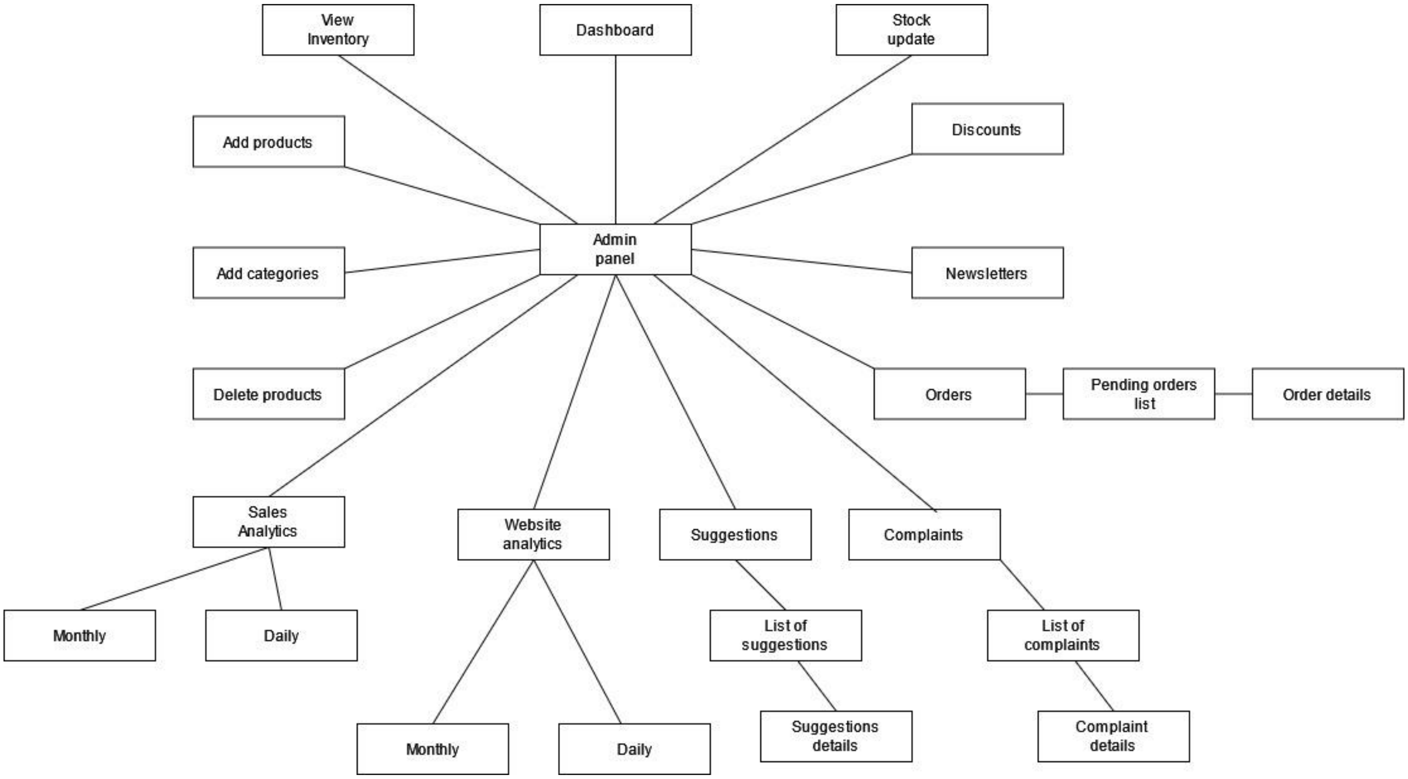


Fig. 5.2.2 Admin side information Architecture

The admin after logging in will be shown the dashboard by default. All other functionalities can be accessed by the sidebar. These include stock update, setting discounts, sending newsletters, see pending orders, list of complaints or suggestions, website analytics, sale analytics, adding or deleting products, adding categories and viewing inventory. Functionalities such as orders, complaints and suggestions at first show all the list of items of interest. Details of each item can be seen after clicking it. While sales analytics and website analytics give admin the option of seeing the monthly or daily analytics.

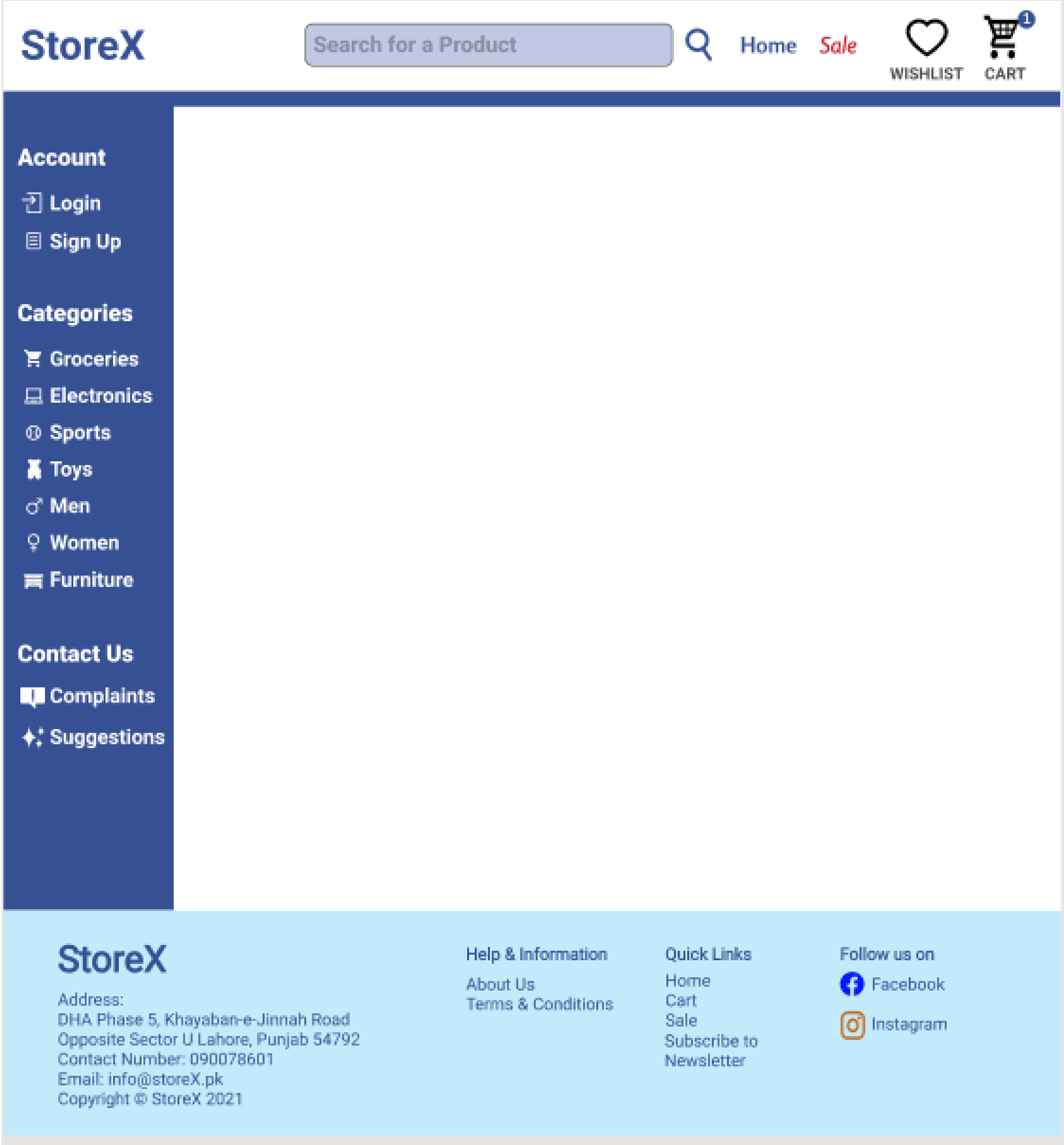
5.3 Screens

The following table presents the formal screen names that we will be using to refer to our screens from here onwards:

Screen Name	Type	Brief Description
Layout	User	Screen that shows the Layout to be followed by all screens on the user side
Homepage	User	Main screen where the user will be directed first when they visit our web application
Category ABC	User	Screen showing all the products of the category ABC which the store sells
Wishlist	User	Screen showing the user their Wishlist. Please refer to section 2.4 for the definition of Wishlist
Login and Signup	User	Screen where the user can either log in to their account or create a new account
My Orders	User	Screen where the user can view their pending and past orders, as well as their details
Account Settings	User	Screen from where the user can change their account details or delete their account
Shopping Cart	User	Screen where the user can view their cart before purchasing all the products present in it
Product Details	User	Screen where the user can see the details of the product they are interested in, and the user can write their own or see other reviews
Search Results	User	Screen that displays the results of what the user searched using the search bar
Footer	User	Screen showing how our footer will look like throughout
Contact Us	User	Screen where the user can share their suggestions/complaints with the admin
Sale	User	Screen where the user can see all the products that are currently on sale.
Sign in	Admin	Screen that allows an admin to access his/her dashboard
Dashboard	Admin	Screen that allows the admin to see a summary of all the pending orders, web application analytics
View Inventory	Admin	Screen that allows the admin to view the current inventory of the store
Stock Update	Admin	Screen that allows the admin to update stock of various products
Add Products	Admin	Screen that allows the admin to add new products to the store
Discounts	Admin	Screen that allows the admin to add discounts to the products
Add Category	Admin	Screen that allows the admin to add new categories of products
Delete Products	Admin	Screen that allows the admin to delete the products from the store
View Orders	Admin	Screen that allows the admin to view pending orders or all orders
Send Newsletter	Admin	Screen that allows the admin to send newsletters to subscribed users

Sales Analytics	Admin	Screen that allows the admin to view the transactions and sales analytics
Website Analytics	Admin	Screen that allows the admin to view the analytics of the web application
View Complaints	Admin	Screen that allows the admin to view and reply to complaints
View Suggestions	Admin	Screen that allows the admin to view suggestions made by users

5.3.1 **Layout** – User

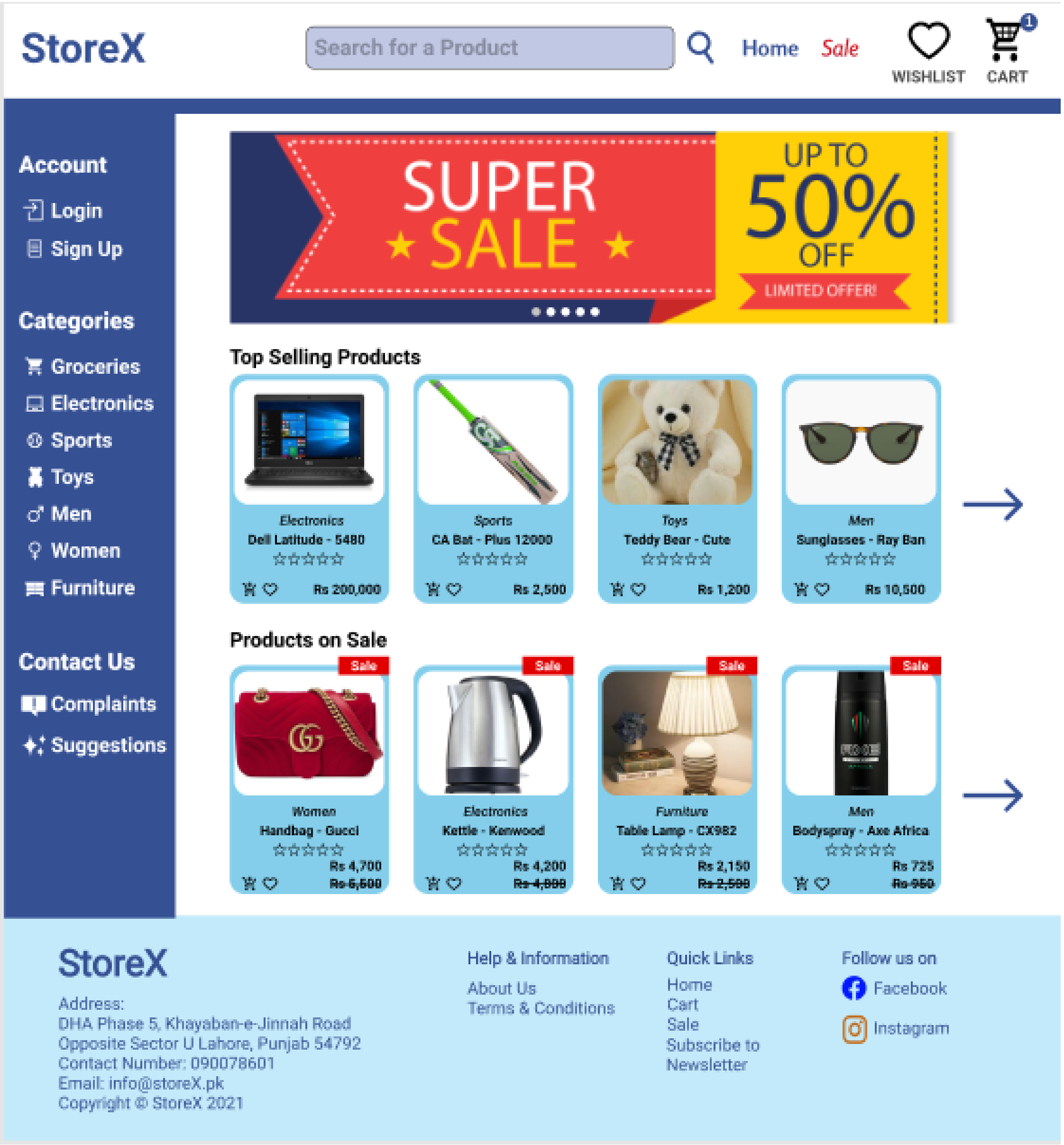


Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

The layout screen shows the design that we intend to follow while making all of our user screens. It starts off with a header from where the user can select certain options such as searching for a product by its name, going to the sales page by clicking on *sale*, viewing their Wishlist by clicking on the heart icon, or viewing their shopping cart by clicking on the cart icon. On the left side is the navigation bar which the user can use to navigate across different pages of the web application. The options within the navigation bar can be seen on the screen image. At the bottom of the screen, there is a footer with the address and contact information about our store, quick links to important webpages within the site and a social media information section.

Note: All other screens on the user side follow the same template as the layout screen.

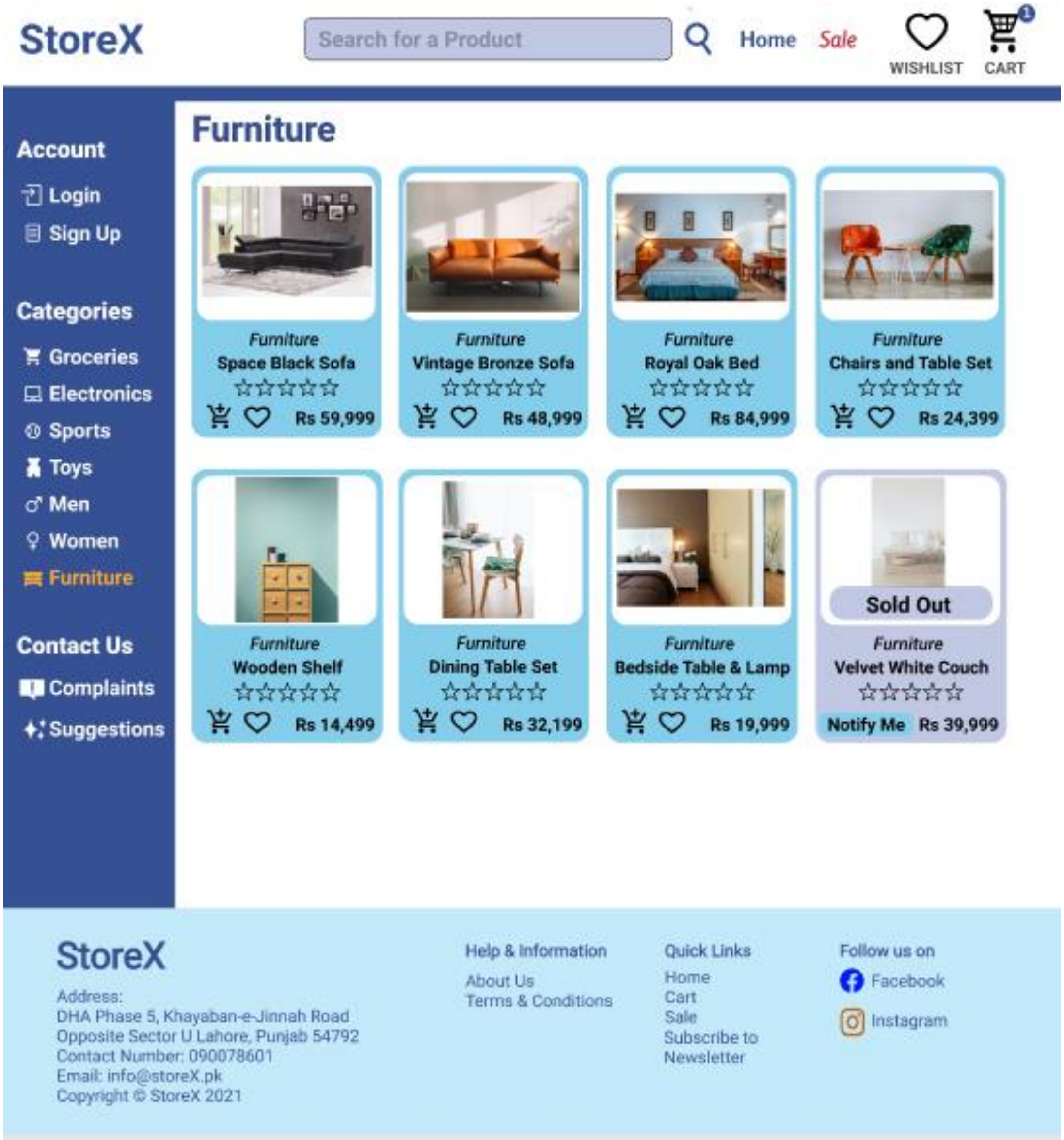
5.3.2 **Homepage** – User



At the top of the home screen below the header, we have displayed a slideshow. The slideshow displays different ads and promotions of products and discounts offered by our store such as sale, product of the day etc. It is a timed slideshow and will move to the next ad after every 5 seconds. Below the slideshow, we have displayed a section for our *top selling products* which can be swiped sideways as shown by the arrow to see more products than the ones already shown on the screen. Below that, we have a similar *products on Sale* section that shows the most sought out products with sale. This section is different from the Sale section because it shows the sale of only the most sought out products whereas the sale section shows the sale of all the products. This user can swipe left in this section as well to view more products.

Note: The user can scroll down below to view different sections of the homepage which have been shown as a single screen within the image currently.

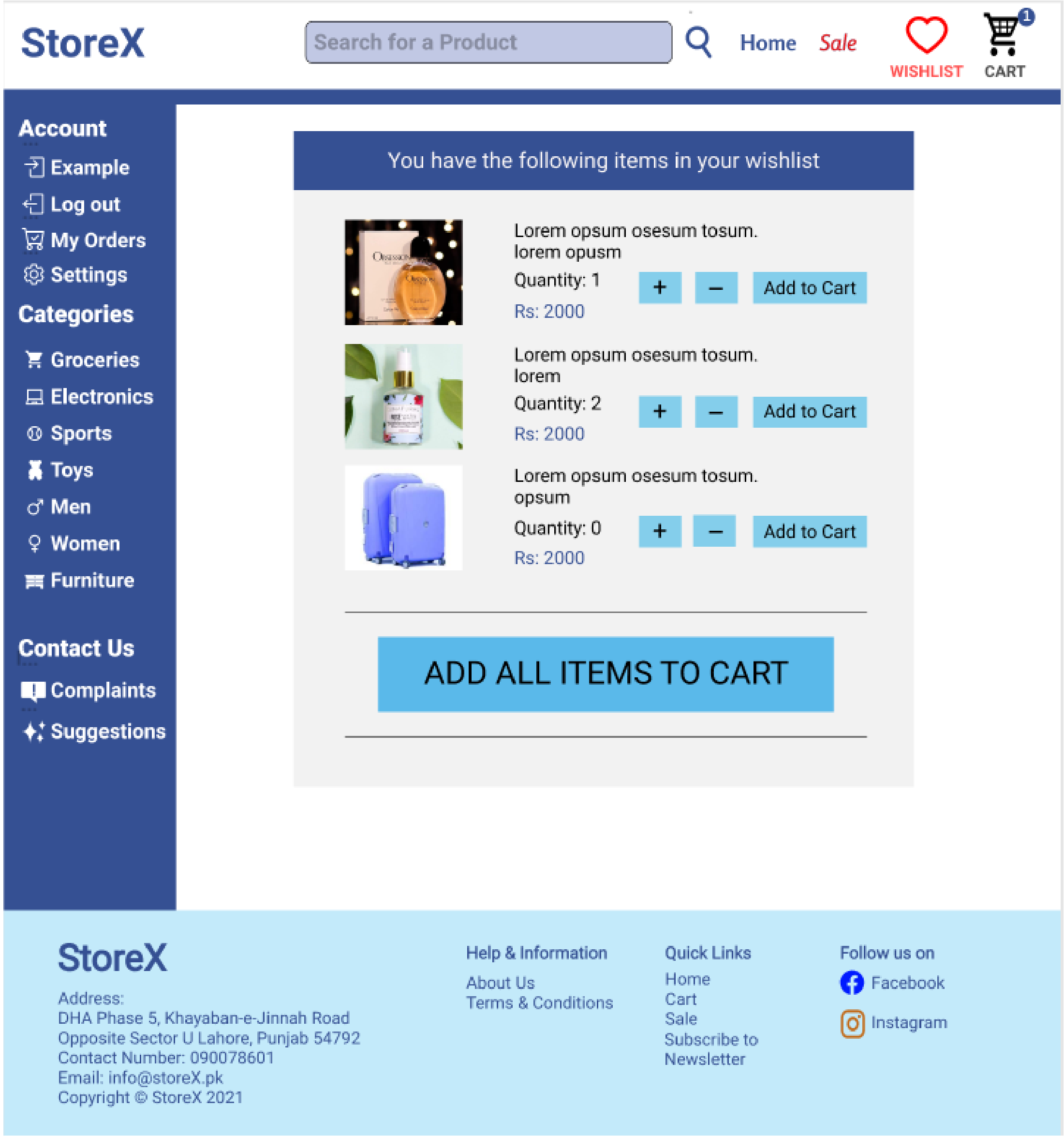
5.3.3 Category ABC – User



The user can open a particular category ABC's page by clicking on a category in the left navigation bar. The category page will display various products of that particular category, and the user can view their prices and ratings too. The user can add products of their choosing to their cart or wishlist.

Note: If a product (on any screen where it is visible) is sold out/out of stock, it will be displayed in a greyed-out manner with a Sold Out label. There will be a button which the user can click to request for that product, and hence they will be notified through email when that particular product is back in stock.

5.3.4 **Wishlist** – User



You have the following items in your wishlist

Lorem opsum osesum tosum. lorem opusm
Quantity: 1
Rs: 2000

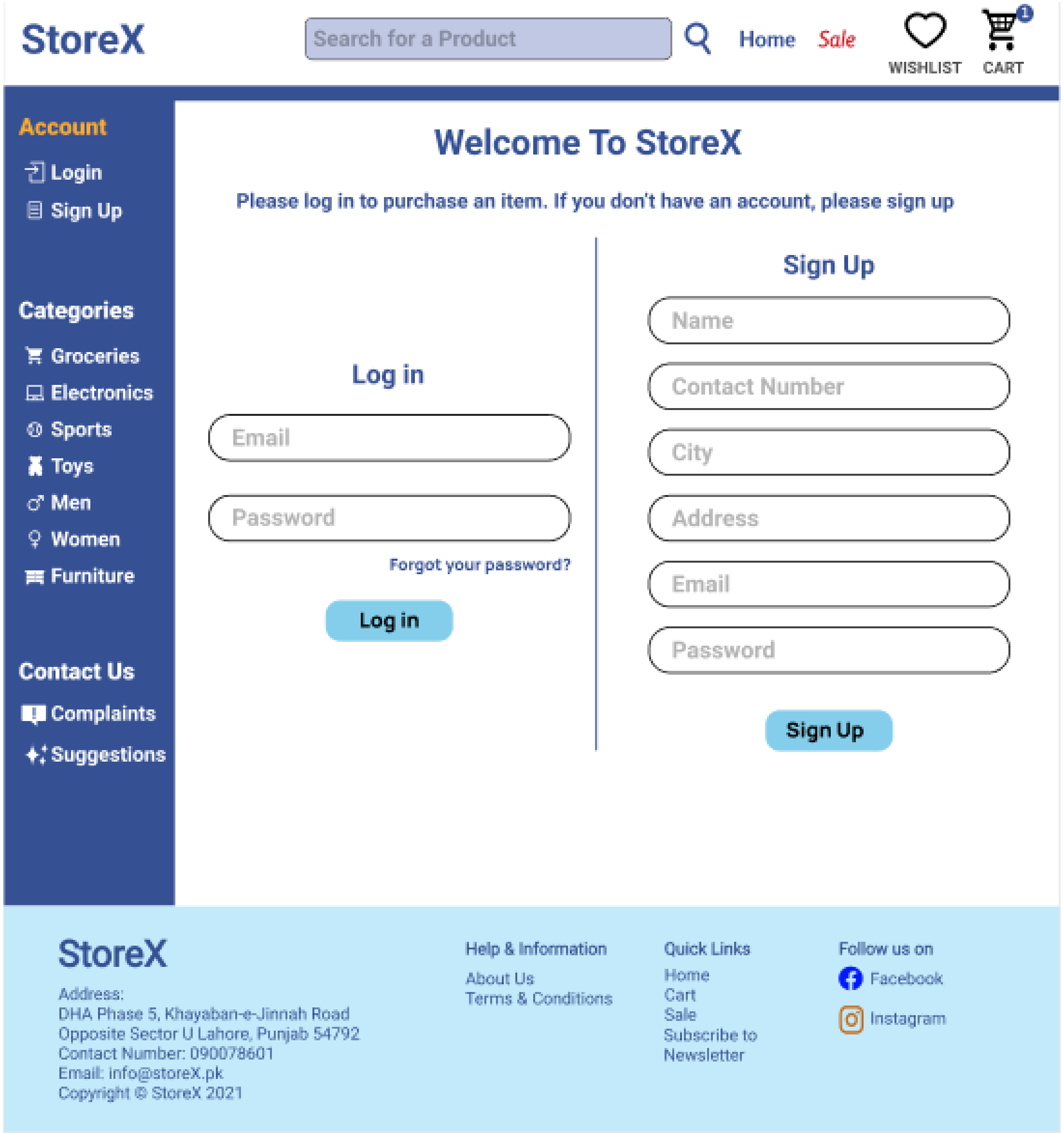
Lorem opsum osesum tosum. lorem
Quantity: 2
Rs: 2000

Lorem opsum osesum tosum. opsum
Quantity: 0
Rs: 2000

StoreX
Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

In this screen, the user can view all the products that they have added into their wishlist. The products in the wishlist can be added to cart individually by clicking on the add to cart button next to each product. Alternatively, if the user wishes to add all products in the wishlist to their shopping cart, they can click on Add All Items to Cart button to do so. Also, if there is a sale on any product that the user has added in their wishlist, then they will be notified of that sale through email.

5.3.5 **Logi**n and Signup – User

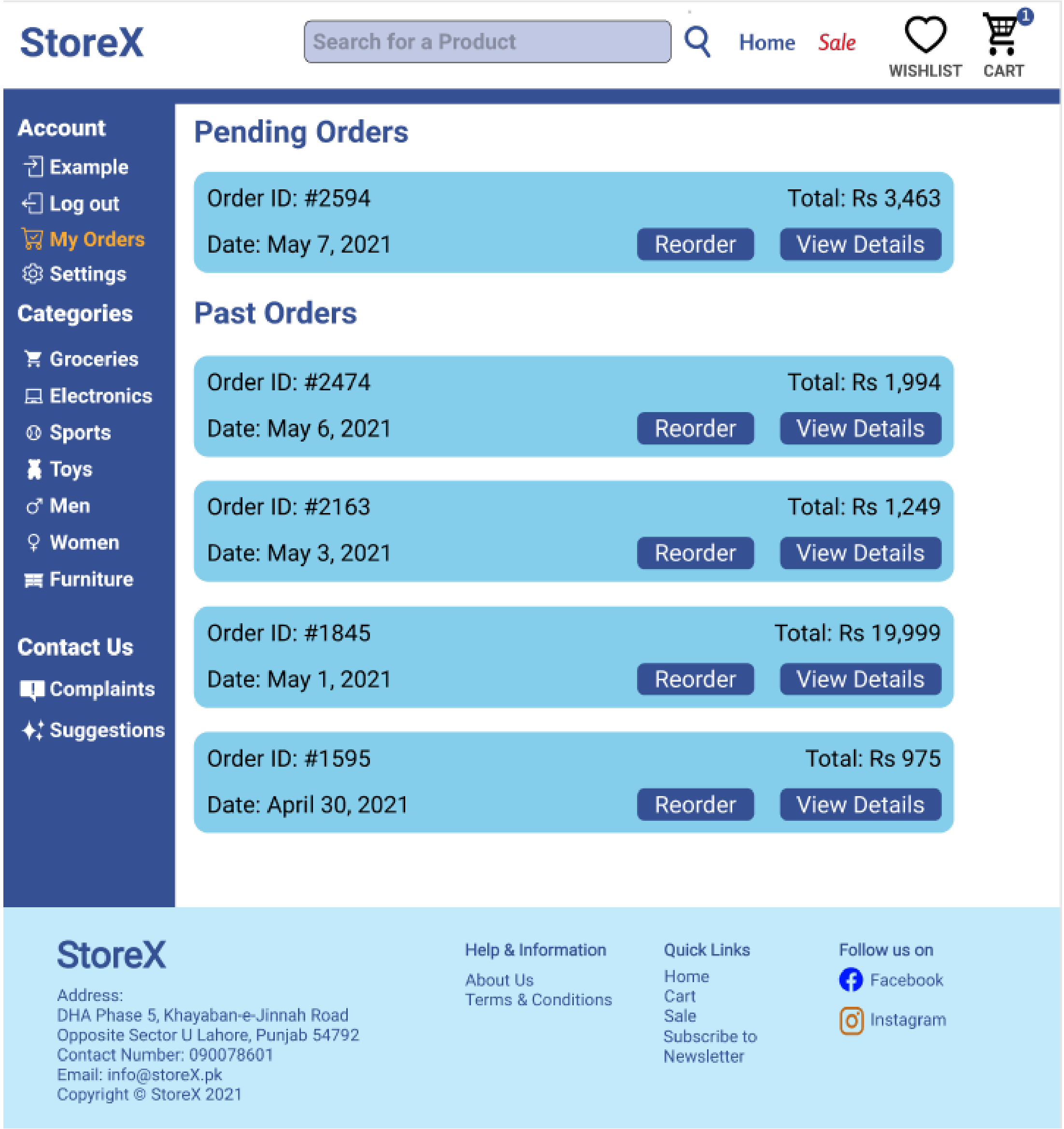


Please log in to purchase an item. If you don't have an account, please sign up

StoreX
Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

There are two sides of this screen. The left side has the option for logging in where the user is required to enter their email and password for logging in. The user can click the “Forgot your password?” option if they forget their password. The right side has the signup section where the user will need to enter their name, contact number, city, address, email and password to create their account. When the user clicks signup after entering this information, their account is created.

5.3.6 **My Orders** – User



Pending Orders

Order ID: #2594 Total: Rs 3,463
Date: May 7, 2021

Past Orders

Order ID: #2474 Total: Rs 1,994
Date: May 6, 2021

Order ID: #2163 Total: Rs 1,249
Date: May 3, 2021

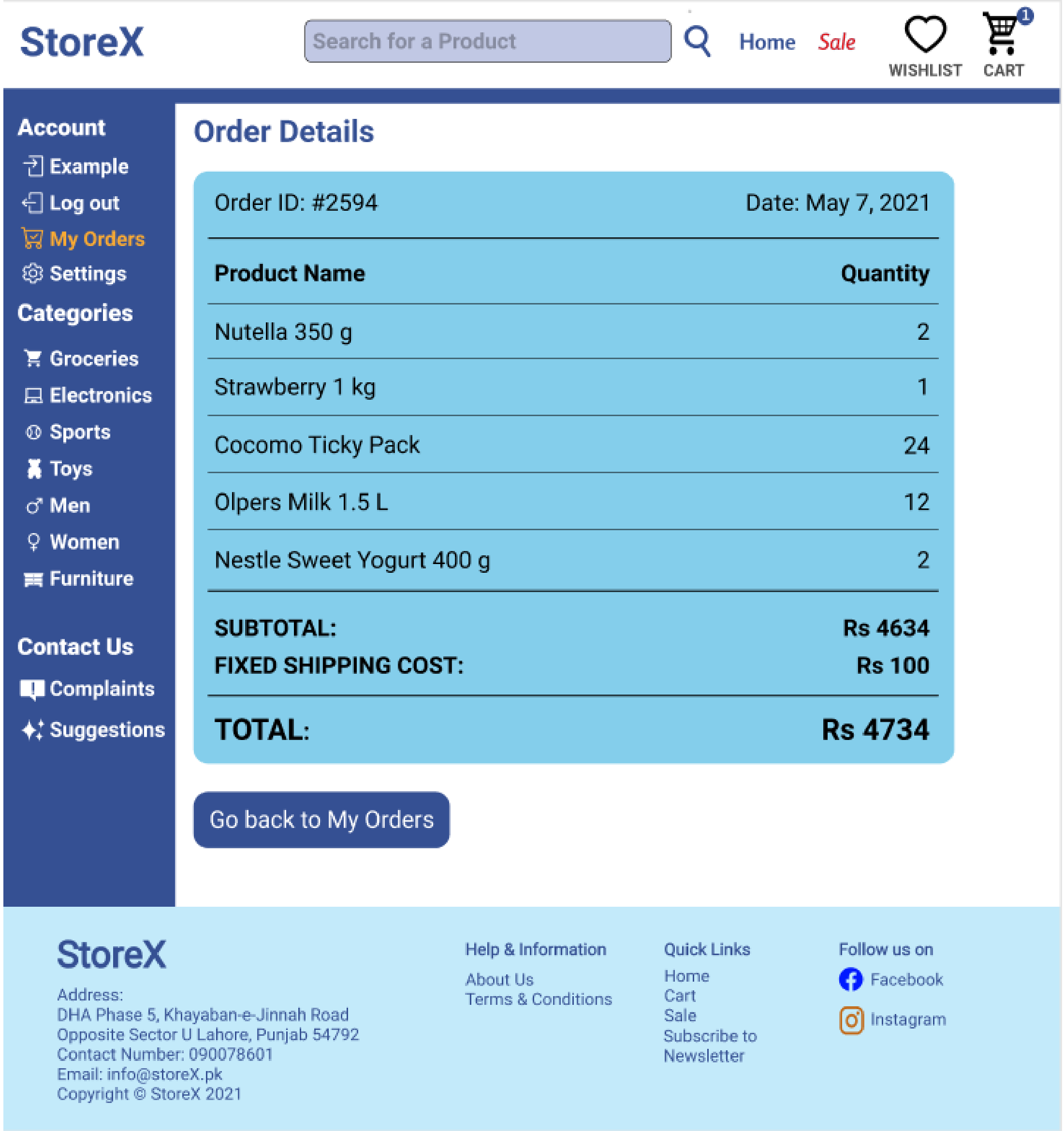
Order ID: #1845 Total: Rs 19,999
Date: May 1, 2021

Order ID: #1595 Total: Rs 975
Date: April 30, 2021

StoreX

Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

Fig 5.3.6 (a) My Orders Main Screen



Account
- Example
- Log out
- My Orders
- Settings

Categories
- Groceries
- Electronics
- Sports
- Toys
- Men
- Women
- Furniture

Contact Us
- Complaints
- Suggestions

Order Details

Order ID: #2594 Date: May 7, 2021

Product Name	Quantity
Nutella 350 g	2
Strawberry 1 kg	1
Cocomo Ticky Pack	24
Olpers Milk 1.5 L	12
Nestle Sweet Yogurt 400 g	2

SUBTOTAL:	Rs 4634
FIXED SHIPPING COST:	Rs 100
TOTAL:	Rs 4734

Go back to My Orders

StoreX

Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

Help & Information
About Us
Terms & Conditions

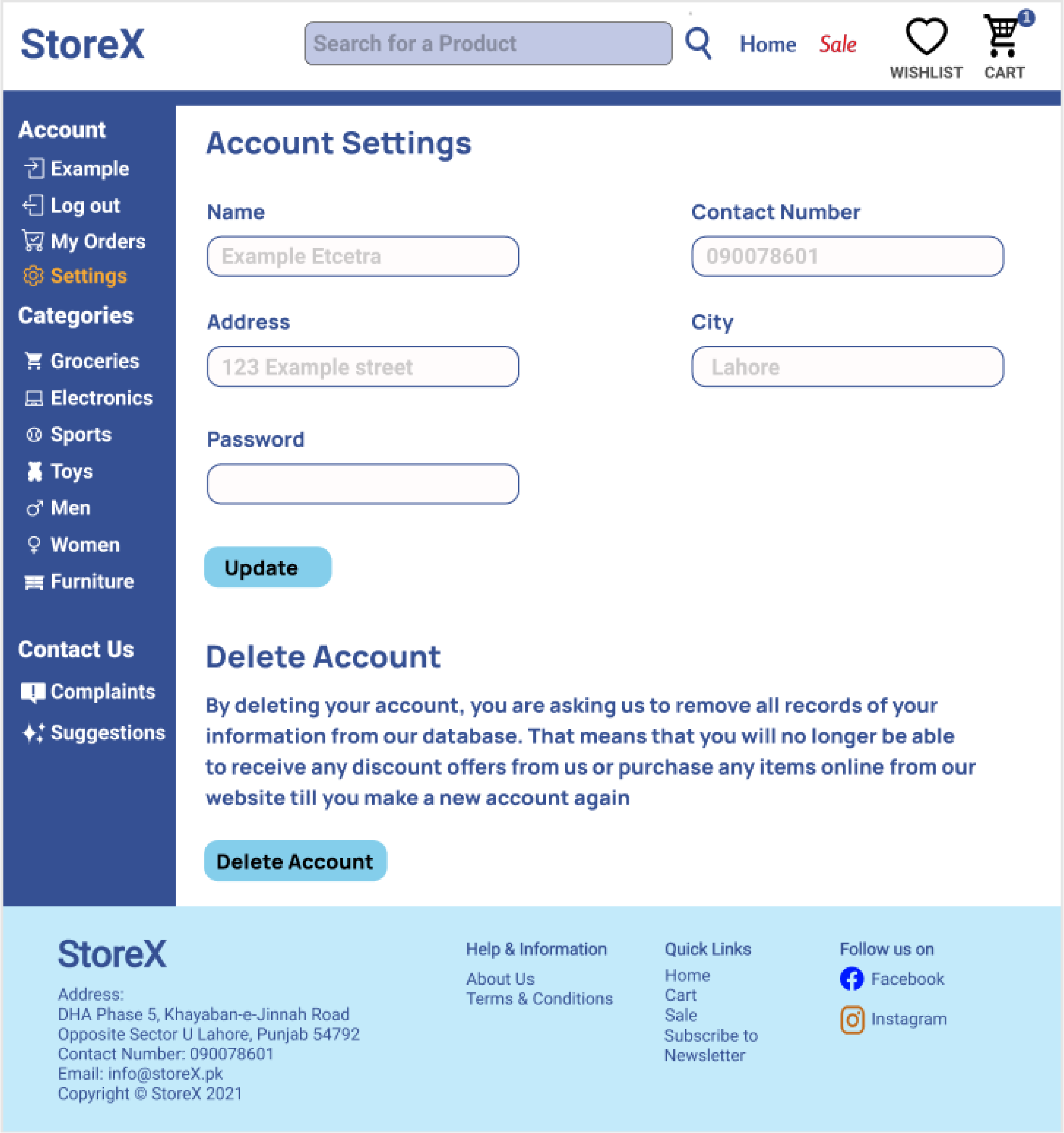
Quick Links
Home
Cart
Sale
Subscribe to Newsletter

Follow us on
Facebook
Instagram

Fig 5.3.6 (b) Individual Order Details

The user can view their pending and past orders by clicking on My Orders in the left navigation bar. This will take them to the main screen for My Orders, as shown in Fig 5.3.6 (a) above. The user can see which orders are pending (meaning they have not been dispatched yet), and which orders they have ordered in the past. The Order ID, Date, and total bill amount is visible for each order. The user can click on Reorder button to order any past order again (this will add the products with respective quantities of that particular order into their cart). The user can click on View Details to check more details about a specific order, as shown in Fig 5.3.6 (b). The user can check the names of the products, along with their respective quantities that they ordered. The subtotal, fixed shipping cost and total bill amount for that order are visible to the user too. The user can go back to the main screen of My Orders by clicking Go back to My Orders button.

5.3.7 Account Settings – User



Account Settings

Name: Example Etcetra

Contact Number: 090078601

Address: 123 Example street

City: Lahore

Password:

Delete Account

By deleting your account, you are asking us to remove all records of your information from our database. That means that you will no longer be able to receive any discount offers from us or purchase any items online from our website till you make a new account again

StoreX
Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

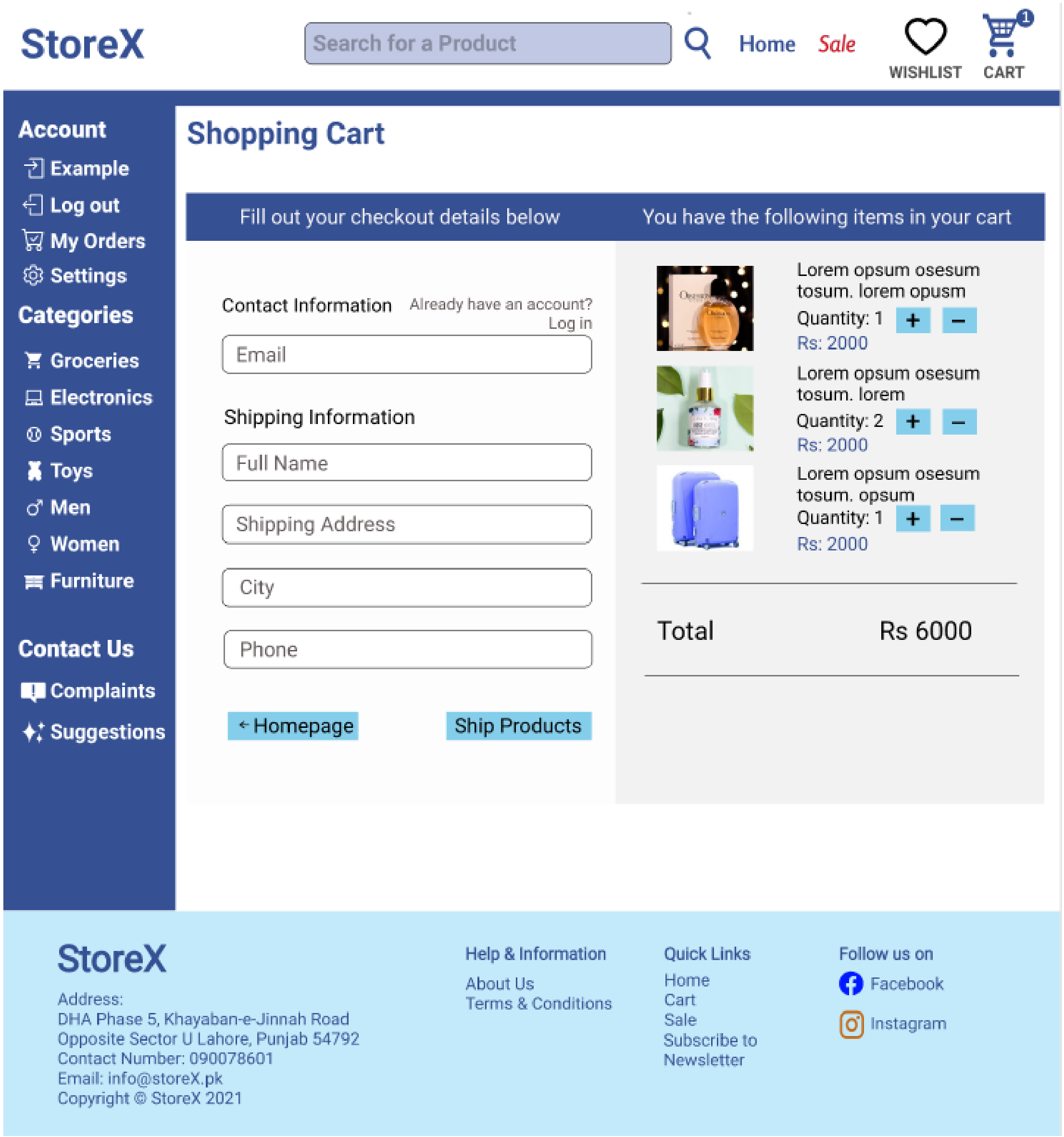
Help & Information
About Us
Terms & Conditions

Quick Links
Home
Cart
Sale
Subscribe to Newsletter

Follow us on
Facebook
Instagram

The user will only be able to visit this page if they are logged in. The user can either change their account credentials such as their name, contact number, address, city, or password, or can delete their account. This can be seen in the screenshot of the screen.

5.3.8 Shopping Cart – User



Shopping Cart

Fill out your checkout details below

Contact Information — Already have an account? Log in

Email

Shipping Information

Full Name

Shipping Address

City

Phone

You have the following items in your cart

Lorem opsum osesum tosum. lorem opusm
Quantity: 1
Rs: 2000

Lorem opsum osesum tosum. lorem
Quantity: 2
Rs: 2000

Lorem opsum osesum tosum. opsum
Quantity: 1
Rs: 2000

Total	Rs 6000

StoreX

Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

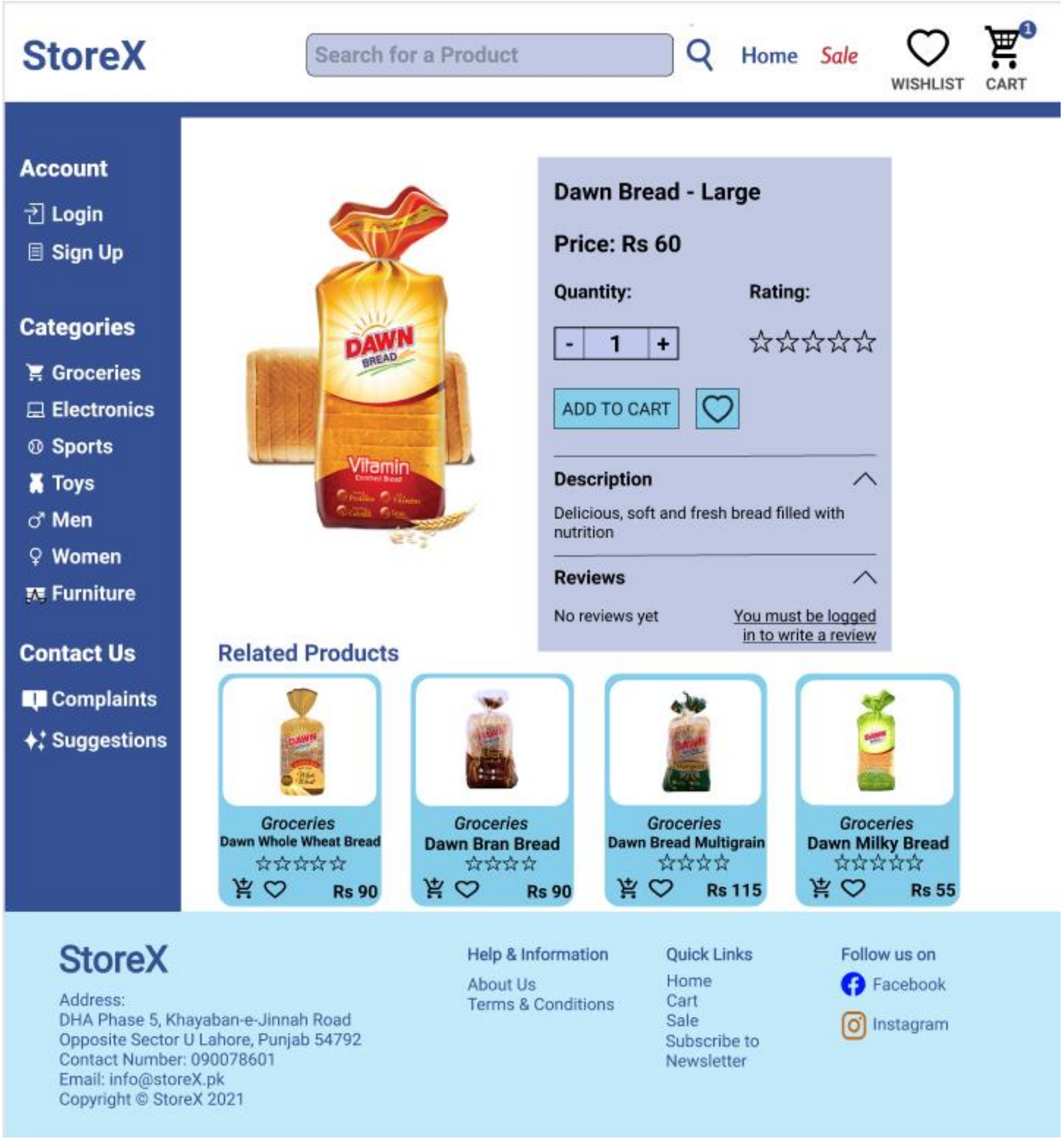
Help & Information
About Us
Terms & Conditions

Quick Links
Home
Cart
Sale
Subscribe to Newsletter

Follow us on
Facebook
Instagram

Shopping cart allows the users to add items which they want to buy. The shopping cart can contain multiple items at a time. It also allows the user to adjust the quantity of the items and displays the bill accordingly. Then right next to shopping cart section there is a form in which the users can enter their name, contact, and shipping details. The user then clicks on "ship products" button which completes the process.

5.3.9 **Product details – User**



Dawn Bread - Large

Price: Rs 60

Description

Delicious, soft and fresh bread filled with nutrition

Reviews

No reviews yet

You must be logged in to write a review

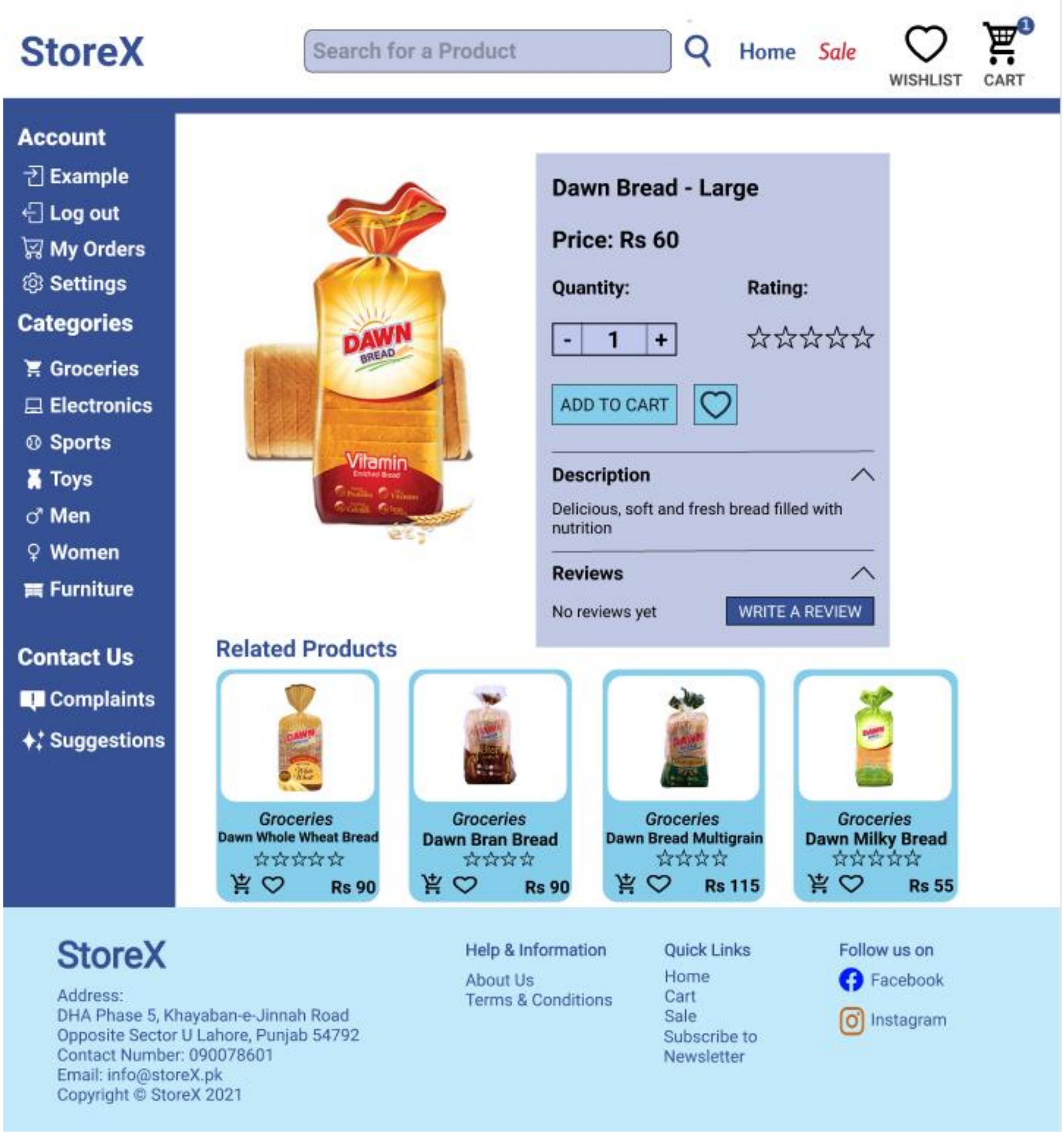
Related Products

Groceries	Groceries	Groceries	Groceries
Dawn Whole Wheat Bread	Dawn Bran Bread	Dawn Bread Multigrain	Dawn Milky Bread
☆☆☆☆☆	☆☆☆☆	☆☆☆☆	☆☆☆☆☆
Rs 90	Rs 90	Rs 115	Rs 55

StoreX

Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

Fig 5.3.9 (a) Product Details Page - User Not Logged In



Dawn Bread - Large

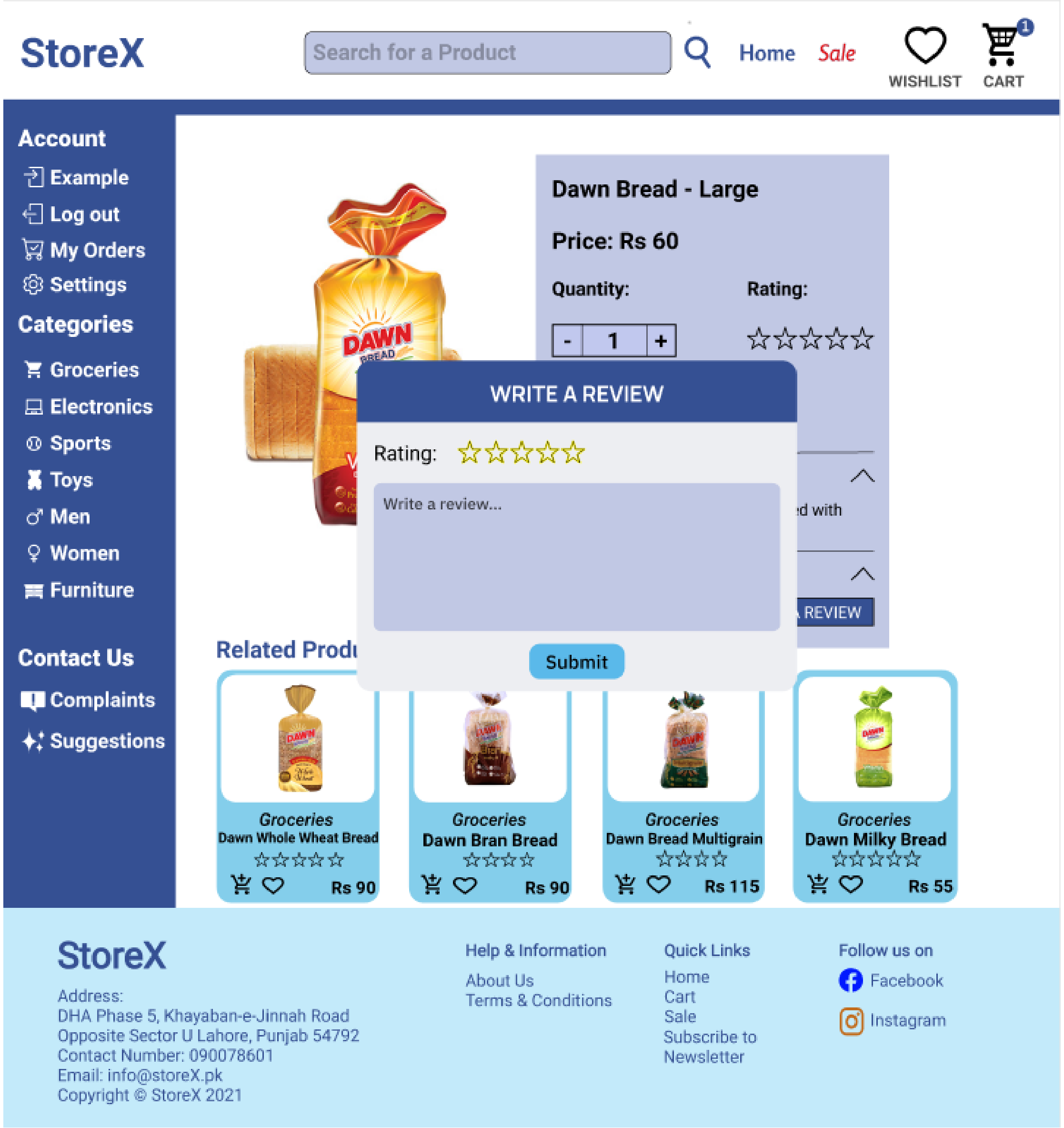
Price: Rs 60

Delicious, soft and fresh bread filled with nutrition

No reviews yet

Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

Fig 5.3.9 (b) Product Details Page - User Logged In



Dawn Bread - Large

Price: Rs 60

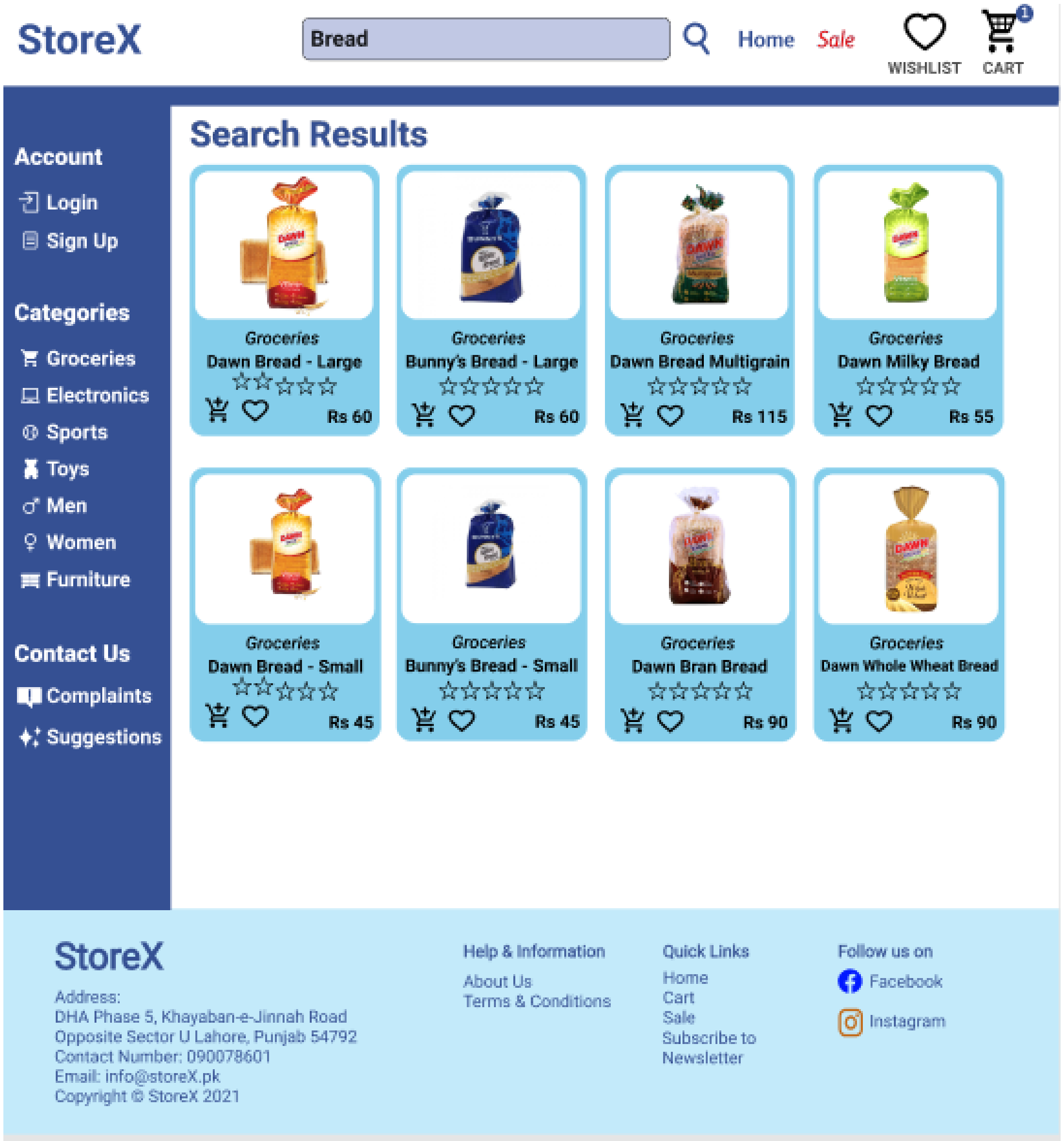
StoreX

Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

Fig 5.3.9 (c) Product Details Page - Write A Review Dialog Box

The user can view details of a particular product on its product details page (which can be accessed by clicking on the image of the product in any screen where the product is displayed). The name, image, price, and rating of the product are displayed. The user can select how much quantity of the product do they want to add to their cart. More information about the product is also given in form of its description. The user can go through reviews made by other users for that particular product. The user, if they are logged in, can also write their own review by clicking the Write A Review button (Fig 5.3.9 (b)), which will open up a dialog box where the user can add a rating and text review for that product (Fig 5.3.9 (c)). If the user is not logged in, the Write a Review button will be replaced by a link stating 'You must be logged in to write a review', and the link will lead to the login/signup page (Fig 5.3.9 (a)). There is also a section for related products, which can be helpful for the users if they want to check similar products from the same category. The user can compare the ratings and prices of these related products with the one they are currently viewing too.

5.3.10 **Search Results – User**



Bread

Search Results

Groceries
Dawn Bread - Large
☆☆☆☆☆
Rs 60

Groceries
Bunny's Bread - Large
☆☆☆☆☆
Rs 60

Groceries
Dawn Bread Multigrain
☆☆☆☆☆
Rs 115

Groceries
Dawn Milky Bread
☆☆☆☆☆
Rs 55

Groceries
Dawn Bread - Small
☆☆☆☆☆
Rs 45

Groceries
Bunny's Bread - Small
☆☆☆☆☆
Rs 45

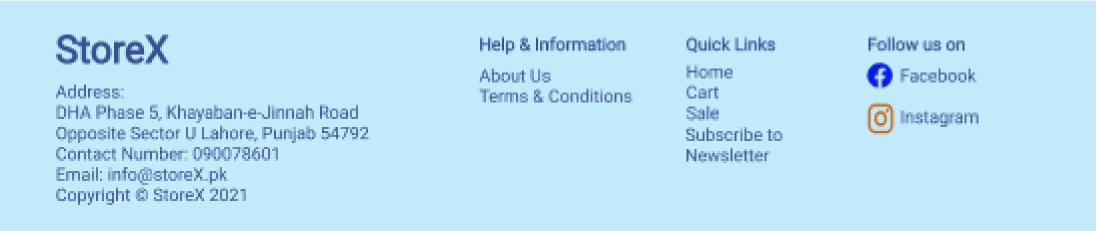
Groceries
Dawn Bran Bread
☆☆☆☆☆
Rs 90

Groceries
Dawn Whole Wheat Bread
☆☆☆☆☆
Rs 90

StoreX
Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

The user can search for a particular product by entering its name in the search bar in the header. This would open up the search results page, where products with matching names, or whose names contain the search term, will be displayed along with their categories, prices and ratings. The user can add products to their cart or wishlist by clicking on the respective icons. Also, the user can scroll down to see more results (if the displayed results do not all fit in the current view). If no matching results are found, the page will say that no products are found, and then the user can modify their search query if and as needed.

5.3.11 **Footer – User**



StoreX

Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

Help & Information

About Us
Terms & Conditions

Quick Links

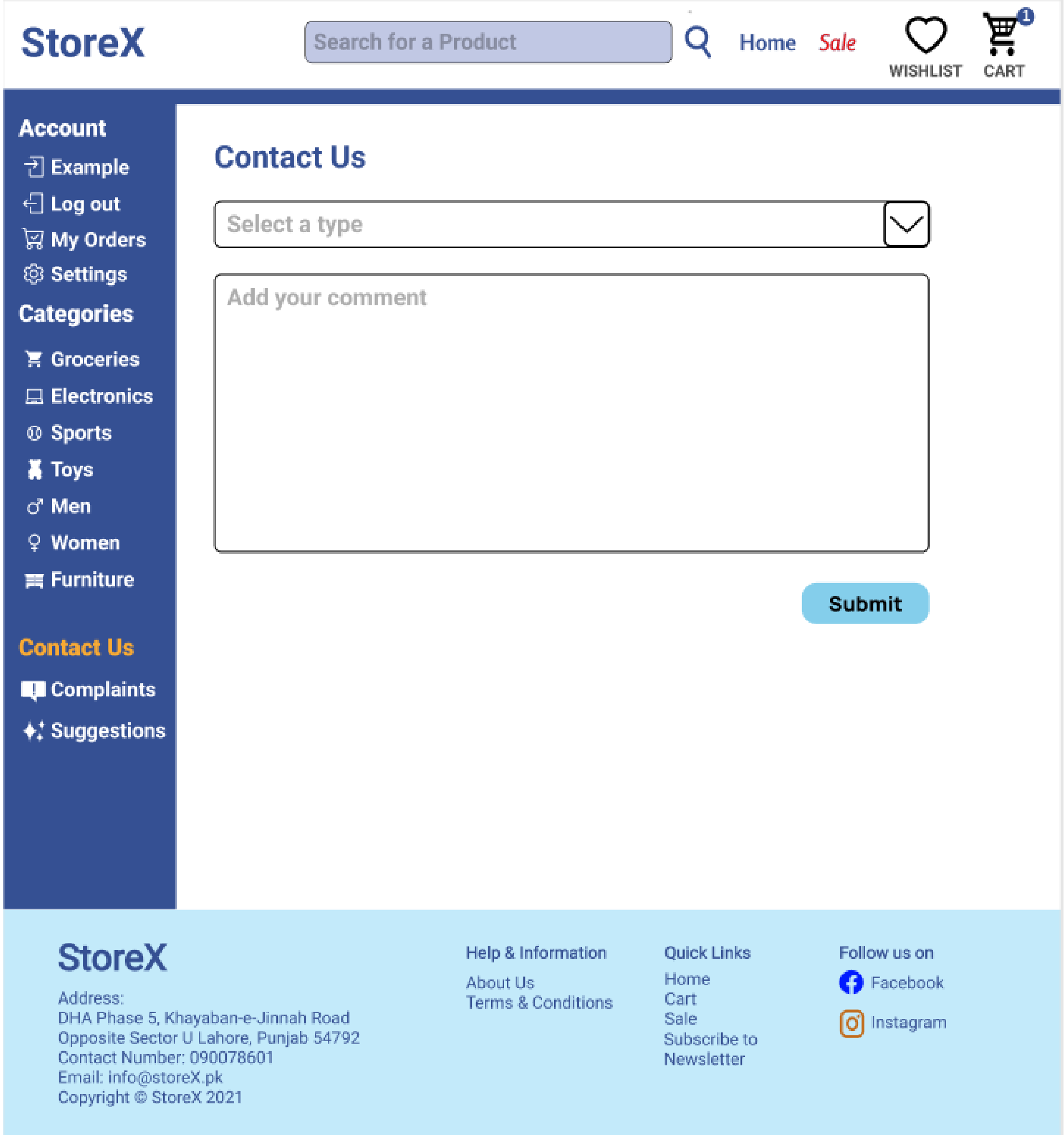
Home
Cart
Sale
Subscribe to Newsletter

Follow us on

Facebook
Instagram

The footer contains important information such as the contact details, working hours and company's office address. There is a "help&Info" section where the users can read about the company and its policies. The footer also has quick links for easy access to shopping cart, newsletter, and sale items. Lastly, there is a "follow us" section where the user can follow the company's facebook and Instagram page.

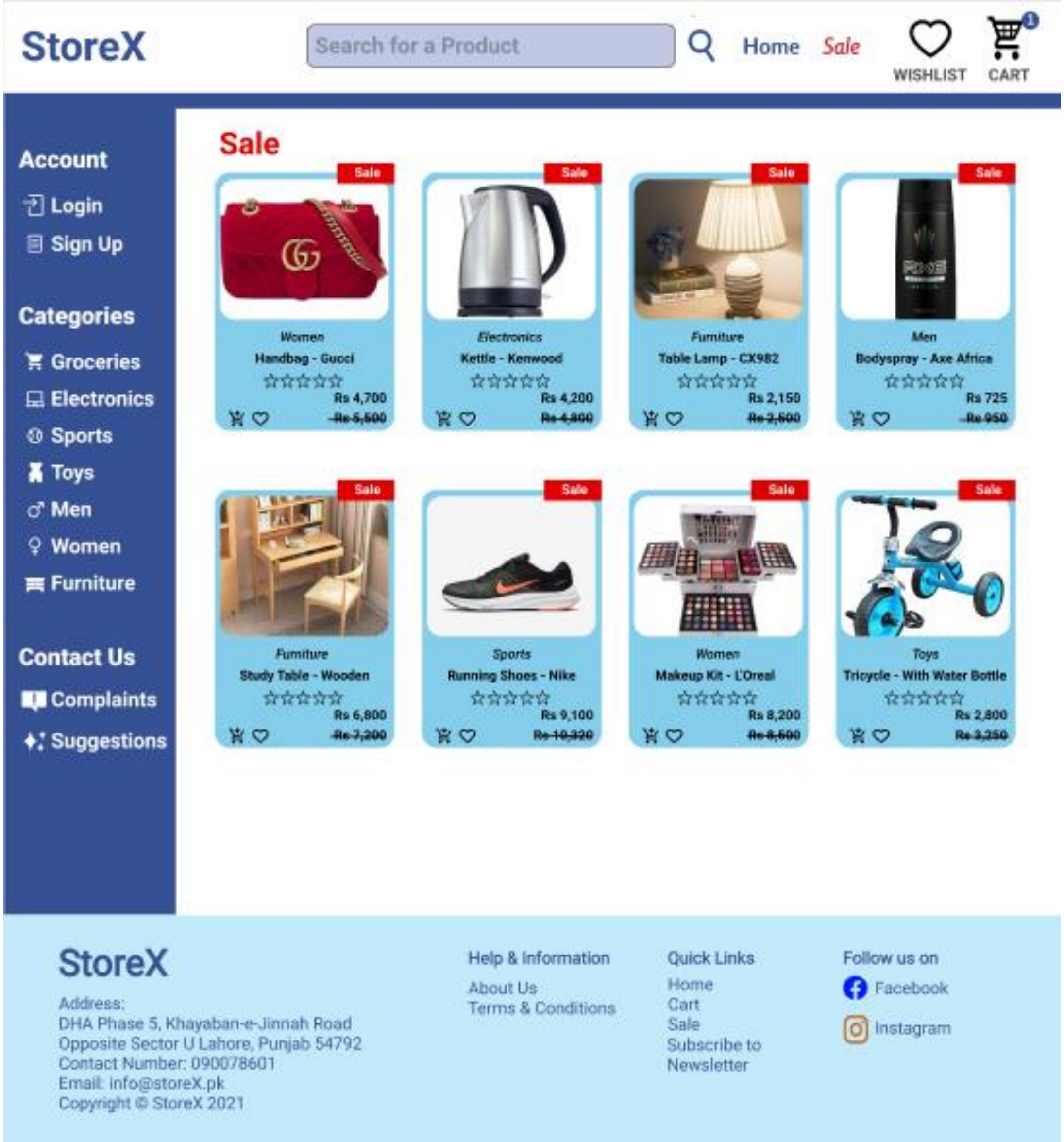
5.3.12 **Contact Us – User**



Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

The user will be able to submit their queries i.e complaints or suggestions depending on the type of query they choose to submit. They can choose the type of query using the drop down box shown on the screen. They can then add their comments in the comments box and click on submit. If they are logged in to the system, their query will be submitted. If not, they will have to first login (Or create an account if they still haven't created one) and then submit their query.

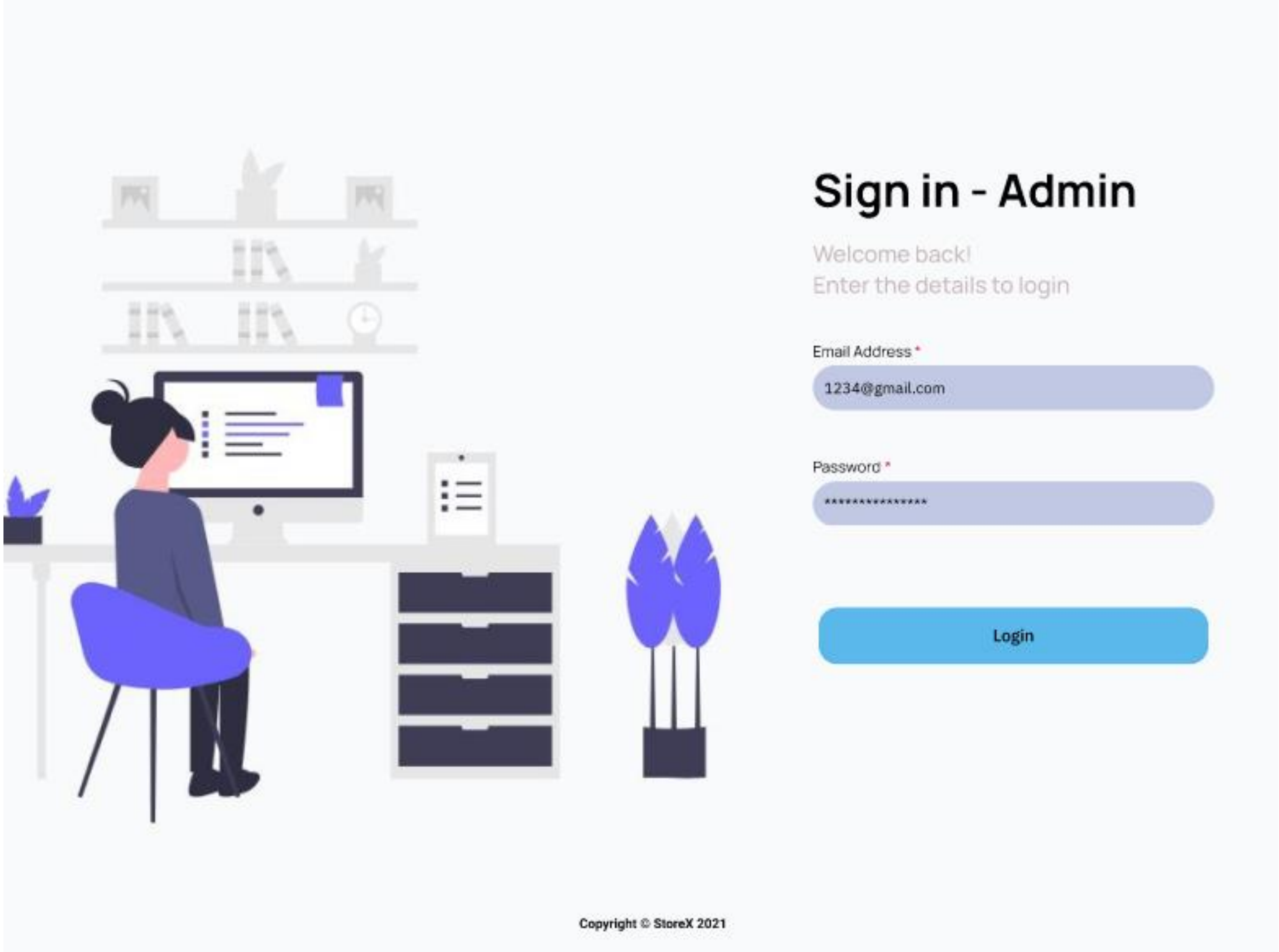
5.3.13 Sale - User



Address:
DHA Phase 5, Khayaban-e-Jinnah Road
Opposite Sector U Lahore, Punjab 54792
Contact Number: 090078601
Email: info@storeX.pk
Copyright © StoreX 2021

The user will be directed to this screen upon clicking the "sale" button in the header. The sale screen will be displaying all the products that are currently on sale. The user can scroll down to view more products, if more products than the one shown on screen are on sale. Clicking on any product will direct the user to the Products page.

5.3.14 **Sign in – Admin**



Sign in - Admin

Welcome back!
Enter the details to login

Email Address *

1234@gmail.com

Password *

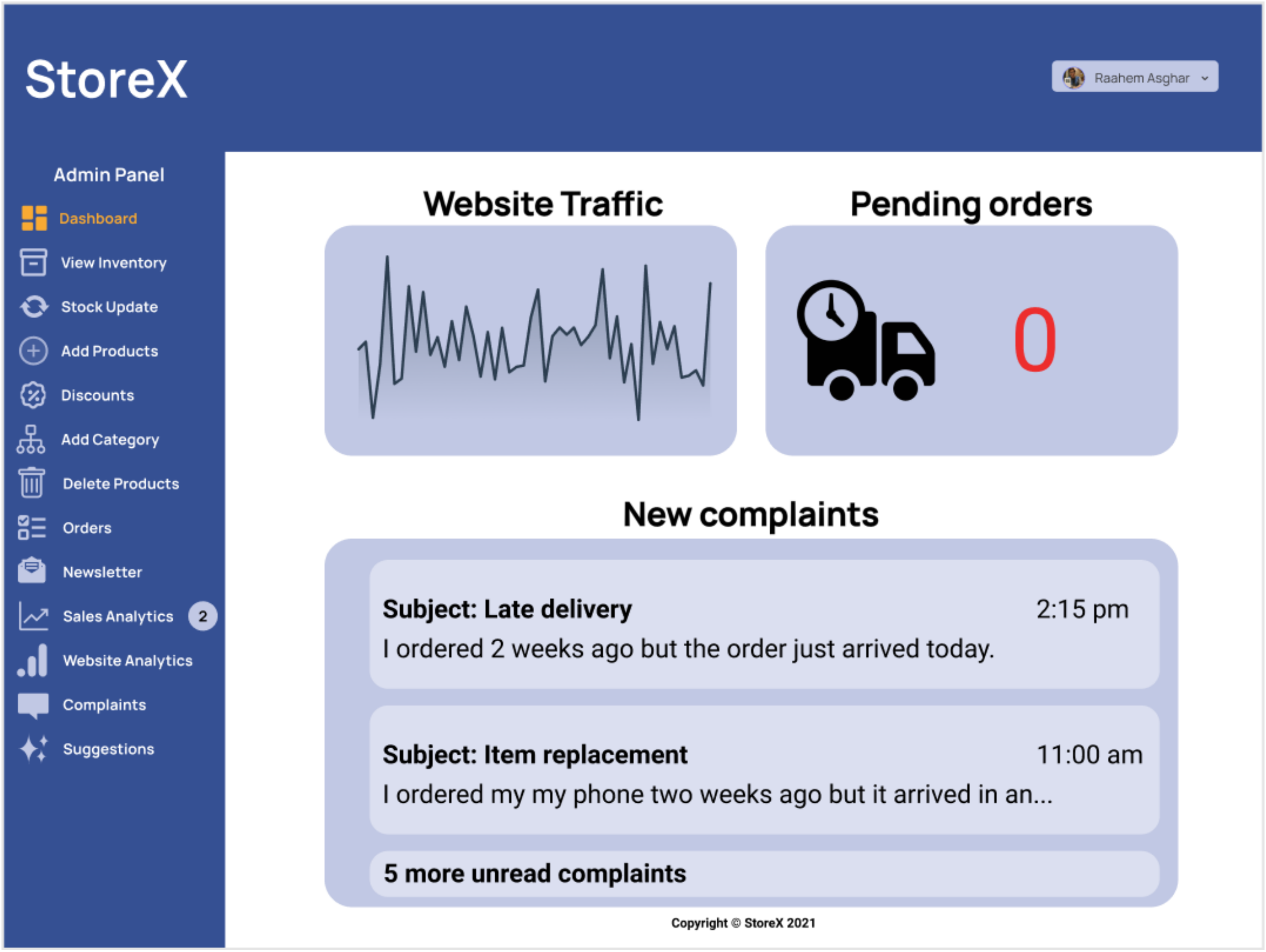
****

Copyright © StoreX 2021

Fig 5.3.14

This screen allows the admin to sign in to the store to perform various tasks. The admin keys in his/her email ID along with the password into the correct fields. Once the admin presses sign in, the information entered in the fields is compared with the one stored in the database. If the email ID and password entered are correct i.e. match with the ones in the database, the admin is redirected to the dashboard. If the email ID or password is incorrect, access is denied.

5.3.15 **Dashboard – Admin**



Website Traffic

Pending orders

0

New complaints

Subject: Late delivery 2:15 pm
I ordered 2 weeks ago but the order just arrived today.

Subject: Item replacement 11:00 am
I ordered my my phone two weeks ago but it arrived in an...

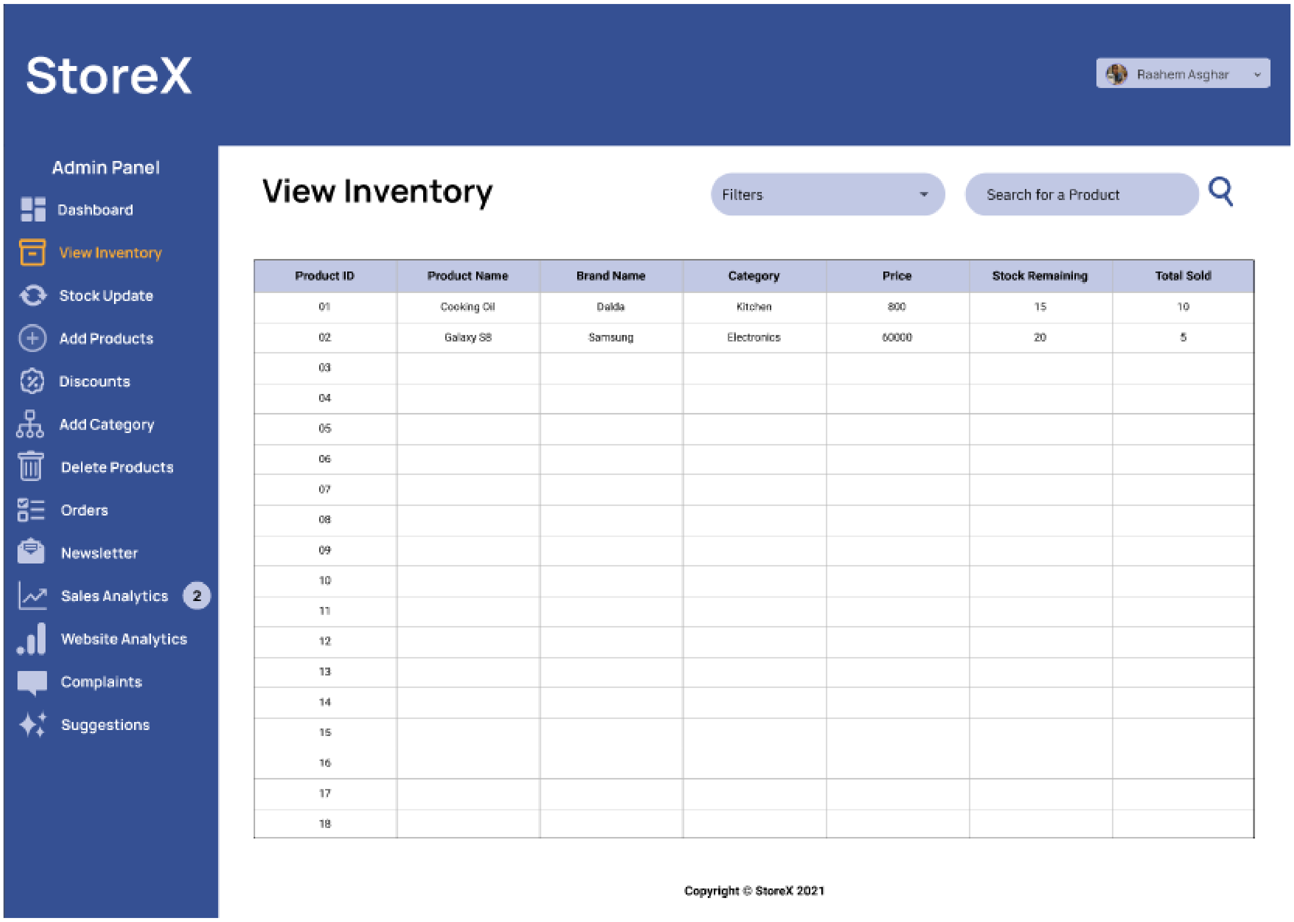
5 more unread complaints

Copyright © StoreX 2021

Fig 5.3.15

The dashboard is "homepage" of the admin side. This is opened by default or can be returned to from the toolbar. This screen gives admin the summary of the web application. The web application traffic gives an overview of the number of visitors over last few days. The pending order counter display the number of orders that have yet to be delivered and new complaints displays the most recent complaints to the admin. He/she can then go to the relevant part through the toolbar.

5.3.16 **View Inventory – Admin**



View Inventory

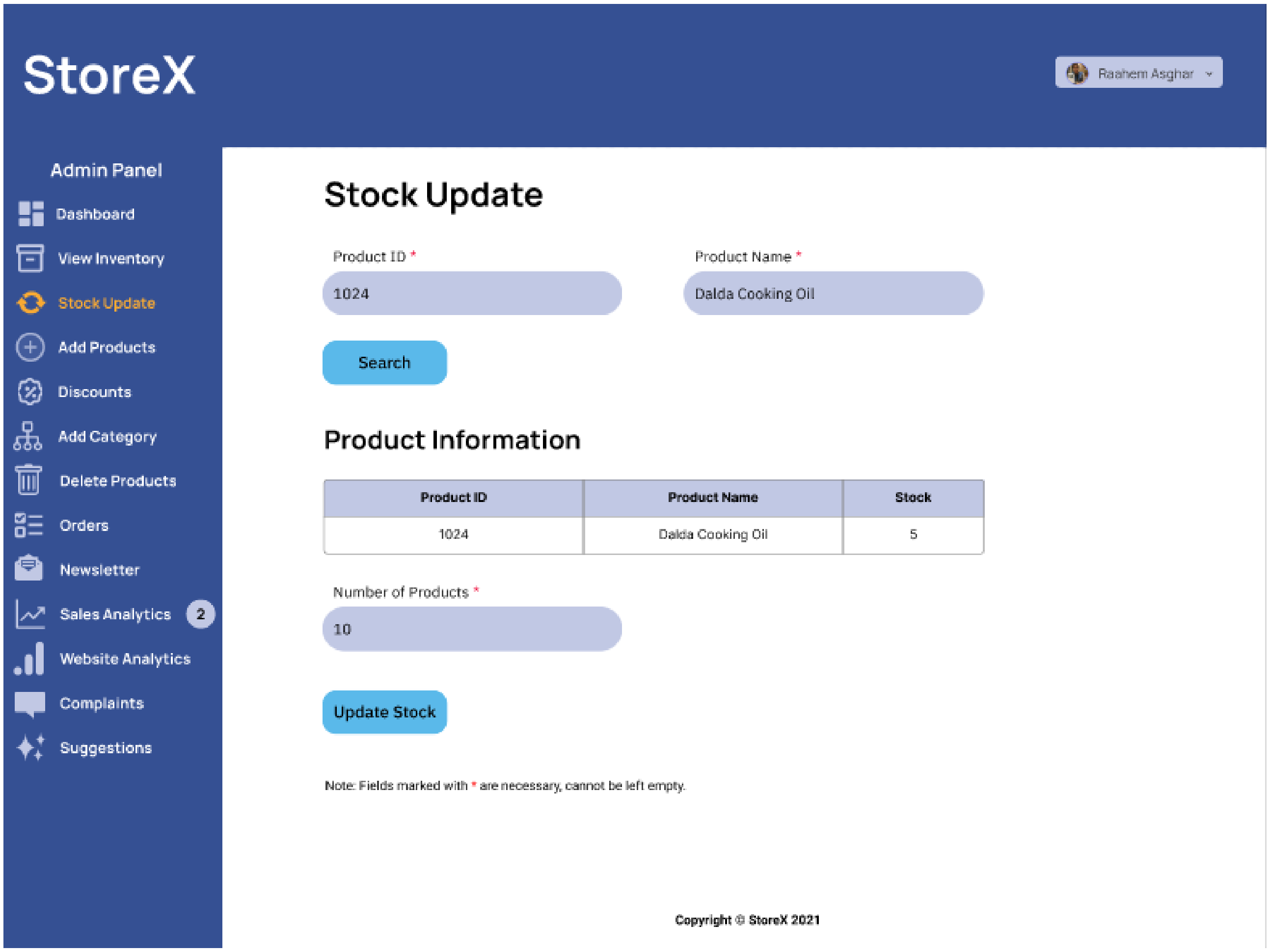
Product ID	Product Name	Brand Name	Category	Price	Stock Remaining	Total Sold
01	Cooking Oil	Dalda	Kitchen	800	15	10
02	Galaxy S8	Samsung	Electronics	60000	20	5
03						
04						
05						
06						
07						
08						
09						
10						
11						
12						
13						
14						
15						
16						
17						
18						

Copyright © StoreX 2021

Fig 5.3.16

The sidebar on the screen shows the various tasks the admin can perform. Currently, the admin is on the "View Inventory" screen (the name has been highlighted in the side bar). A table showing the inventory of the store is displayed showing the product ID, product name, Price etc. The admin can apply filters using the dropdown menu (next to the search bar) to narrow down products. Other than that, he/she can key in an item in the search bar and once the search icon is pressed, the product with its details is shown in the table.

5.3.17 **Stock** Update – Admin



Stock Update

Product ID * 1024

Product Name * Dalda Cooking Oil

Product Information

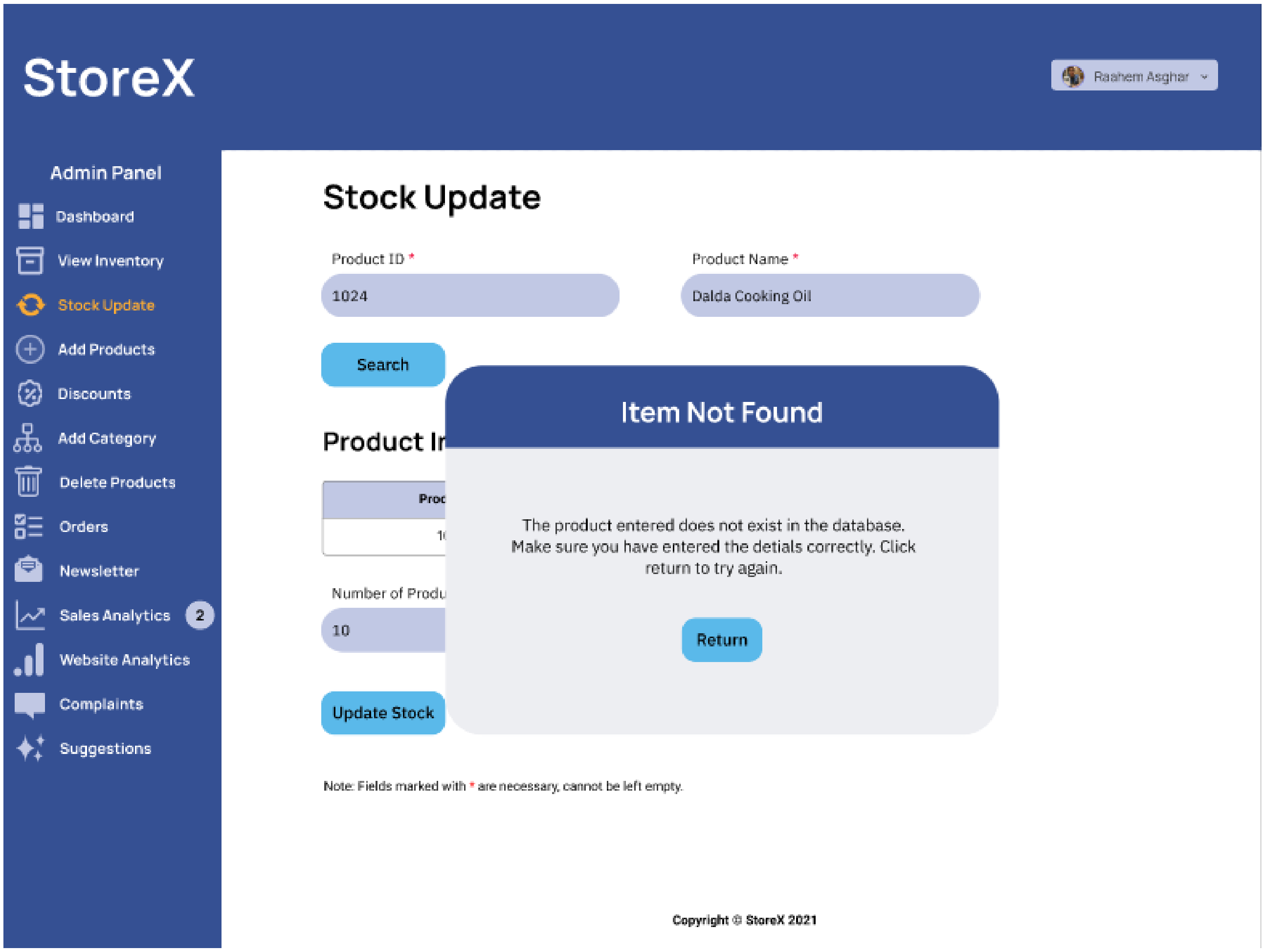
Product ID	Product Name	Stock
1024	Dalda Cooking Oil	5

Number of Products * 10

Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.17 (a)



Stock Update

Product ID * 1024

Product Name * Dalda Cooking Oil

Product In

Proc
1(

Number of Produ 10

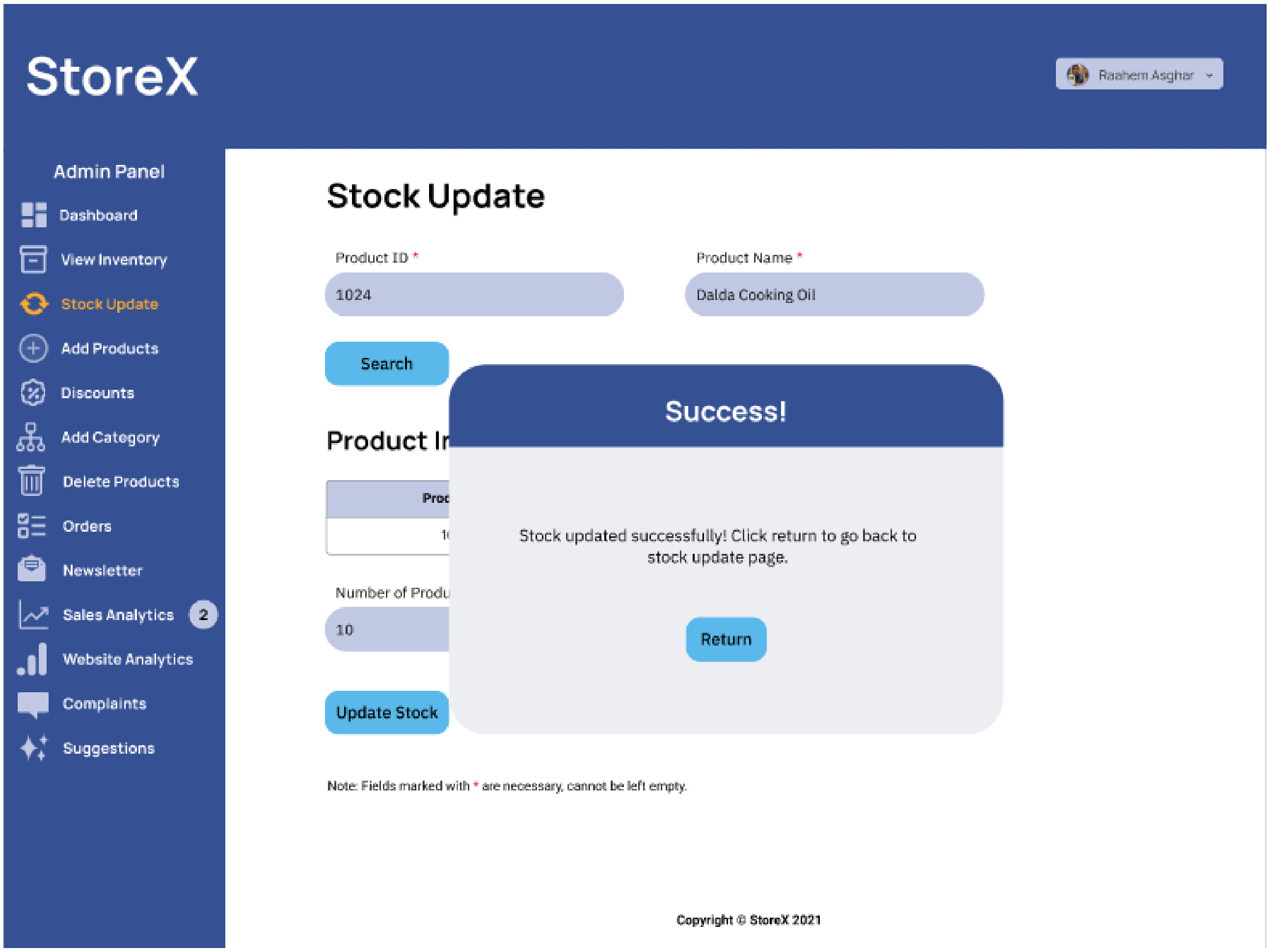
Item Not Found

The product entered does not exist in the database. Make sure you have entered the detials correctly. Click return to try again.

Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.17 (b)



Stock Update

Product ID * 1024

Product Name * Dalda Cooking Oil

Number of Produ 10

Success!

Stock updated successfully! Click return to go back to stock update page.

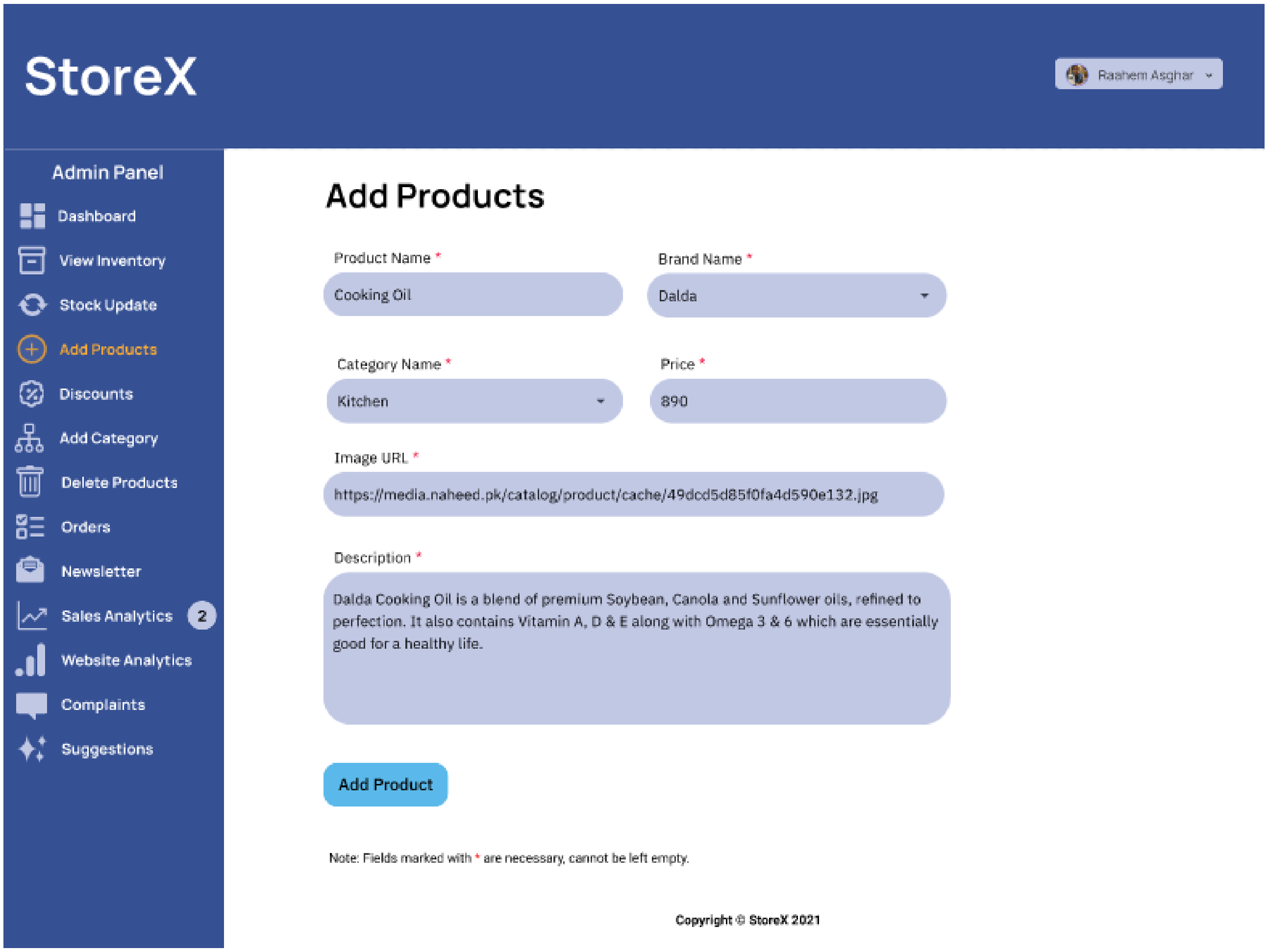
Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.17 (c)

Currently, the admin is on the “Stock Update” screen (the name has been highlighted in the side bar). Here the admin enters the name and ID of the product whose stock the admin wishes to update. Once the admin has entered the respective details, he/she presses search to see if the product actually exists (in case he/she enters incorrect information or doesn’t remember if the product is in the store or not). If the product exists, then it is displayed with the current stock in the table. If it does not exist, message is displayed showing “No match found” (Fig 5.3.17 (b)). The admin then enters the number of items he wishes to add in the “Number of Products” field and then presses the “Update Stock” button. Now, the stock of that particular item has been updated with the new value and a success message is displayed (Fig 5.3.17 (c)). The admin then returns to the previous screen by pressing return in the dialog box.

5.3.18 **Add Products –** Admin



Add Products

Product Name *: Cooking Oil

Brand Name *: Dalda

Category Name *: Kitchen

Price *: 890

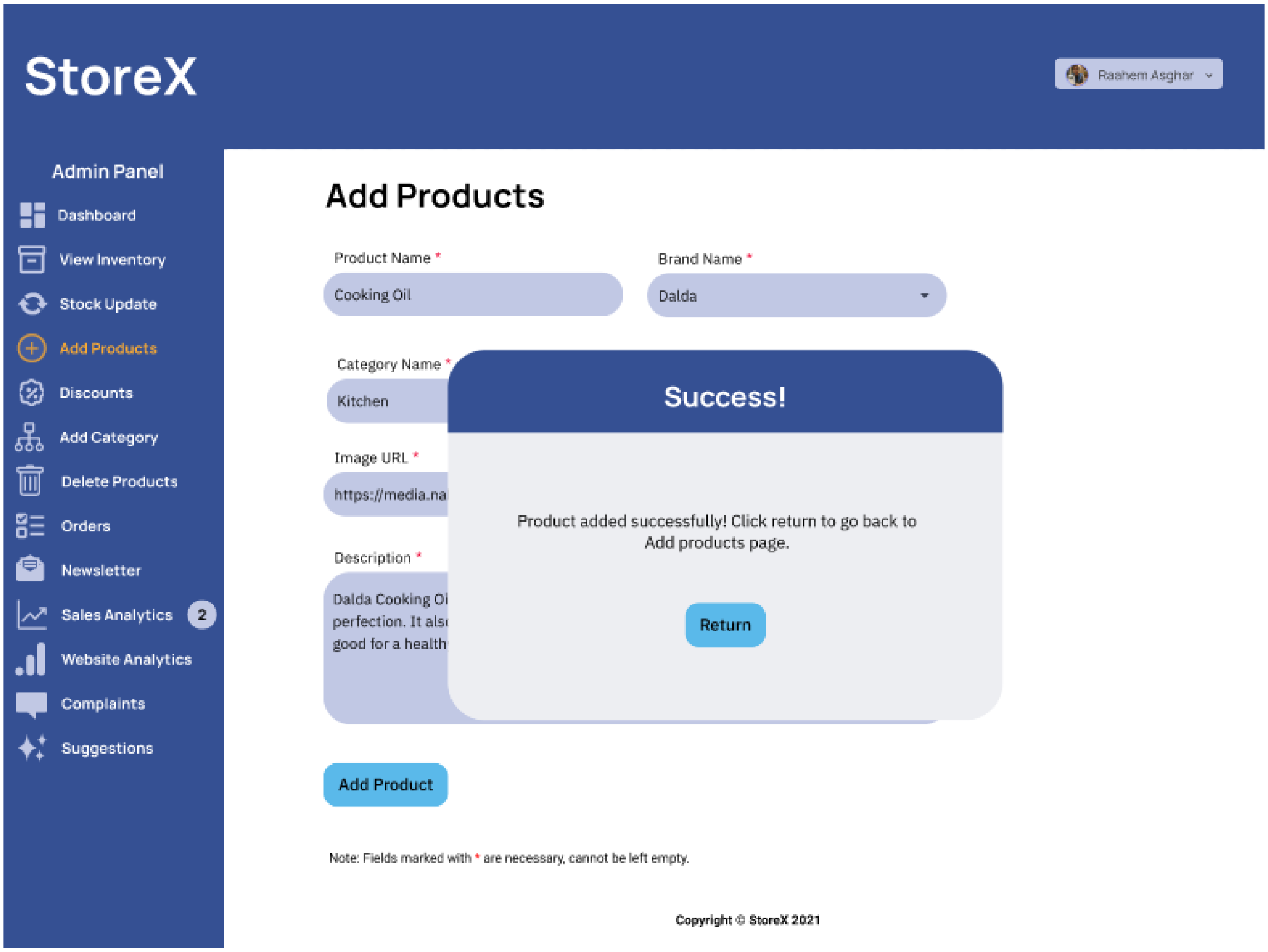
Image URL *: https://media.naheed.pk/catalog/product/cache/49dcd5d85f0fa4d590e132.jpg

Description *: Dalda Cooking Oil is a blend of premium Soybean, Canola and Sunflower oils, refined to perfection. It also contains Vitamin A, D & E along with Omega 3 & 6 which are essentially good for a healthy life.

Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.18 (a)



Add Products

Product Name * Cooking Oil

Brand Name * Dalda

Category Name * Kitchen

Image URL * https://media.na

Description * Dalda Cooking Oi perfection. It als good for a health

Success!

Product added successfully! Click return to go back to Add products page.

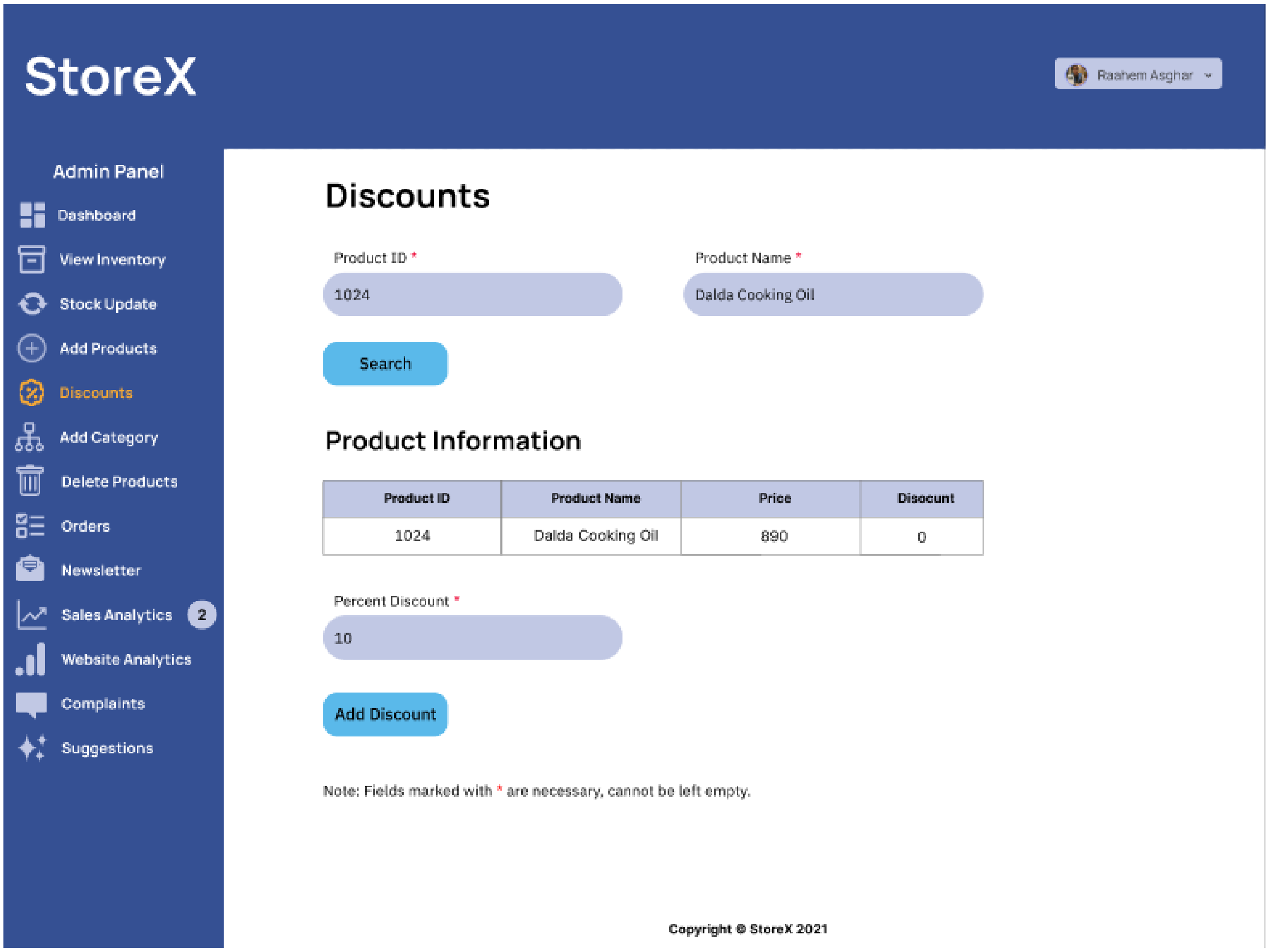
Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.18 (b)

Currently, the admin is on the "Add Products" screen (the name has been highlighted in the side bar). This screen consists of a form that requires the admin to enter the product details of a new product which the admin wishes to add in the store. He/she keys in the product name in the product name field, selects the brand and category ID from the dropdown menu, enters the price, image URL, and description in the respective fields. Once all the information has been keyed in, the admin presses the "Add Product" button to submit the form. Once the button has been pressed, the database is updated with the information of the new product and a random product ID has been assigned to the product. A message is displayed showing that the product has successfully been added (Fig 5.3.18 (b)). The admin then returns to the previous screen by pressing return in the dialog box.

5.3.19 Discounts – Admin



Discounts

Product ID * 1024

Product Name * Dalda Cooking Oil

Product Information

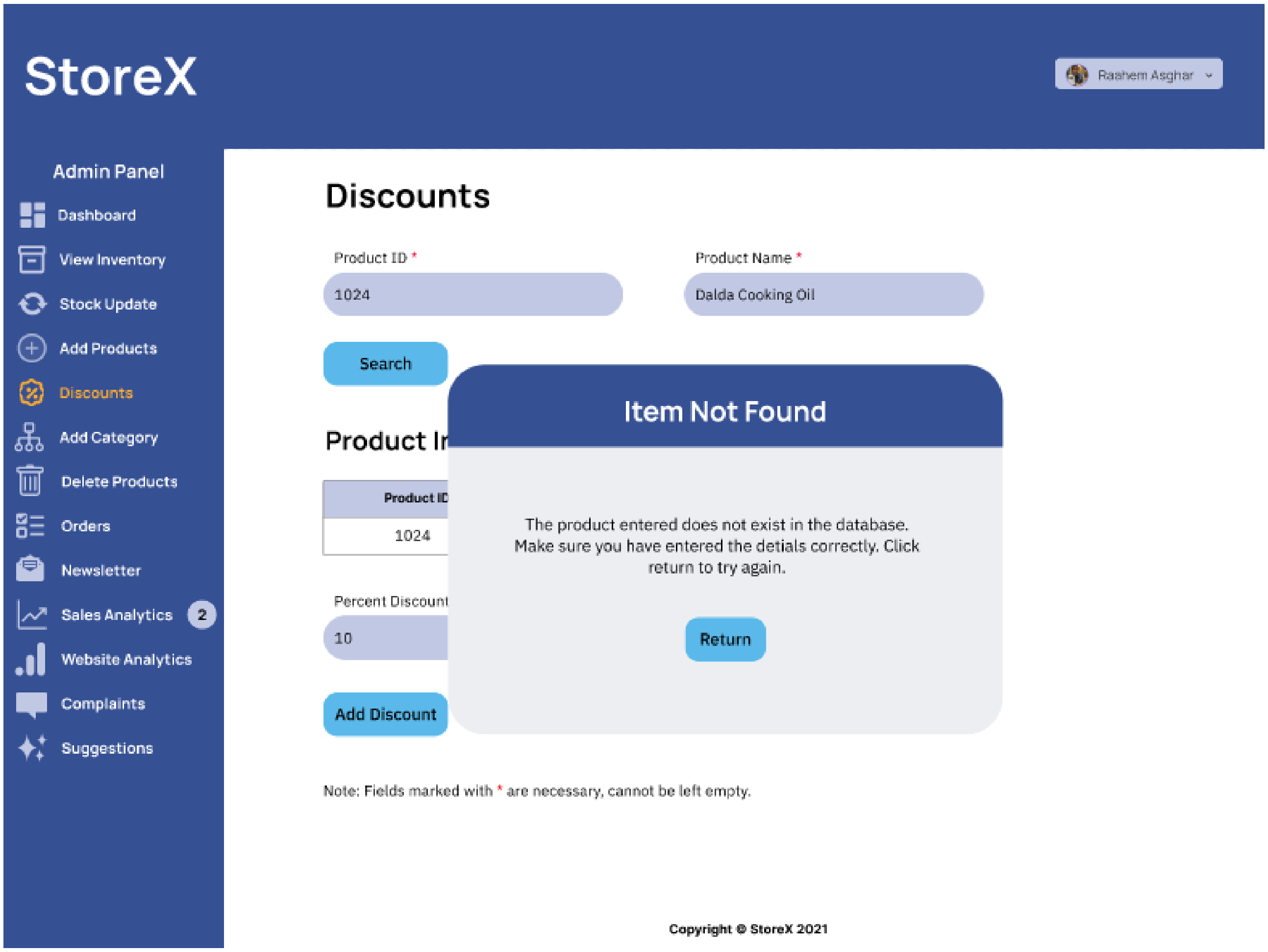
Product ID	Product Name	Price	Discount
1024	Dalda Cooking Oil	890	0

Percent Discount * 10

Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.19 (a)



Discounts

Product ID * 1024

Product Name * Dalda Cooking Oil

Percent Discount 10

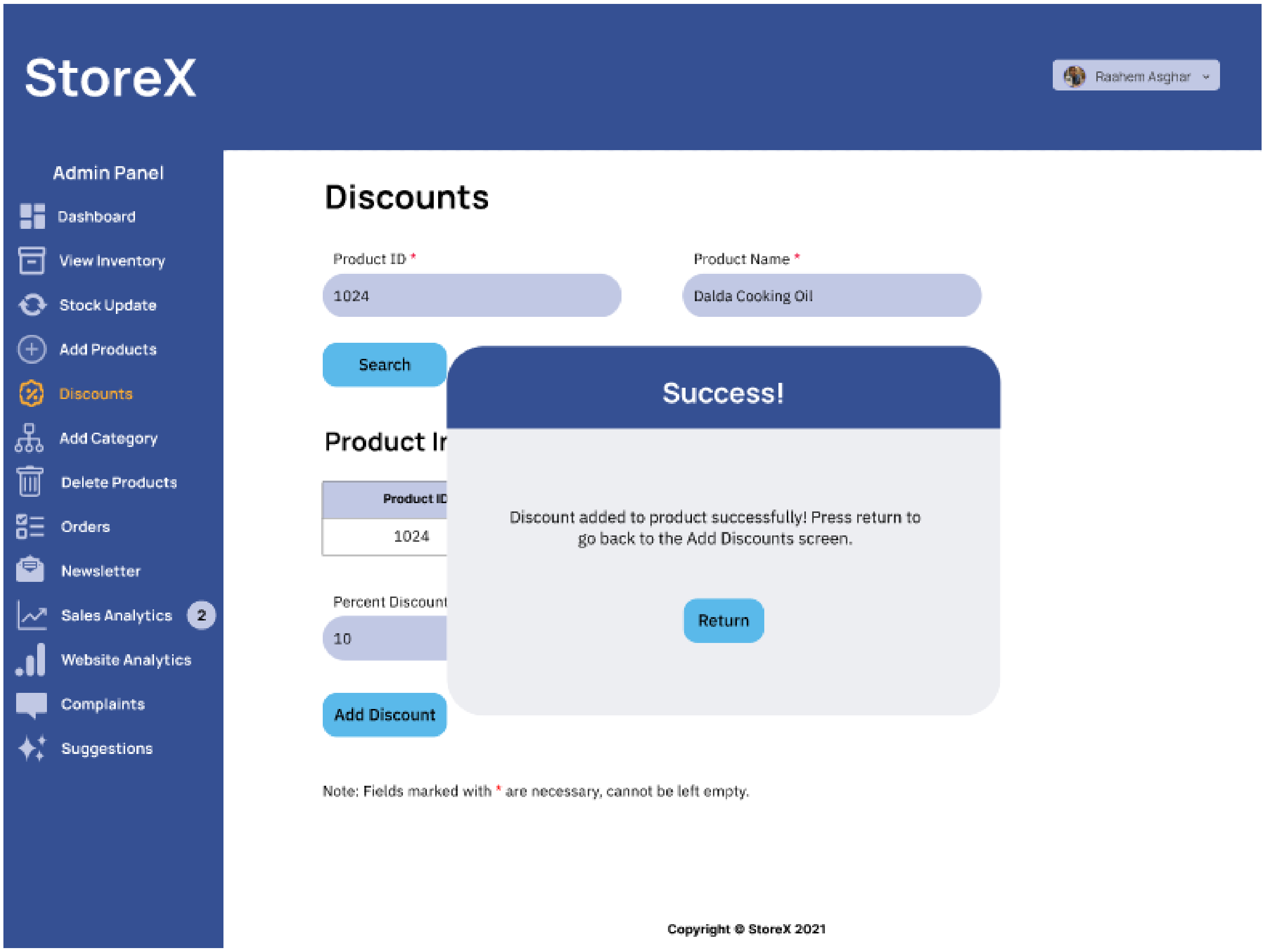
Item Not Found

The product entered does not exist in the database. Make sure you have entered the detials correctly. Click return to try again.

Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.19 (b)



Discounts

Product ID * 1024

Product Name * Dalda Cooking Oil

Product In

Product ID
1024

Percent Discount 10

Success!

Discount added to product successfully! Press return to go back to the Add Discounts screen.

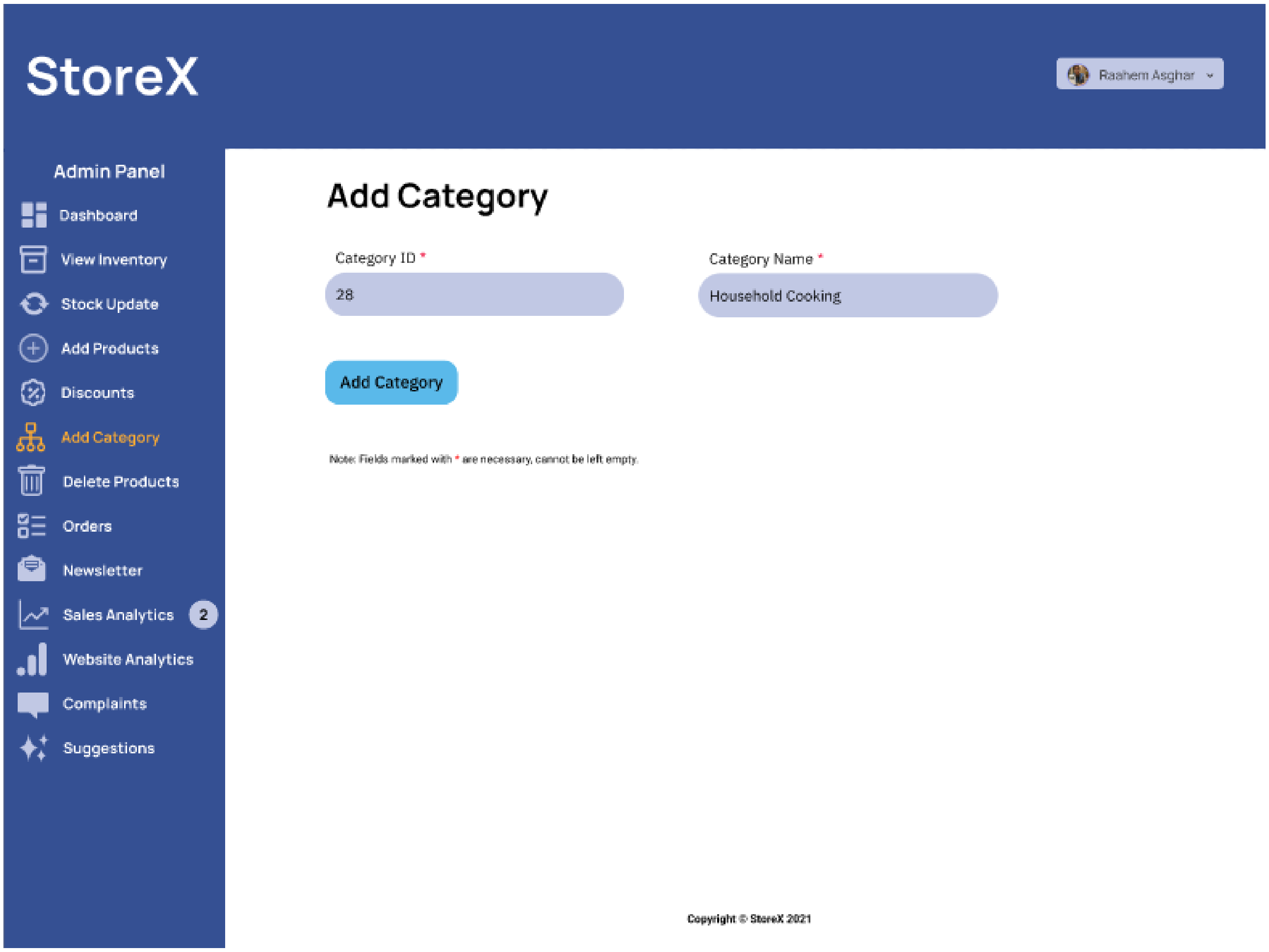
Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.19 (c)

Currently, the admin is on the “Discounts” screen (the name has been highlighted in the side bar). Here the admin enters the name and ID of the product onto which the admin wishes to add discount. Once the admin has entered the respective details, he/she presses search to see if the product actually exists (in case he/she enters incorrect information or doesn’t remember if the product is in the store or not). If the product exists, then it is displayed with its information in the table. If it does not exist, message is displayed showing “No match found” (Fig 5.3.19 (b)). Next, the admin enters the percent discount in “Percent Discount” field and presses the “Add Discount” button. A message showing that the discount has been added successfully is displayed and the product information has been updated (Fig 5.3.19 (c)). The admin then returns to the previous screen by pressing return in the dialog box.

5.3.20 **Add Category – Admin**



Category ID *

28

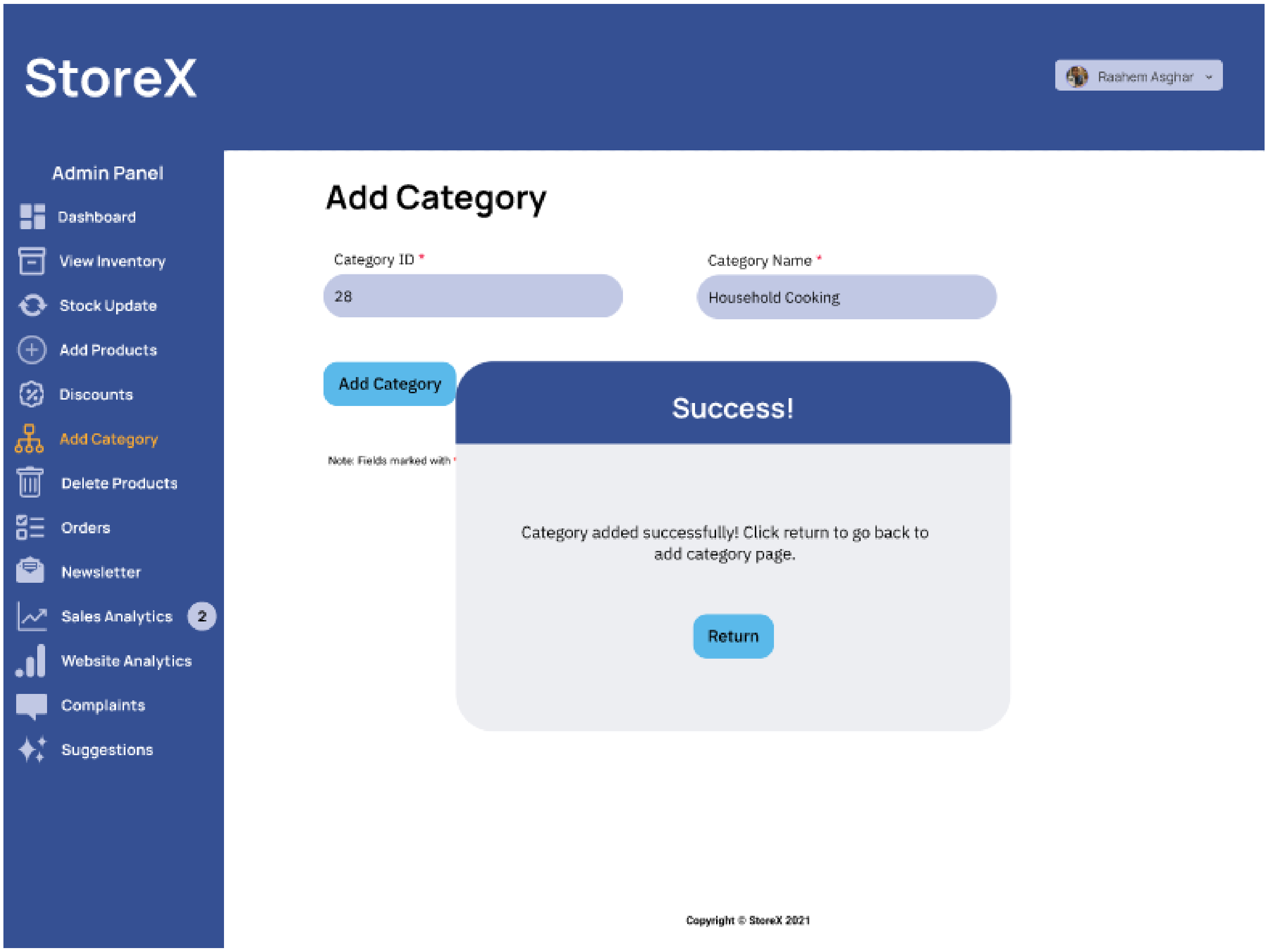
Category Name *

Household Cooking

Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.20 (a)



Category ID * 28

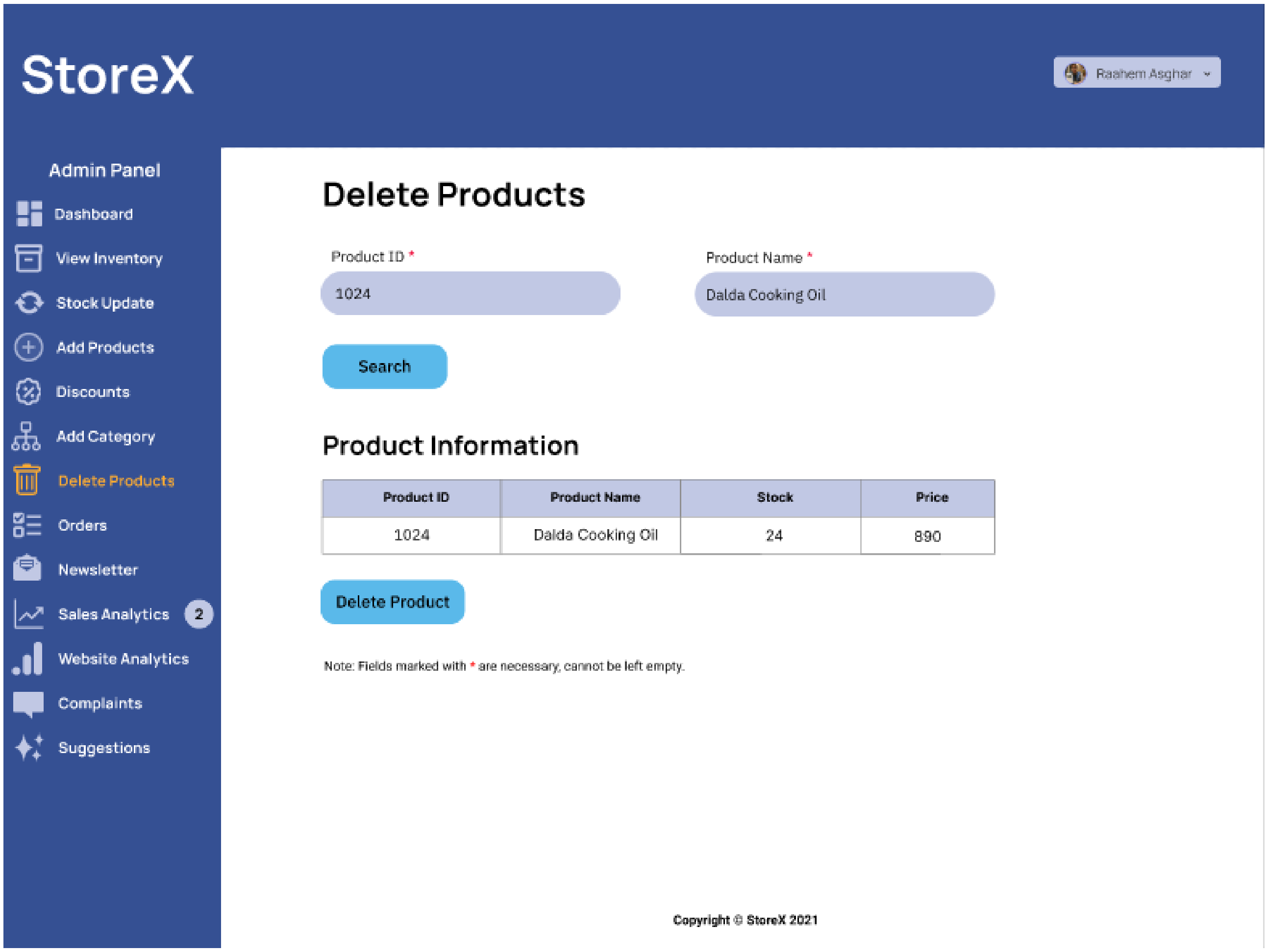
Category Name * Household Cooking

Category added successfully! Click return to go back to add category page.

Fig 5.3.20 (b)

Currently, the admin is on the "Add Category" screen (the name has been highlighted in the side bar). The screen consists of a form where the admin enters details regarding the new category. The admin enters category ID and the name of the new category to be added in the respective fields. He/she then presses "Add Category" button and the database is updated with the new information and a success message is displayed on the screen (Fig 5.3.20 (b)). The admin then returns to the previous screen by pressing return in the dialog box.

5.3.21 **Delete Products – Admin**



Delete Products

Product ID * 1024

Product Name * Dalda Cooking Oil

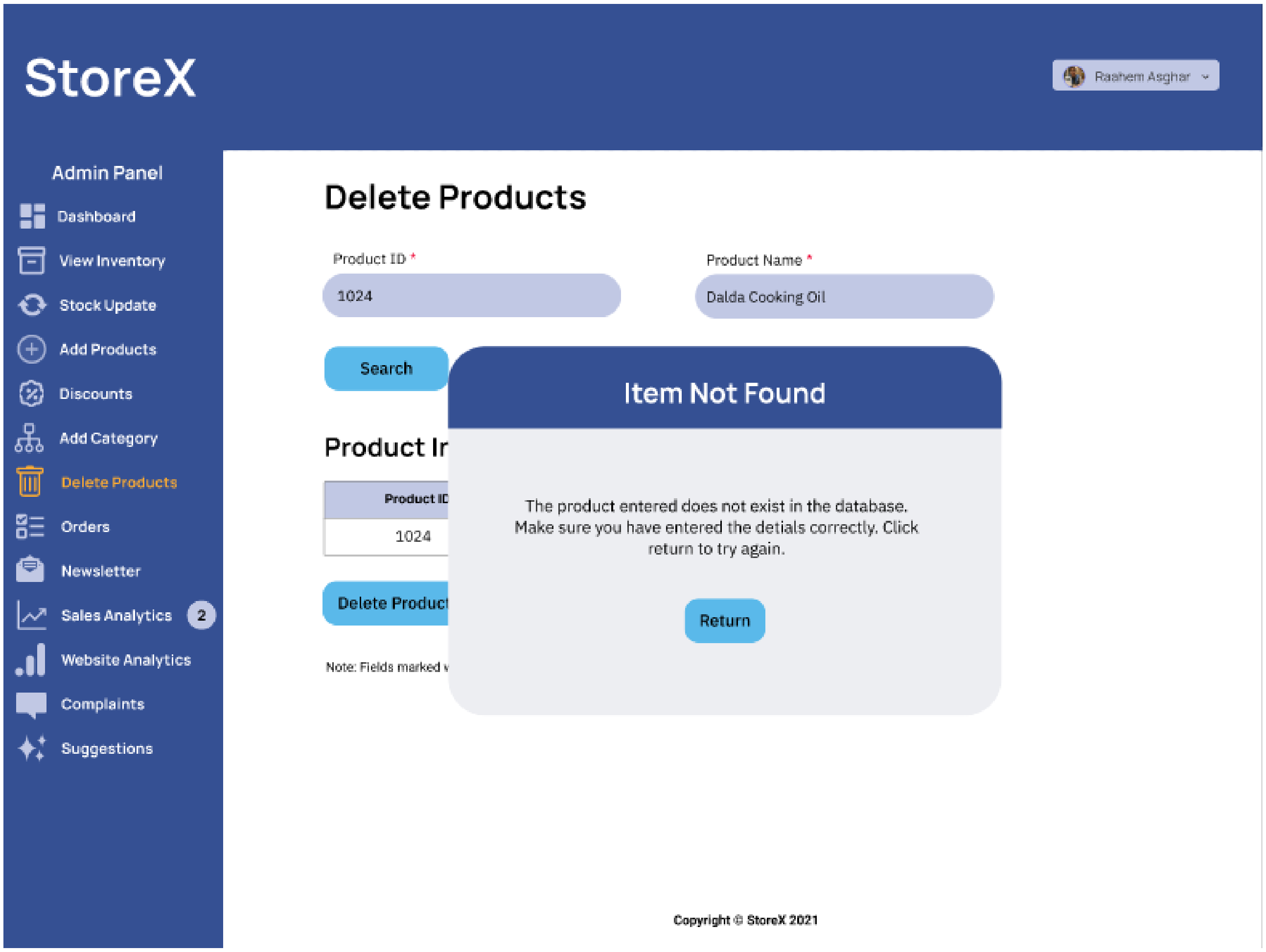
Product Information

Product ID	Product Name	Stock	Price
1024	Dalda Cooking Oil	24	890

Note: Fields marked with * are necessary, cannot be left empty.

Copyright © StoreX 2021

Fig 5.3.21 (a)



Delete Products

Product ID *

1024

Product Name *

Dalda Cooking Oil

Product In

Product ID
1024

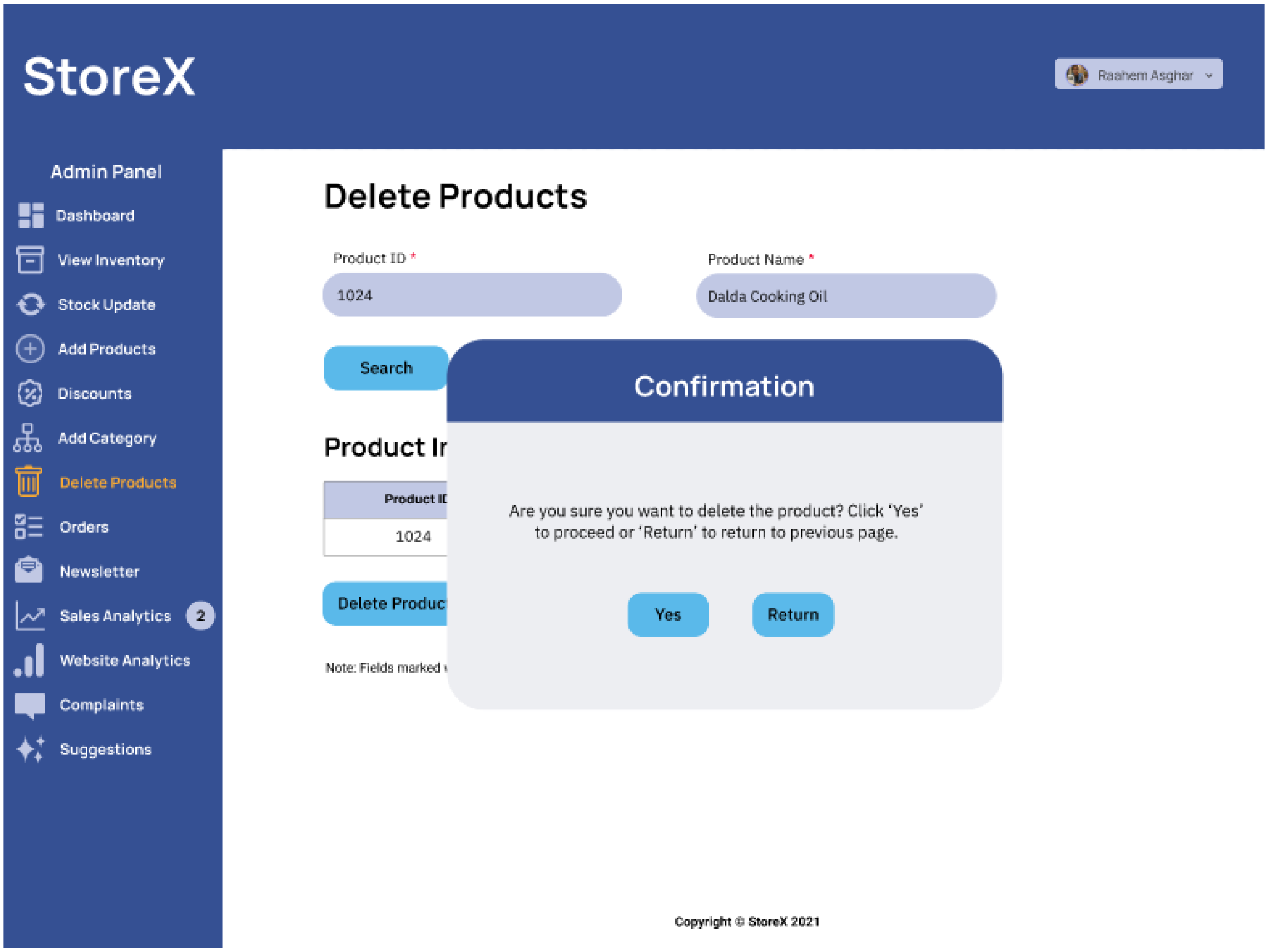
Note: Fields marked v

Item Not Found

The product entered does not exist in the database. Make sure you have entered the detials correctly. Click return to try again.

Copyright © StoreX 2021

Fig 5.3.21 (b)



Product ID * 1024

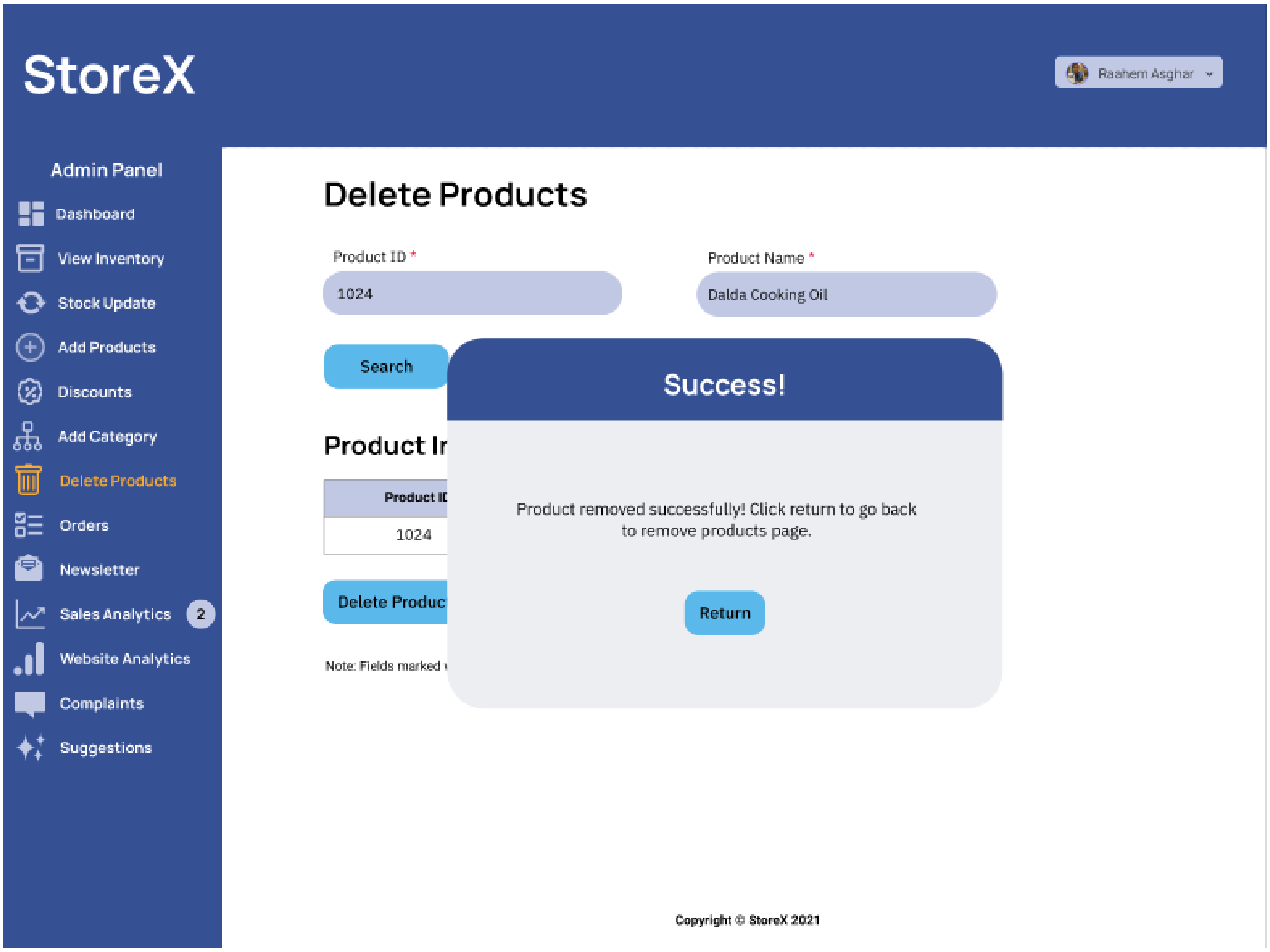
Product Name * Dalda Cooking Oil

Product ID
1024

Note: Fields marked

Are you sure you want to delete the product? Click 'Yes' to proceed or 'Return' to return to previous page.

Fig 5.3.21 (c)



Delete Products

Product ID * 1024

Product Name * Dalda Cooking Oil

Product ID
1024

Note: Fields marked

Success!

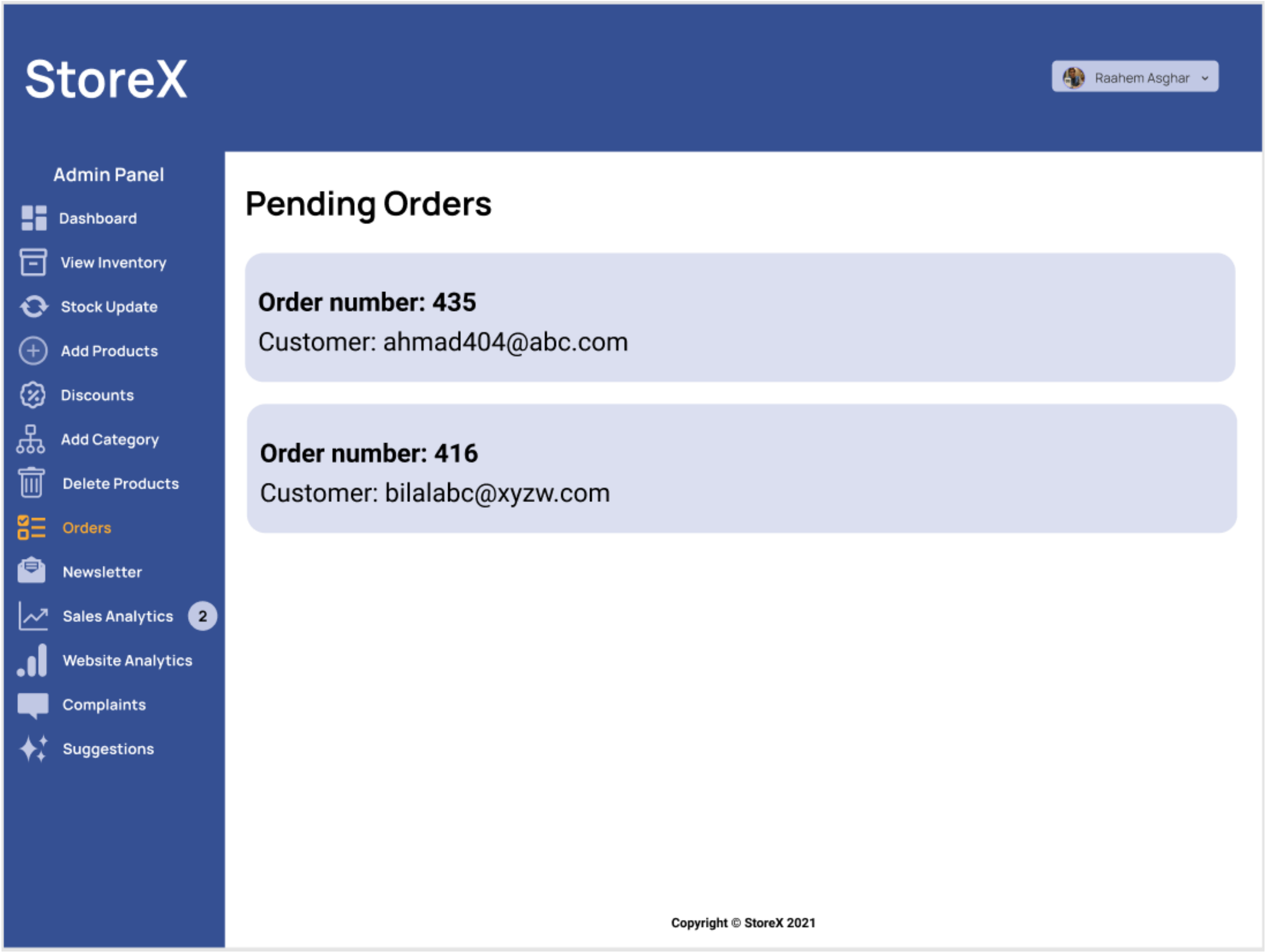
Product removed successfully! Click return to go back to remove products page.

Copyright © StoreX 2021

Fig 5.3.21 (d)

Currently, the admin is on the "Delete Products" screen (the name has been highlighted in the side bar). Here the admin enters the name and ID of the product the admin wishes to delete. Once the admin has entered the respective details, he/she presses search to see if the product actually exists (in case he/she enters incorrect information or doesn't remember if the product is in the store or not). If the product exists, then it is displayed with the current stock in the table. If it does not exist, message is displayed showing "No match found" (Fig 5.3.21 (b)). The admin presses the "Delete Product" button. A message is displayed asking for confirmation (Fig 5.3.21 (c)). If the admin actually wishes to delete the product, he/she presses "Yes" causing a success message to be displayed (Fig 5.3.21 (d)). The database is updated accordingly. If the admin does not want to delete the product, he/she presses "Return" to return to the previous screen.

5.3.22 **View Orders – Admin**



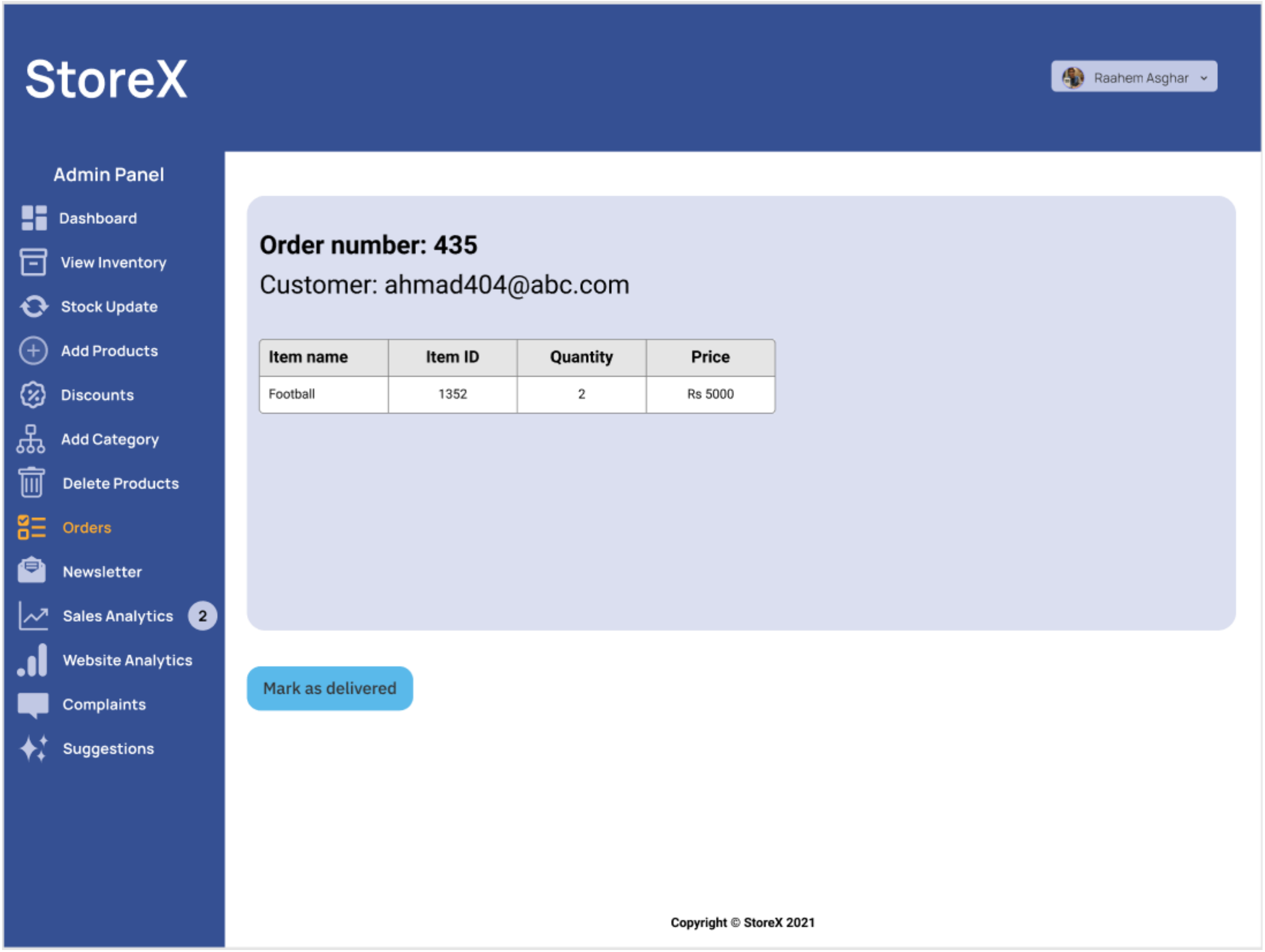
Pending Orders

Order number: 435
Customer: ahmad404@abc.com

Order number: 416
Customer: bilalabc@xyzw.com

Copyright © StoreX 2021

Fig 5.3.22 (a)



Order number: 435

Customer: ahmad404@abc.com

Item name	Item ID	Quantity	Price
Football	1352	2	Rs 5000

Fig 5.3.22 (b)

When admin clicks the pending orders in the tool bar then this screen is opened. It gives the list of orders and its users (Fig 5.3.22 (a)). The admin can then select the orders that have been delivered. Now the full details of that order are shown. The admin can now click the option of mark as delivered. This will remove the order from the pending order list showing that it has been dealt with (Fig 5.3.22 (b)).

5.3.23 **Send Newsletter** – Admin

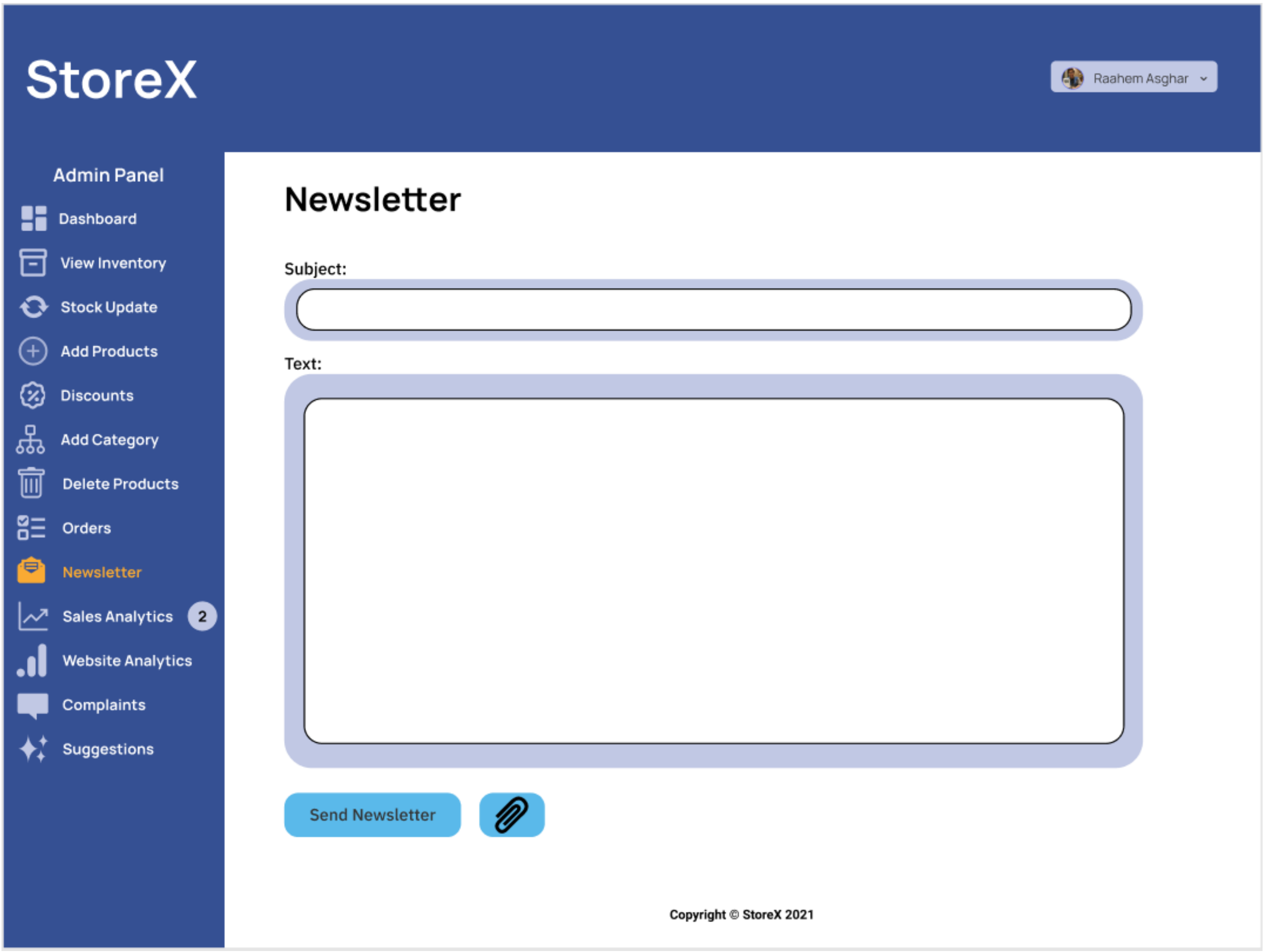
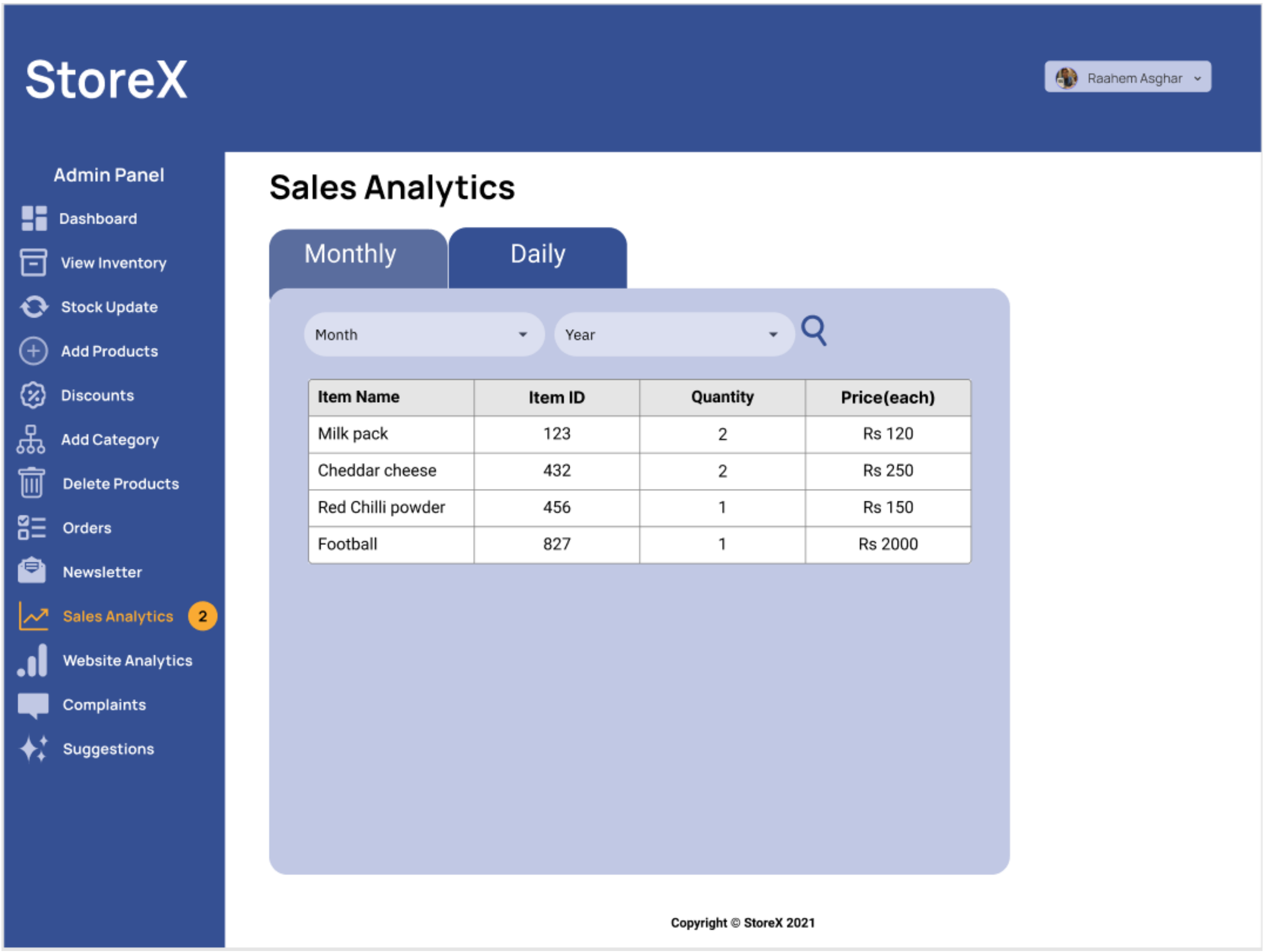


Fig 5.3.23

The newsletter screen allows the admin to send an email to all the users that subscribe to the newsletter. Admin can write the text they want to send. The screen also gives the option of attaching an image to the message. Allowing the admin to be able to send flyer as images to the users as well. The text and image is then sent to the server which then goes through the list of all subscribers and emails the admin's message.

5.3.24 **Sales Analytics** – Admin



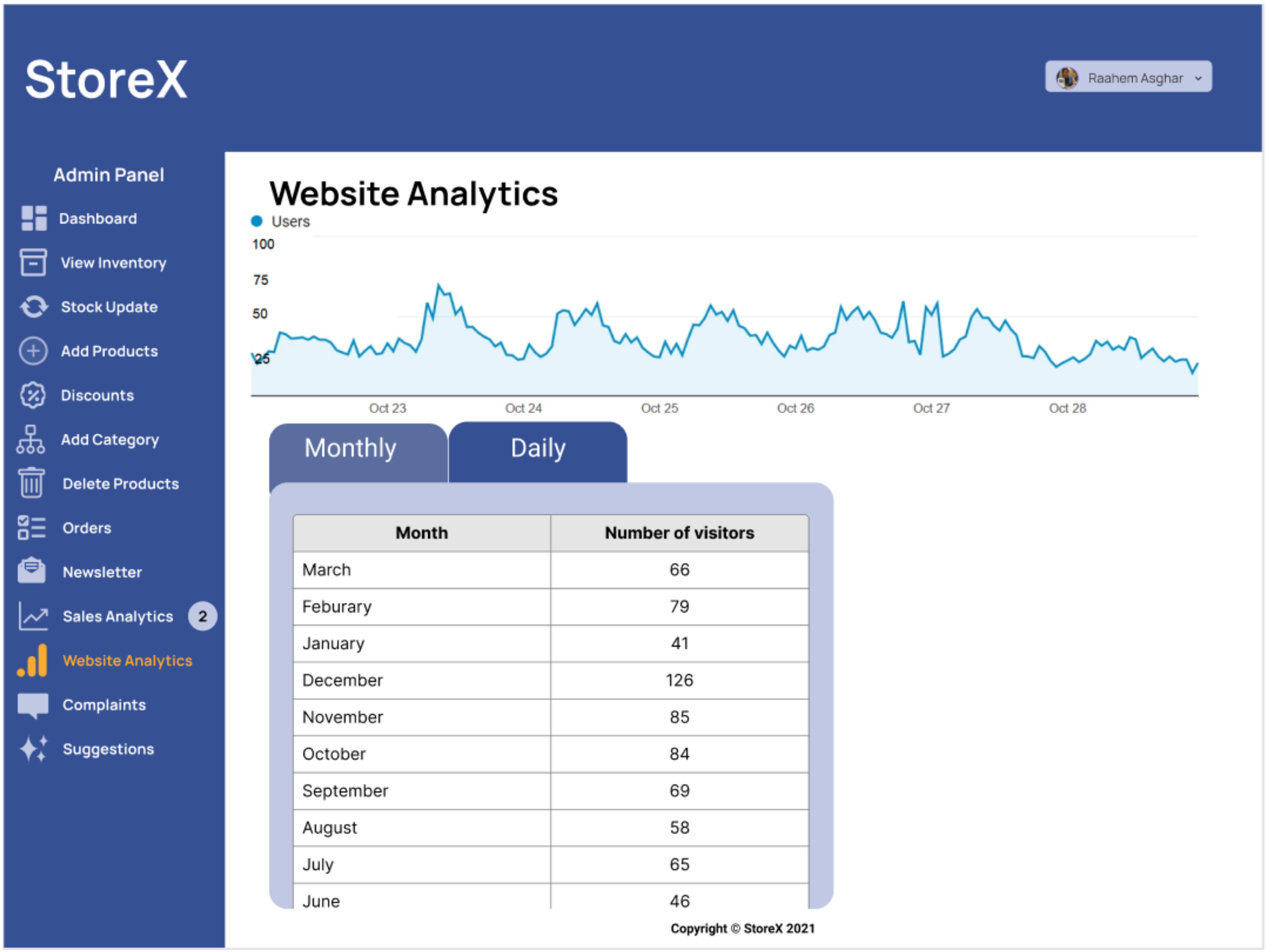
Sales Analytics

Item Name	Item ID	Quantity	Price(each)
Milk pack	123	2	Rs 120
Cheddar cheese	432	2	Rs 250
Red Chilli powder	456	1	Rs 150
Football	827	1	Rs 2000

Fig 5.3.24

This screen allows the admin to see the number of times sold over in a month or a day. If the user selects the monthly option, they can use the dropdown menu to select the year and month. Then the items sold in that month will be displayed in descending order according to the quantity of items sold. Instead, is the user picked the daily option then they are shown the number of items for a specific day in the month same way as in monthly option. Both of these cases fetch the data from the server and displays them in the form of a table.

5.3.25 **Website Analytics** – Admin

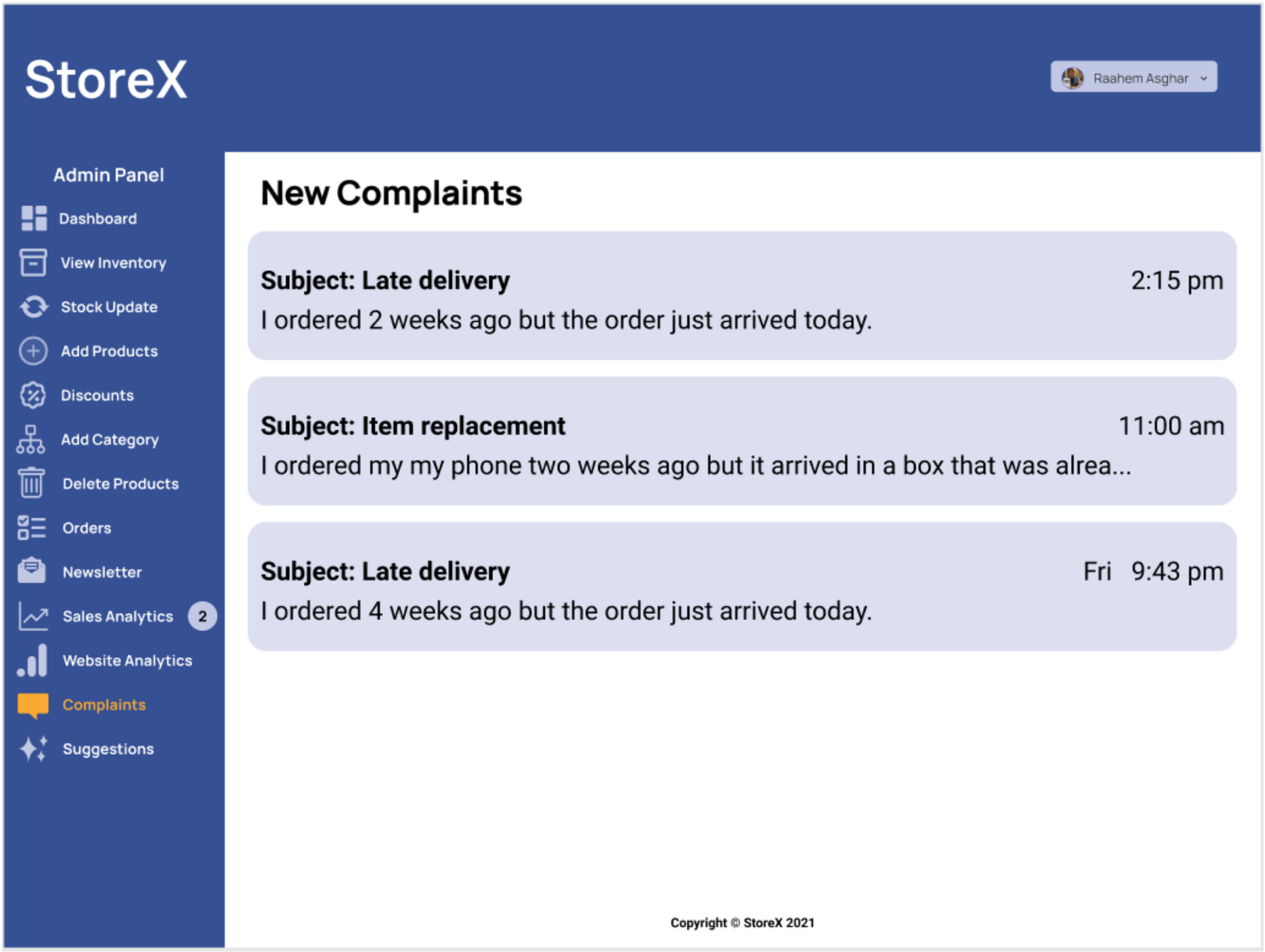


Month	Number of visitors
March	66
Feburary	79
January	41
December	126
November	85
October	84
September	69
August	58
July	65
June	46

Fig 5.3.25

This screen shows the admin number of visitors the site has gotten. The graph above shows the overview of the number of visitors over the last few days. For the graph the data stored in the database is fetched and is converted to a graph. If he/she wants a detailed data then they can use the table below the graph. If the monthly is clicked on, then the information is shown according to the months in descending order. While if he/she clicked on the daily option then the same information if displayed according to the days in descending order. Both of these cases fetch the data from the server and displays them in the form of a table.

5.3.26 **View Complaints** – Admin



New Complaints

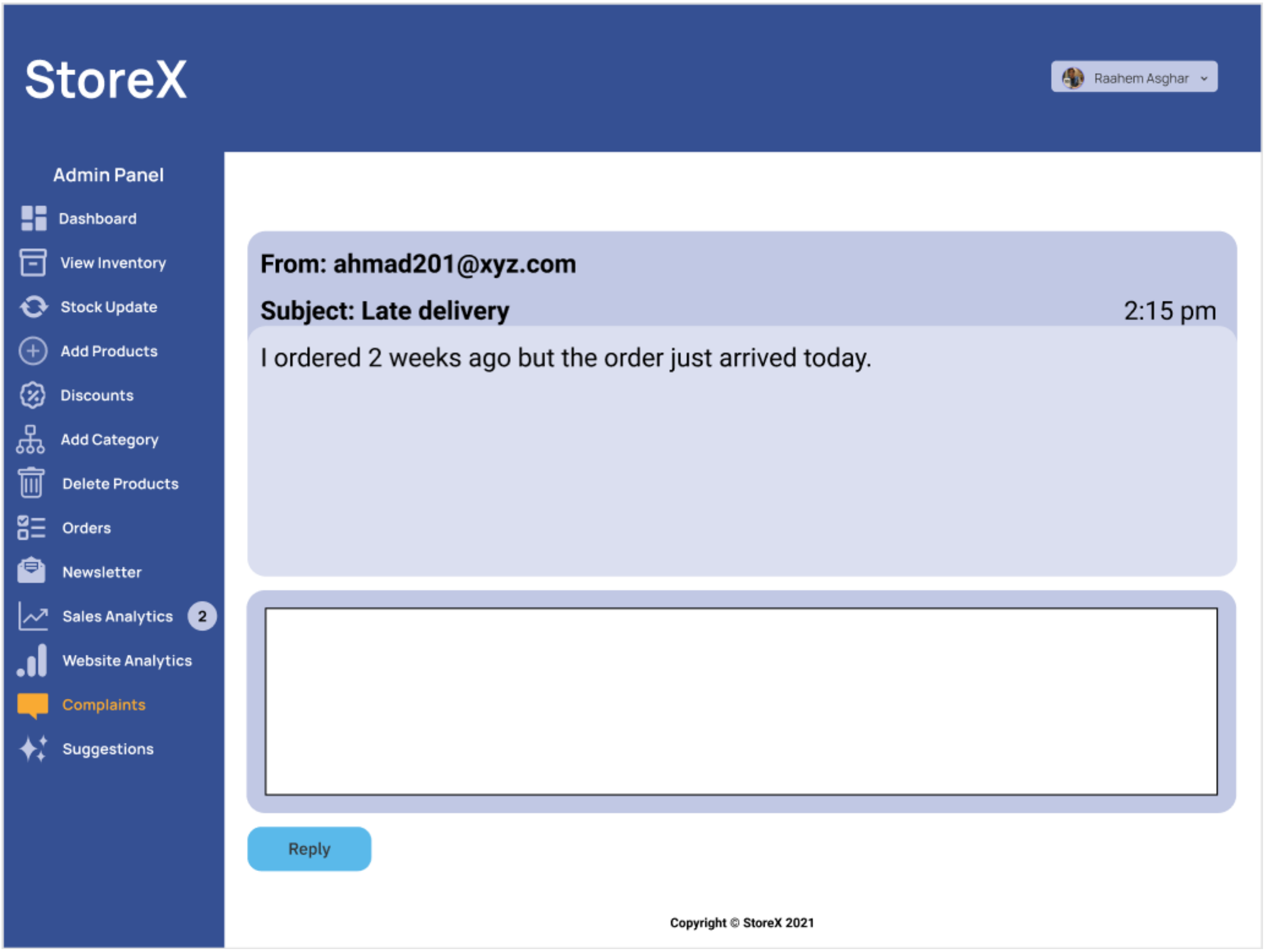
Subject: Late delivery 2:15 pm
I ordered 2 weeks ago but the order just arrived today.

Subject: Item replacement 11:00 am
I ordered my my phone two weeks ago but it arrived in a box that was alrea...

Subject: Late delivery Fri 9:43 pm
I ordered 4 weeks ago but the order just arrived today.

Copyright © StoreX 2021

Fig 5.3.26 (a)



From: ahmad201@xyz.com

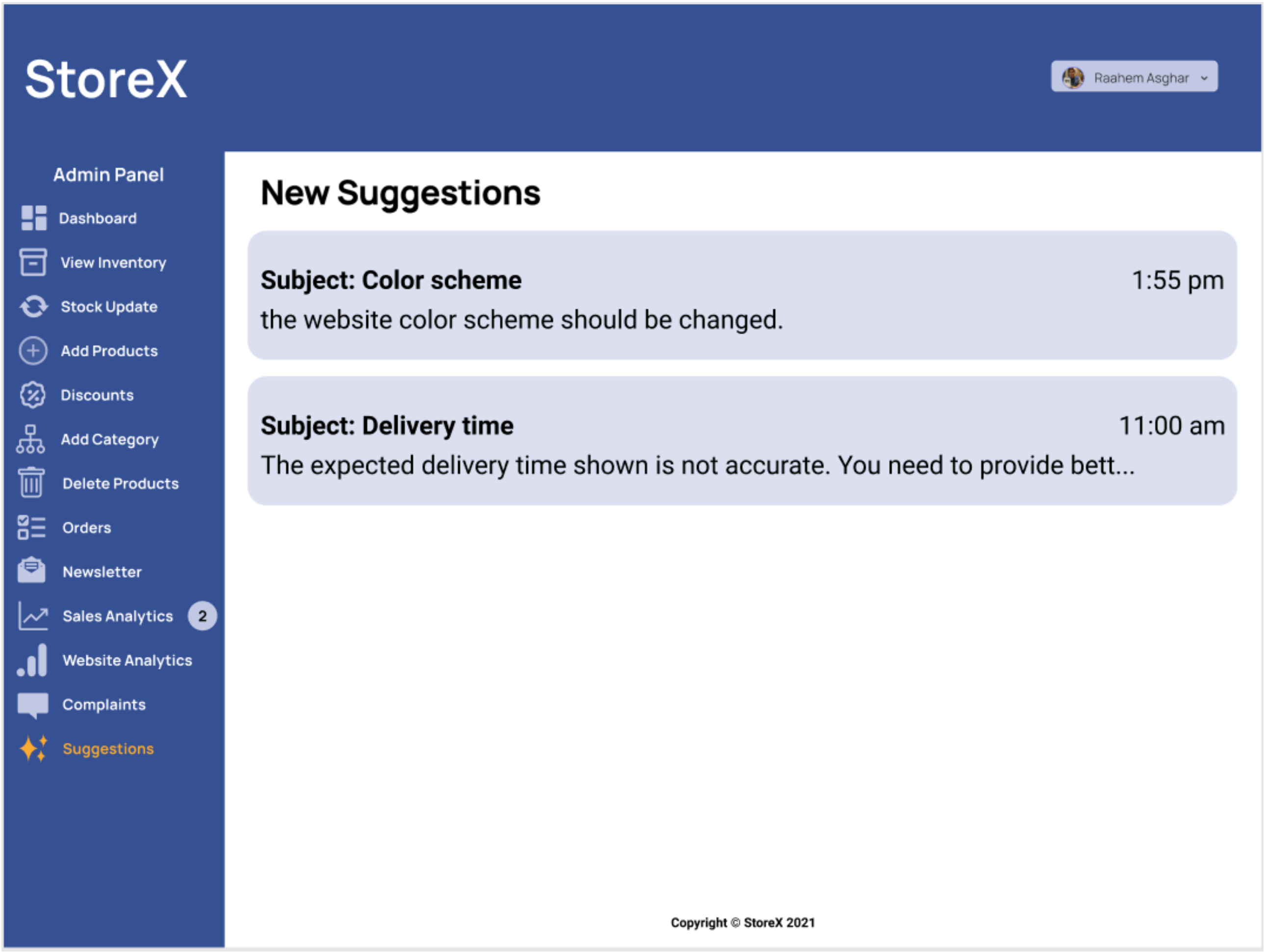
Subject: Late delivery 2:15 pm

I ordered 2 weeks ago but the order just arrived today.

Fig 5.3.26 (b)

Complaints screen allows the admin to see the complaints received and reply to them if needed. When the admin clicks on complaints on the tool bar, he/she is given a list of messages received in descending order by time (Fig 5.3.26 (a)). He/she can click on the message they want to read. This would open the full message and also give a text box to reply if he/she needs to (Fig 5.3.26 (b)).

5.3.27 **View Suggestions** – Admin



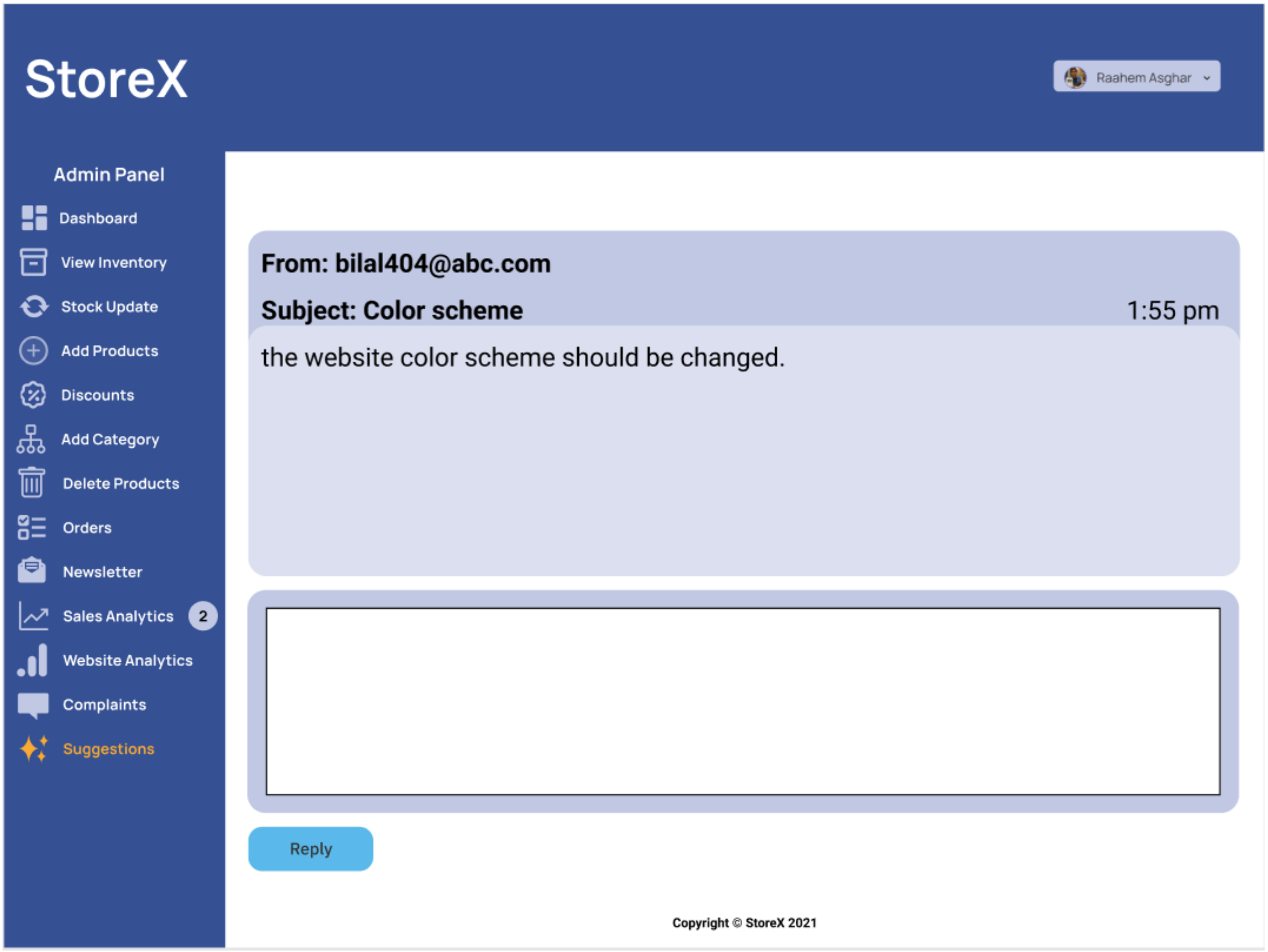
New Suggestions

Subject: Color scheme 1:55 pm
the website color scheme should be changed.

Subject: Delivery time 11:00 am
The expected delivery time shown is not accurate. You need to provide bett...

Copyright © StoreX 2021

Fig 5.3.27 (a)



From: bilal404@abc.com

Subject: Color scheme 1:55 pm

the website color scheme should be changed.

Fig 5.3.27 (b)

Suggestion's screen allows the admin to see the suggestions received and reply to them if needed. When the admin clicks on suggestions on the tool bar, he/she is given a list of messages received in descending order by time (Fig 5.3.27 (a)). He/she can click on the message they want to read. This would open the full message and also give a text box to reply if he/she needs to (Fig 5.3.27 (b)).

5.4 User interface design rules

5.4.1 Consistency

The entire interfaces of the admin and user side were made while keeping consistency in mind i.e. the same color schemes were followed throughout. Moreover, the layout of different features along with the fonts and capitalization of text were kept the same. Other than that, identical terminology was used in prompts, fields and menus.

5.4.2 User Familiarity

Icons that the users are familiar with have been used throughout the store. Other than that, terminology related to an E commerce web application have been used so that the user does not find any difficulty in navigating the different parts of the web application. Both sides of the web application have been designed keeping both type of actors in mind i.e. the admin and the users.

5.4.3 User Guidance

In case of alternative flows or errors, appropriate messages will be displayed, guiding the user on what to do to in order to complete the task that he/she was trying to do. Other than that, the user can locate himself/herself on the web application using the sidebar.

5.4.4 Minimal Surprise

The user will be able to perform the exact task for which the screens and buttons are designed for i.e. there will be no surprise component. For example, if the admin presses “Add Product”, only the product will be added with a success message, no other task will be performed.

6 Other Non-functional Requirements

6.1 Performance Requirements

- The response time for the web application shall not exceed 0.5s.
- The web application should be able to handle 100,000 concurrent users at a time.
- The system must implement standard web security practices to prevent any chance of attack.
- The downtime for the web application shall not exceed 30 minutes on any day in-order to prevent significant loss in revenue for the owner.
- Web application maintenance should be easy and cost-effective such that the maintenance cost shall not exceed 10% of the total cost each year.
- The UI should be user friendly such that it should not take more than 5 minutes for any new user to understand the interface.

6.2 Safety and Security Requirements

6.2.1 Safety Requirements

- The web application should safely store user's data ensuring that there is no data breach. It should prevent attackers from being able to get access to user.
- It should use HTTPS instead of HTTP because data transfer over HTTP is not encrypted. Data is transferred between systems as plain text if HTTP is used as a protocol for communication. HTTPS on the other hand ensures end-to-end encryption of data when transferring it between systems. Hence many users don't even use e-commerce platforms that use HTTP for end to end communication.
- There should be an application gateway between the system and the rest of the network so that all the network traffic is received at this gateway and the gateway only allows authorized packets to reach the system.

6.2.2 **Security Requirements**

- The web application should be secure from DoS attacks
- Alongside using cookies, the web application e should have a unique token for every round trip for every user to prevent CSRF attacks
- The web application should follow the Same Origin Policy.
- Ensure Validation of cookies and query strings against a rigorous specification of what should and should not be allowed. This will ensure prevention of XSS attacks.
- Query structure should be specified independent of the user to prevent SQL injection attacks Frame-busting must be implemented to ensure that frames can not be included in other websites.

6.3 Software Quality Attributes

6.3.1 Usability

The web-based application should be user-friendly i.e. should be easy to learn, navigation should be simple. For the store to be user friendly, a simple, clean, responsive, intuitive and reliable design will be required. Other than that, a guide will be prepared to aid new users, which will be available on the web application.

6.3.2 Reliability

The system shall be used by multiple users simultaneously. It is desired that the web application does not crash when multiple users are logged in or are browsing simultaneously. The data should be stored in a secure hard disk and data should be backed up regularly to prevent any loss of data. All in all, the users shall be able to access the web application 98% of the time without any failures.

6.3.3 Performance

Our system will have a very fast response time i.e. the site should load in 3 seconds when the number of simultaneous users are > 5000. Each request should be processed within 10 seconds.

6.3.4 Portability

Portability is established in terms of operating systems, hardware devices, browsers, software systems, and their versions. For now, a cross-platform, cross-browsing, and mobile-responsive solution is a common standard for web applications. Our web based system will be accessible on all latest devices i.e. mobile phones, laptops, tablets.

6.3.5 Availability

It is desired that any user shall be able to buy a product whether it's 6AM in the morning or whether its 12PM at midnight as it is designed to be a 24 hours store. In the case of unplanned system downtime, all features will be available again after maximum one working day.

6.3.6 Maintainability

Each page of our store will contain a button which would redirect the user to form where they could report any bugs or errors. These bugs and errors will be discussed by the support team and they would aim to fix these errors as soon as possible (support team might alert the users that the particular feature would be down for maintenance).

7 Appendix A - Group Log

Meeting Date	Remarks	Duration
25th February, 2021	Overview of the SDS document	1 hour
28th February, 2021	General Discussion (plan)	2 hours
2nd March, 2021	Dividing the workload	1 hour
7th March, 2021	Reviewing sections done	2 hours
8th March, 2021	Color Scheme of the UI & Figma team	3 hours
10th March,2021	Divided the screens amongst ourselves	2 hours
12th– 15th March 2021	Screen Design and Discussion	20 hours
16th - 18th March, 2021	Making of diagrams in SDS	20 hours
19th – 20st March, 2021	Remaining Sections	10 hours
21st March, 2021	Final touches to the document	4 hours
Total		**65 hours**

8 Appendix B – Contribution Statement

Name	*Contributions in this phase*	*Approx. Number of hours*	*Remarks*
Mati Ur Rehman	All the work was equally divided among the group members. The SDS sections which I contributed to are 3.2, 4.5. I also made the activity diagram and sequence diagram of shopping cart. Moreover, I also designed figma screens related to shopping cart, footer and jointly collaborated on the design of homescreen, DB schema, class diagram and component diagram with my team members.	50	This phase of the project was full of learning for me. I got an hands on experience of making SDS and learnt various aspects about system planning and documentation. The most interesting part of SDS was making application screens. It required a lot of brainstorming from coming up with the color scheme to designing a user friendly layout. Overall, it was a great experience working with the team and I learnt a lot.
Muhammad Raahem Asghar	I completed Sections 3.3, 5.4. The activity diagram of search products. 7 Figma screens of the admin side along with the use case mapping. Helped in the making of DB schema, class diagram, component diagram, information architecture with my team members. Suggestions for design of the admin and user screens. Other than that, I helped in the overall formatting of the document.	50	This phase of the project made me understand the importance of planning everything before moving on to development of the final product. Also the importance of designing a user friendly interface. This phase helped me understand the importance of teamwork, how easy it is to do task when it is divided amongst team members. The most interesting and fun part was the designing and making of the screens

			on Figma. All in all, documentation is very important in software engineering.
Muhammad Hamza Shahzad	I completed Section 2, Section 3.1 and section 4.4. I also made The login/signup screen, the account settings screen, the contact us screen, the sale screen and contributed towards making the Homepage screen. Additionally, I made the sequence diagram of search, and contributed towards making the component diagram, the class diagram and the DB schema	50	This phase of the project helped me realize the importance of planning in an engineering discipline. As I write this, we approach the end of our SDS, having planned each and everything of our project in theory. Theoretically, we have completely set up our system including its screen designs, db schema, and important transitioning and flow diagrams. I really enjoyed setting up all the components, connecting them with eachother and integrating the entire system as a whole while ensuring harmony amongst the screens and the DB. Overall, it was a great learning experience for me.
Muhammad Haris	I completed section 5.2 and section 4.3. I also made Admin's screens which include dashboard, send newsletter, complaints, suggestions, pending orders, sales analytics and web analytics. I made the sequence diagram for add product screen. I also contributed to making DB schema, class diagram, component diagram, information architecture with my team members.	50	In this part of the project I learned about how to make an SDS document and its importance. I learned how important it is to plan all the components before starting the development phase of a project. This phase has given us a clear direction which will guide us through the next phase. All the planning meetings were

			a great learning experience and helped in learning about collaboration.
Muhammad Hamza Sajjad	I designed eight user screens, which include 3 screens for product details page, 2 screens for My Orders page, search results page, wishlist, and the category ABC page, and wrote the respective use case mapping for all these screens in subsections of section 5.3. I contributed in section 3.3 too. I also made the activity diagram for add product use case. I also contributed to the making of DB schema, class diagram and component diagram. I also helped in the making of header and footer and in adding it to all user screens. I also contributed to deciding the colour scheme and overall design of our web app StoreX. I also reviewed the formatting of the SDS document before its submission.	50	This phase of the project made me understand the importance of the designing and planning part of software engineering, as it actually helps you lay the foundations and gives you clear directions for the development phase. Without organizing the schemas and other diagrams, and without the user and admin screens, development would have been really haphazard. Having the bigger picture in front of you, as well as how different screens and components relate to and interact with each other is of immense help before starting development. Collaborating with my teammates was a great learning experience too, and it also taught me the importance of effective teamwork in group projects like this.