

A  
Project Report  
On

# SHOPZEN

Ecommerce Website  
**BTech-IT, Sem VI**

Prepared by:  
**HELI VACHHANI (IT049),**  
**KANSAGARA MITSU NITIN (IT054),**

Guided by:  
**Prof. Anand Dave**  
Dept. of Information Technology



**Department of Information Technology,  
Faculty of Technology,  
Dharmsinh Desai University  
College Road, Nadiad-387001.**

**April-2021**

## **CANDIDATE'S DECLARATION**

We declare that the semi-final year report entitled “Shopzen” is our own work conducted under the supervision of the internal guide Mr. Anand Dave from DDUniversity.

We further declare that to the best of our knowledge the report for B.Tech. semi-final year does not contain part of the work which has been submitted for the award of B.Tech. Degree without proper citation.

Candidate's Name: Heli Vachhani  
Student ID: 18ITUON113

Candidate's Name: Kansagara Mitsu Nitin  
Student ID: 18ITUON118

**DHARMSINH DESAI UNIVERSITY**  
**NADIAD - 387001**



**CERTIFICATE**

This is to certify that the project carried out in the subject of System Design Practice, entitled 'SHOPZEN' and recorded in this report is a bonafide report of work of

- 1) **Ms. HELI VACHHANI** Roll No. **IT049** ID No: **18ITUON113**  
2) **Ms. KANSAGARA MITSU NITIN** Roll No. **IT054** ID No: **18ITUON118**

of the Department of Information Technology, semester VI . They were involved in Project work during the academic year 2020 -2021.

**Prof. Anand Dave**

(Project Guide)

Department of Information Technology,  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad

Date:

**Prof. Vipul Dabhi**

Head, Department of Information Technology,  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad

Date:

## **ACKNOWLEDGMENT**

On the very outset of this report, we would like to extend our sincere & heartfelt obligations towards all the personages who have directly or indirectly helped in this endeavor. Without their active guidance, help, cooperation & encouragement, we wouldn't have made the headway in the project.

We are ineffably indebted to our instructor Mr. Anand Dave for his guidance, encouragement and conscientiousness to accomplish this project. We are extremely thankful for his support.

We would also like to express our gratitude to our Head, Mr. Vipul Dabhi along with Dharmsinh Desai University for giving us this opportunity.

Last but not least, we would also like to acknowledge with a deep sense of reverence, our gratitude towards our family members who have always supported us morally.

Any omission in this brief acknowledgement does not mean lack of gratitude.

Thank You,

- 1. Heli Vachhani**
- 2. Kansagara Mitsu Nitin**

# TABLE OF CONTENTS

<b>CANDIDATE'S DECLARATION</b>	<b>i</b>
<b>CERTIFICATE</b>	<b>ii</b>
<b>ACKNOWLEDGMENT</b>	<b>iii</b>
<b>TABLES</b>	
LIST OF FIGURES	vi
LIST OF TABLES	vi(conti.)
<b>ABSTRACT</b>	<b>vii</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Project Details	1
1.2 Purpose	1
1.3 Scope	1
1.4 Objective	2
1.5 Technology And Literature Review	2
<b>2. PROJECT MANAGEMENT</b>	<b>3</b>
2.1 Feasibility Analysis	3
2.1.1 Technical Feasibility	3
2.1.2 Economic Feasibility	3
2.2 Project Planning	3
2.2.1 Project Development Approach And Justification	3
2.2.2 Project Plan	5
2.2.3 Roles And Responsibilities	5
2.2.4 Group Dependencies	5
2.3 Project Scheduling	6
<b>3. SYSTEM REQUIREMENT STUDY</b>	<b>7</b>
3.1 Current System Study	7
3.2 Weakness/Problems Of Current System	7
3.3 Types Of Users Of This Platform	7
3.4 Minimum Hardware/Software Requirements	8
3.5 Constraints	8
3.6 Assumptions & Dependencies	8
<b>4. SYSTEM ANALYSIS</b>	<b>9</b>
4.1 REQUIREMENTS OF NEW SYSTEM(SRS)	9
4.1.1 Functional Requirements	9
4.1.1.1 User Requirements	10
4.1.1.2 Admin Requirements	14
4.1.2 Non-Functional Requirements	16
4.2 Database Design	17
<b>5. SYSTEM DESIGN</b>	<b>18</b>
5.1 Use-Case Diagram	18
5.2 Class Diagram	20
5.3 Sequence Diagrams	21
5.4 Component Diagram	30
5.5 Deployment Diagram	30

<b>6</b>	<b>IMPLEMENTATION PLANNING</b>	<b>31</b>
6.1	IMPLEMENTATION ENVIRONMENT	31
6.2	PROGRAMS/MODULES SPECIFICATION	31
6.3	CODING STANDARDS	31
<b>7.0</b>	<b>TESTING</b>	<b>32</b>
7.1	TESTING PLAN	32
7.2	TESTING STRATEGY	32
7.3	TESTING METHODS	32
7.4	TEST CASES	32
<b>8.0</b>	<b>USER MANUAL</b>	<b>35</b>
<b>9.0</b>	<b>LIMITATION AND FUTURE ENHANCEMENT</b>	<b>64</b>
9.1	Limitation	64
9.2	Future Enhancements	64
<b>10.</b>	<b>CONCLUSION AND DISCUSSION</b>	<b>65</b>
10.1	Conclusion	65
10.2	Discussion	65
10.2.1	Self-Analysis of Project Viabilities	65
10.2.2	Problems Encountered and Possible Solutions	65
10.2.3	Summary of Project Work	65
<b>11.</b>	<b>BIBLIOGRAPHY</b>	<b>66</b>

## LIST OF FIGURES

<b>Figure Index</b>	<b>Name</b>	<b>Pg. no</b>
Fig 2.2.1	Iterative Waterfall Model	3
Fig 4.2	MongoDB database Model	28-29
Fig 5.1	Use Case Diagrams for Customer and Admin	18-19
Fig 5.2	Class Diagram	20
Fig 5.3.1.1	Registration and Verification	21
Fig 5.3.1.2	Login	22
Fig 5.3.1.3	Forgot and Reset password	22
Fig 5.3.1.4	Search Product	23
Fig 5.3.1.5	Add to Cart	23
Fig 5.3.1.6	Cart -Edit/ Remove Product	24
Fig 5.3.1.7	Place Order	24
Fig 5.3.1.8	Payment	25
Fig 5.3.1.9	Logout	26
Fig 5.3.2.1	Edit Shop details	26
Fig 5.3.2.2	Edit Product details	27
Fig 5.3.2.3	Add Product details	27
Fig 5.3.2.4	Search Orders	28
Fig 5.3.2.5	Edit Orders	28
Fig 5.3.2.6	View Orders	29
Fig 5.3.2.7	View Customer	29
Fig 5.4	Component Diagram	30
Fig 5.5	Deployment Diagram	30
Fig 8.1	Registration Page	36

Fig 8.2	Registration Page Popup	36
Fig 8.3	User Mail	37
Fig 8.4	Set Password to complete Registration	37
Fig 8.5	Complete Registration	38
Fig 8.6	Login Page	38
Fig 8.7	Login Failed Page	39
Fig 8.8	Login with Google	39
Fig 8.9	Home Page	40
Fig 8.10	Forgot Password	41
Fig 8.11	Forgot Password continue	42
Fig 8.12	Forgot Password continue	42
Fig 8.13	Set New Password	43
Fig 8.14	Login with New Password	43
Fig 8.15	Shop Page	44
Fig 8.16	Search	44
Fig 8.17	Filter List (list to filters that can be applied)	45-46
Fig 8.18	Filter Applied	47
Fig 8.19	Add to Cart	47
Fig 8.20	Add to Cart sidebar popup	48
Fig 8.21	Cart Page	48
Fig 8.22	On Click Proceed to Checkout	49
Fig 8.23	Complete purchase, Enter card details	49
Fig 8.24	Pay	50
Fig 8.25	See Purchase History	50
Fig 8.26	On Click Cash On Delivery	51

Fig 8.27	On Click Proceed to Checkout	51
Fig 8.28	User Purchase History Page with Single order purchase	52
Fig 8.29	User Purchase History Page with Multiple order purchases	52
Fig 8.30	Invoice Download	53
Fig 8.31	Reset Password	53
Fig 8.32	Product Rating (and display Average Rating)	54
Fig 8.33	Wishlist (and how to wishlist a Product)	55
Fig 8.34	Related Products	56
Fig 8.35	Add Product	57
Fig 8.36	Update Product	57
Fig 8.37	Add Brand Category	58
Fig 8.38	Edit/Update Brand Category	58
Fig 8.39	Add Sub Categories under Brand	59
Fig 8.40	Edit/Update Subcategories under Brand	59
Fig 8.41	Set/Update Delivery Status	60
Fig 8.42	Create Coupon	61
Fig 8.43	Apply Coupon, if Invalid display Error	62
Fig 8.44	Firebase Authentication	63
Fig 8.45	Real Time Payment Statistics	63

## LIST OF TABLES

Table Index	Name	Pg. no
Table 2.3	Project Scheduling Chart	6
Table 7.3	Testing Cases	33-34

## **ABSTRACT**

SHOPZEN is created with the aim to provide an efficient eCommerce platform which facilitates to sell goods to buyers with an on-click facility E-Commerce features.

In the current COVID pandemic situation, people don't have access to local stores and supermarkets, hence this platform is the perfect choice for the customers to buy day to day essential products, just by on-click facility from the comfort of their homes without the need of stepping out.

It has a very friendly user interface, very easy to use which can accurately filter out user's requirements and give the appropriate result for the product that the user wants to buy. SHOPZEN platform can also be enhanced by incorporating various features that we listed in the following pages of this document below under scope and future enhancement.

# 1. INTRODUCTION

## 1.1 Project Details -

Title: ***Online Shopping store(SHOPZEN)***

Shopzen is created with the aim to provide a very efficient eCommerce platform which facilitates to sell goods to buyers with an on-click facility E-Commerce features.

Features:

Implemented all the features any random ecommerce website would've ranging from:

- Login
- Logout
- Orders
- Search-bar
- Intensive filters
- Add to Cart
- Wishlist
- Coupon
- Cash On Delivery
- Realtime Payment
- Email Authentication
- Rating modal
- to Invoice Download

## 1.2 Purpose -

We developed an eCommerce website which can be used to sell any goods. A single platform where all types of products could be sold by registered sellers only. If a user wants to sell their products, they need to first verify the national and international quality standards of the product(s) they want to sell and if they are allowed to sell, they first have to get registered under us and then they are entitled to sell their products on our website.

## 1.3 Scope -

Although, we are starting from a very basic eCommerce platform to sell goods to buyers with an on-click facility but in future we can expand the scope to adding assistants like google assistant which will hear the needs of the user and filter out the option and show the appropriate product.

Another scope can be to expand the number of places/ locations/ states/ country the product can be delivered to the buyer.

And another scope can be to increase the range of variety of products under each category and increase the number of different types of categories by making more associations with concerned companies/ sellers.

## 1.4 Objective -

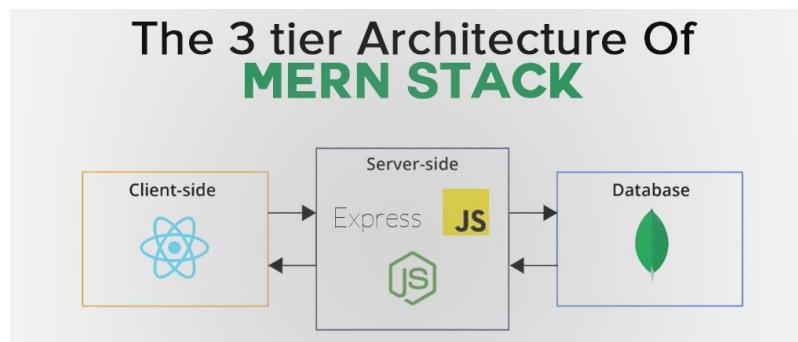
We aim at providing an efficient platform with a very friendly user interface which can accurately filter out user's requirements and give the appropriate result for the product that the user wants to buy.

## 1.5 Technology and Literature Review -

MERN Stack, cloudinary, stripe online payment, firebase (only for authentication)

**MERN** stack is a web development framework. It consists of MongoDB, ExpressJS, ReactJS, and NodeJS as its working components. Here are the details of what each of these components is used for in developing a web application when using MERN stack:

- **MongoDB**: A document-oriented, No-SQL database used to store the application data.
- **NodeJS**: The JavaScript runtime environment. It is used to run JavaScript on a machine rather than in a browser.
- **ExpressJS**: A framework layered on top of NodeJS, used to build the backend of a site using NodeJS functions and structures. Since NodeJS was not developed to make websites but rather run JavaScript on a machine, ExpressJS was developed.
- **ReactJS**: A library created by Facebook. It is used to build UI components that create the user interface of the single page web application.



As shown in the illustration above, the user interacts with the **ReactJS** UI components at the application front-end residing in the browser. This frontend is served by the application backend residing in a server, through **ExpressJS** running on top of NodeJS. Any interaction that causes a data change request is sent to the **NodeJS** based Express server, which grabs data from the **MongoDB** database if required, and returns the data to the frontend of the application, which is then presented to the user.

We have also used **CLOUDINARY** to render **multiple** images for our products on our website. **Cloudinary** is an end-to-end image-video-management solution for websites and mobile apps, covering everything from image-video uploads, storage, manipulations, optimizations to delivery.

## 2. PROJECT MANAGEMENT

### 2.1 Feasibility Analysis

#### 2.1.1 Technical Feasibility -

We analysed the technical needs of our project and we can come down to a conclusion using the below given things.

Backend and Frontend: MERN

Tracking / Version Control: Git and Github

#### 2.1.2 Economic Feasibility-

Any developer develops the project to get maximum return on the invested amount. Here, the technology and tools we are using are basically available for free on the internet and are open source. Hence, minimal economic resources like internet connectivity, good functioning desktop/laptop, electricity, man-power, etc.. In future, we can earn from advertisements on our platform once we have a nice position in the market. So, our proposed system is economically feasible in our opinion.

### 2.2 Project Planning

#### 2.2.1 Project Development Approach And Justification

For our software development we used: Iterative waterfall model

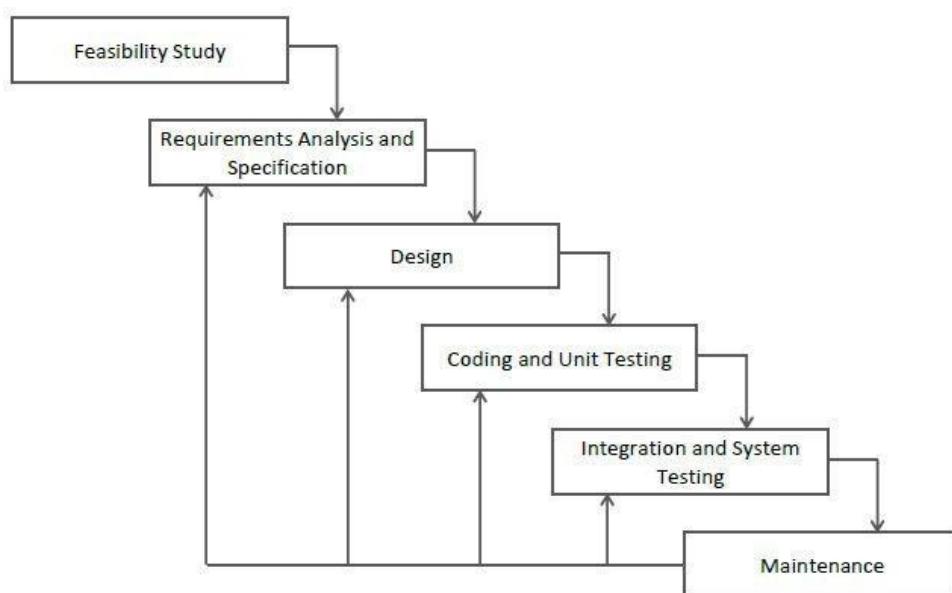


Fig 2.2.1 Iterative Waterfall Model

**Model features:**

- Iterative Waterfall Model is the extension of the Waterfall model.
- This model is almost the same as the waterfall model except some modifications are made to improve the performance of the software development.
- The iterative waterfall model provides customer's feedback paths from each phase to its previous phases.
- There is no feedback path provided for the feasibility study phase, so if any change is required in that phase then the iterative model doesn't have scope for modification or making corrections.
- Iterative waterfall allows us to go back to the previous phase and change the requirements and some modification can be done if necessary.
- This model reduces the developer's effort and time required to detect and correct the errors.
- In an iterative waterfall model, the next phase can only begin when the previous phase is completed as a waterfall model.

**Phases of the model:**

1. Requirement Analysis
2. Feasibility Study
3. Software Design
4. Coding/Implementation
5. Software Testing
6. Software Deployment
7. Software Maintenance

**Advantages of Iterative Waterfall model:**

1. Testing and debugging during smaller iteration is easy.
2. A Parallel development can plan.
3. It is easily acceptable to ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

**Disadvantages of Iterative Waterfall model:**

1. It is not suitable for smaller projects.
2. More Resources may be required.
3. Design can be changed again and again because of imperfect requirements.
4. Requirement changes can cause over budget.
5. Project completion date not confirmed because of changing requirements.

**2.2.2 Project Plan**

Our project is web based and hence follows client-server paradigm. Also, we follow object oriented, MVC based design methodology for this platform. We have decomposed this system in high level modules. These modules will further be analysed and developed using OO design and considering MVC approach. After that, all modules will be integrated as components of a single system.

**2.2.3 Roles And Responsibilities**

We have done everything parallelly.

**2.2.4 Group Dependencies:****The dependencies among the tasks include the following:**

- Analysis or System Requirement Study (SRS) is independent of all, yet will be started after completion of feasibility study and project planning.
- Designing of prototypes can be done simultaneously with system analysis.
- Development of the project is preceded by the designing of prototype and system analysis.
- Testing can be only done once the development of some major functionalities are completed and are ready to be tested.
- Documentation is independent of all the tasks and can be done as the other tasks proceed.

## 2.3 Project Scheduling

Scheduling the project tasks is an important project planning activity. It involves deciding which task should be taken up and when in order to schedule the project activities; a software project manager needs to be the following :

1. Identify all the tasks needed to complete the project.
2. Break down large tasks into smaller activities.
3. Determine dependencies among different activities.
4. Establish most likely estimates for the time durations necessary to complete the activities.

ID	Task Name	Start Date	Finish Date	Duration
1	Feasibility Study	11/1/21	13/1/21	2d
2	Requirement Gathering	15/1/21	20/1/21	5d
3	Analysis	21/1/21	28/1/21	1w
4	Design	2/2/21	16/2/21	2w
5	Coding	17/2/21	17/3/21	1m
6	Testing	18/3/21	23/3/21	5d

Table 2.3 Project Scheduling Chart

### **3. SYSTEM REQUIREMENT STUDY**

#### **3.1 Current System Study -**

There already exists some systems with similar functionalities.

##### **3.1.1 Amazon -**

Amazon.com, Inc. is an American multinational technology company based in Washington, which focuses on e-commerce, cloud computing, digital streaming, and artificial intelligence for selling various products.

##### **3.1.2 Flipkart -**

Flipkart is an e-commerce company. The company initially focused on online product sales before expanding into other product categories such as consumer electronics, fashion, home essentials, groceries, and lifestyle products.

#### **3.2 Weakness/ Problem of Current System -**

We saw that there are some excellent platforms, some provided e-commerce facilities, some limited stuff to buy.

We will provide various categories of products internationally available and great user interface for the system.

#### **3.3 Types of User(for our platform) -**

Following types of user will use the system:

- Admin / Owner -**

Admin can add / remove products, update price, coupon details, festival offers, etc.. He will have the administrative rights for the system.

- End user / Buyer -**

End user can view the products and its details, price, etc.. For buying any product, the user will have to login and fill in the necessary details like address to deliver the product and mode of payment and hence the order will be confirmed.

### **3.4 Minimum Hardware / Software requirement -**

Admin will need good internet connection and man-power, database connections, etc.. The end user will need only a computer with internet connection and a decent browser.

### **3.5 Constraints**

#### **3.5.1 Hardware Constraints -**

The first version of our application might work only on computers / laptops.

#### **3.5.2 Hardware Constraints -**

There are no other systems that use this application as an interface.

**3.5.3 Reliability Constraints** - The application demands much reliability and it is fully assured that the particular information about the user is secured and the flow is maintained and accessed according to the rights.

### **3.6 Assumptions & Dependencies**

1. Server is running smoothly.
2. Users have sufficient privileges to access the internet.
3. Database updates are giving expected and accurate results.

## **4. SYSTEM ANALYSIS**

### **4.1 REQUIREMENTS OF NEW SYSTEM(SRS)**

#### **4.1.1 Functional Requirements**

##### **4.1.1.1 User Requirements -**

1. Search / View Products by category / brand name
2. Order Product - The user can order products after redeeming the coupons / discounts.
3. Track the booked product - Once the product is booked, the user can track the progress.
4. Cancel Order - The user can cancel the order if it is not shipped yet.
5. Previous Order List - User can see previously ordered products from the history.
6. Cart Management - Users can create carts, add / remove items to cart.
7. Wishlist Management - Users can create wishlist, add / remove items to wishlist.
8. Feedback - Users can give ratings and add comments on their experience with the system and the products.

##### **4.1.1.2 Admin Requirements -**

1. Add / Remove / Update Products - Admin should be able to add new products, remove existing ones in case of any problem and update various details and prices.
2. Order Management - Admin can view / filter / reject any order.
3. Discounts / Coupon - Festival offers have discounts and first time users have coupons. So, the admin can manage from his side.
4. Feedback Management - Admin can view / remove feedback from the list.
5. Payment Consistency - If there is loss in internet connectivity while payment from users side then admin should be able to undo the entire process.

#### **4.1.1.1 User Requirements (IN DETAIL)**

##### **R1: Register>Email Authentication)**

**Description:**

Once the user types the URL of the portal in the browser, the home page is displayed in it. Once the home page is displayed in the browser the user can choose the register option from the page for creating a new member buyer profile. Upon clicking on the ‘Register’ option, the user will be directed to the sign up for a new account page where he can enter the email. After entering the email one should click on the “register” button, now if the given email id will be valid, a sign-up-link will be sent on the entered mail by the user. On clicking the sign-up-link, the user will be redirected to the Registration-Complete page where they will be asked to set Password for their account.

**R1.1: Click on the “Register” option State:**

The home page is displayed in the browser.

**Pre-condition:** For registration there is No precondition.

**Input:** “Register” option clicked.

**Output:** For New member registration email validation will be displayed, where users will have to provide valid email for account creation.

**R1.2: Click on the “Register” button State:**

After entering the email one should click on the “register” button.

**Input:** The user email

**Output:** If email is valid a message for success will be prompted

**Processing:** Validification of user entered email.

**Post-condition:** And the user will receive a sign-up-link on the provided email.

##### **R2: Login**

**Description:**

Once the user types the URL of the portal in the browser, the login page is displayed in it. The user can login through this page by entering the user id and password. The user that logins can be admin, buyers. This page would display an error message if the incorrect user id or password is entered. Upon successful login the user will be directed to the home page of the application.

**R2.1 Login with Email/Password:** The login page is opened in the browser.

**Pre-condition:** Can login only if the user is registered.

**Input:** “User id” and “Password”.

**Output:** Depending on the type of user the home page is displayed on successful login.

### R2.2 Login with Google

**Description:** If the user wants to login directly with google then he has to click the ‘Login with Google’ button and he will be redirected to the Google window and he has to select the account from which he wants to login.

**Pre-Condition:** User must be registered.

**Input:** Enter email address.

**Processing:** Checks whether user data is valid and already registered in the database.

**Output:** Depending on the type of user the home page is displayed on successful login.

### R2.3 Forgot Password

**Description:** If the user forgets the password, it can be retrieved by the ‘Forgot Password’ button. A click on this button will ask the user his email and a verification mail will be sent there. From there he can enter his new password.

**Pre-Condition:** User must be registered.

**Input:** Enter email address.

**Processing:** Checks whether user data is valid and already registered in the database.

**Output:** An email for password recovery is sent to the user.

**Post-Condition:** User is given option to reset password.

## R3: View Product

**Description:** If a user wants to see the details of any product then the ‘View Product’ button is to be clicked.

**Pre-Condition:** User should be on the home page.

**Input:** Click the ‘View Product’ button.

**Output:** An email for password recovery is sent to the user.

**Post-Condition:** User is given option to reset password.

## R4: Add to Cart

### Description:

Add to cart is a way to temporarily add items by adding them to your cart which will keep track of your items you want to buy on our website.

**Pre-Condition:** User should be on the single product/home page.

**Input:** Click the ‘Add to Cart’ button.

**Output:** A side drawer will appear on the right side and items will be added to cart.

**Post-Condition:** User can see the added product in the cart.

#### R5: Rating

**Description:**

Users can give ratings to any product. Ratings of all the users will be averaged and will be shown on the website.

**Pre-Condition:** User should be logged in.

**Input:** Click the 'Leave Rating' button.

**Output:** Average of all the ratings of all the users will be shown on the website.

**Post-Condition:** Users can continue shopping.

#### R6: Wishlist

**Description:**

It allows our users to create a list of items which they wish to buy in future but not at that particular time of moment.

**Pre-Condition:** User should be logged in.

**Input:** Click the 'Add to Wishlist' button.

**Output:** The item is added to the wishlist.

**Post-Condition:** Users are redirected to the wishlist page.

#### R7: Coupon

**Description:**

Admin will create a coupon and its start and end date with a particular discount percentage. Users can use this to buy single or multiple products.

**Pre-Condition:** User should be logged in.

**Input:** Click the 'Apply' button.

**Output:** Discount of particular percentage is given on the total amount.

**Post-Condition:** Users are kept on the same page.

#### R8: Place Order

**Description:**

For purchasing items in cart click the 'Proceed to Checkout' button.

**Pre-Condition:** User should be on the Cart page.

**Input:** Click the ‘Proceed to Checkout’/‘Cash On Delivery’ button.

**Processing:** Enter delivery address and coupon if any. Click the ‘Place order’ button.

**Output:** User is directed to Payment/user history page as per payment mode.

**Post-Condition:** Users are directed to the checkout page which shows delivery address and order summary.

### R9: Payment(Online)

#### Description:

Online payment will ask for user’s bank account details and once the payment is done, the order is confirmed.

**Pre-Condition:** User should have already placed the order.

**Input:** Enter the bank account details as per asked.

**Output:** Click the ‘Pay’ button.

**Post-Condition:** Users are directed to the user history page where the history of previously bought items is shown.

### R9: Payment(COD)

#### Description:

COD payment will allow the user to pay at the time of delivery.

**Pre-Condition:** User should have already placed the order.

**Input:** Click the ‘Place Order’ button.

**Output:** COD is activated.

**Post-Condition:** Users are directed to the user history page where the history of previously bought items is shown.

### R10: Invoice Download

#### Description:

This feature helps users download their hardcopy of their invoice.

**Pre-Condition:** Users should be on the user history page where the history of previously bought items is shown.

**Input:** Click the ‘Download PDF’ button.

**Output:** Hardcopy of the bill is downloaded in the downloads folder.

**Post-Condition:** Users are kept on the same user history page.

### R11: Logout

**Description:**

To leave the website, logout/checkou from the website.

**Pre-Condition:** Users should be logged in.

**Input:** Click the 'Logout' button.

**Output:** User is logged out and redirected to the login page.

**Post-Condition:** None.

### 4.1.1.2 Admin Requirements (IN DETAIL)

#### R1: Login

**Description:**

It allows only authorized people to access the application. When the admin logs in to the system, the admin has to enter login details in the specific fields. The admin clicks on the login button and if the username and password are validated, then the admin is given access to the application and admin dashboard is displayed.

**Pre-Condition:** admin must be registered.

**Input:** Enter your user name or email address and password.

**Processing :** Checks whether the entered email address and password is correct or not.

**Output:** If valid, then admin is given access to the application and admin dashboard is displayed. Else an error message is displayed indicating some incorrect input.

**Post-Condition :** admin is redirected to Dashboard.

#### R2: Manage Product Details

**Description:** Admin can see all the product details and he can add new product details and edit the existing product details.

##### R2.1 Add Product Details

**Description:** When the admin click on the add new Products button he can add the products.

**Pre-Condition:** Admin has to click on add new button.

**Input:** Enter product name, price, total stock, description, select category and subcategory, upload multiple images and status of shipping.

**Processing:** Checks whether all the entered fields are correct or not according to the database.

**Output:** Product added message will be displayed.

**Post-Condition:** Product details have to be added to the database.

**R2.2: Edit Product Details**

**Description:** If the admin wants to edit the product details he has to click on the edit for that product and then all details of the product will be changed.

**Pre-Condition:** Admin has to click on edit Product.

**Input:** Change the required information of Product.

**Processing:** Checks if the edited information is valid or not.

**Output:** Edited information will be displayed.

**Post-Condition:** Edited information has to update on the database.

**R3: Manage Orders**

**Description:** Admin can search the orders as per the requirements and he can view and edit the orders.

**R3.1 Display Orders in Admin Dashboard**

**Description:** Admin provide the search criteria for displaying the order and clicks on the search button and order details will be displayed.

**Input:** Admin has to select the date of orders, amount range, order status, payment mode.

**Processing:** when the admin clicks on the search button as per the provided information data will be fetched from the database.

**Output:** Required order details will be displayed.

**R3.2 Edit Order**

**Description:** Admin can set the status of product, whether product is processed, not processed, shipped or not shipped. If the admin wants to edit the order details, he/she has to click on the edit button for that order and then information of the order will be changed.

**Pre-Condition:** Admin has to click on edit.

**Input:** Change the required information of order.

**Processing:** Checks if the edited information is valid or not.

**Output:** Edited information will be displayed.

**Post-Condition:** Edited information has to update on the database.

**R4: View Customers**

**Description:** Admin can view the detailed information of registered customers on firebase console.

**Input:** Admin has to go to Firebase console and head over to authentication and then users.

**Output:** Detailed information of customers will be displayed.

**R5: Authorization**

**Description:** The entered id and password are checked by Firebase and if it is valid then the user is allowed to login.

**Input:** The id and password which are the inputs of the login module are passed to the database.

**Output:** positive reply if password is correct.

**R6: Logout**

**Description:** After the required changes admin can log out.

**Pre-Condition:** Admin must be logged in.

**Input:** Click on logout.

**Output:** Admin is redirected to Log in Page.

**Post-Condition:** Admin's changes are reflected in the database and the session is expired.

**4.1.2 Non-Functional Requirements -****4.1.2.1 Security -**

It is a primary requirement of any system. It must maintain/process data in a secure way so that unauthorized entities do not have illegal access to the system.

**4.1.2.2 Performance -**

It is very important to know how well the system performs certain functions under specific conditions. Examples are speed of response, throughput, execution time and storage capacity. This platform should be designed in such a way that its performance is smooth for users.

**4.1.2.3 Scalability -**

It is a property of a system that describes the ability to appropriately handle increasing (and decreasing) workloads. Scalability competes with and complements other non-functional requirements such as availability, reliability and performance. This platform must be designed in such a way that it is able to scale well with increasing/decreasing accesses, users etc..

**4.1.2.4 Availability -**

System must be available to all the authorized users when it is required at any hour of the day. In any abnormal situation like loss of internet connection user's data should not be lost.

## 4.2 DATABASE DESIGN

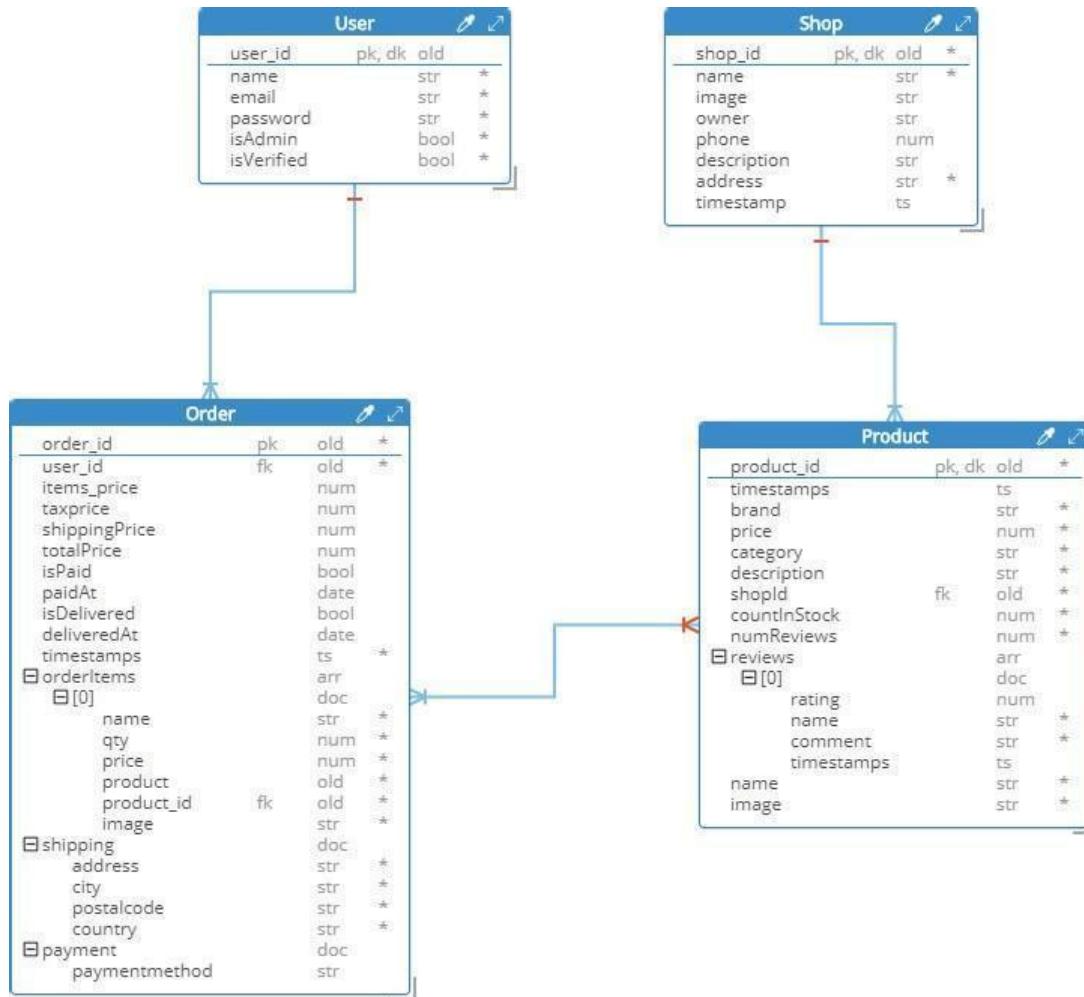


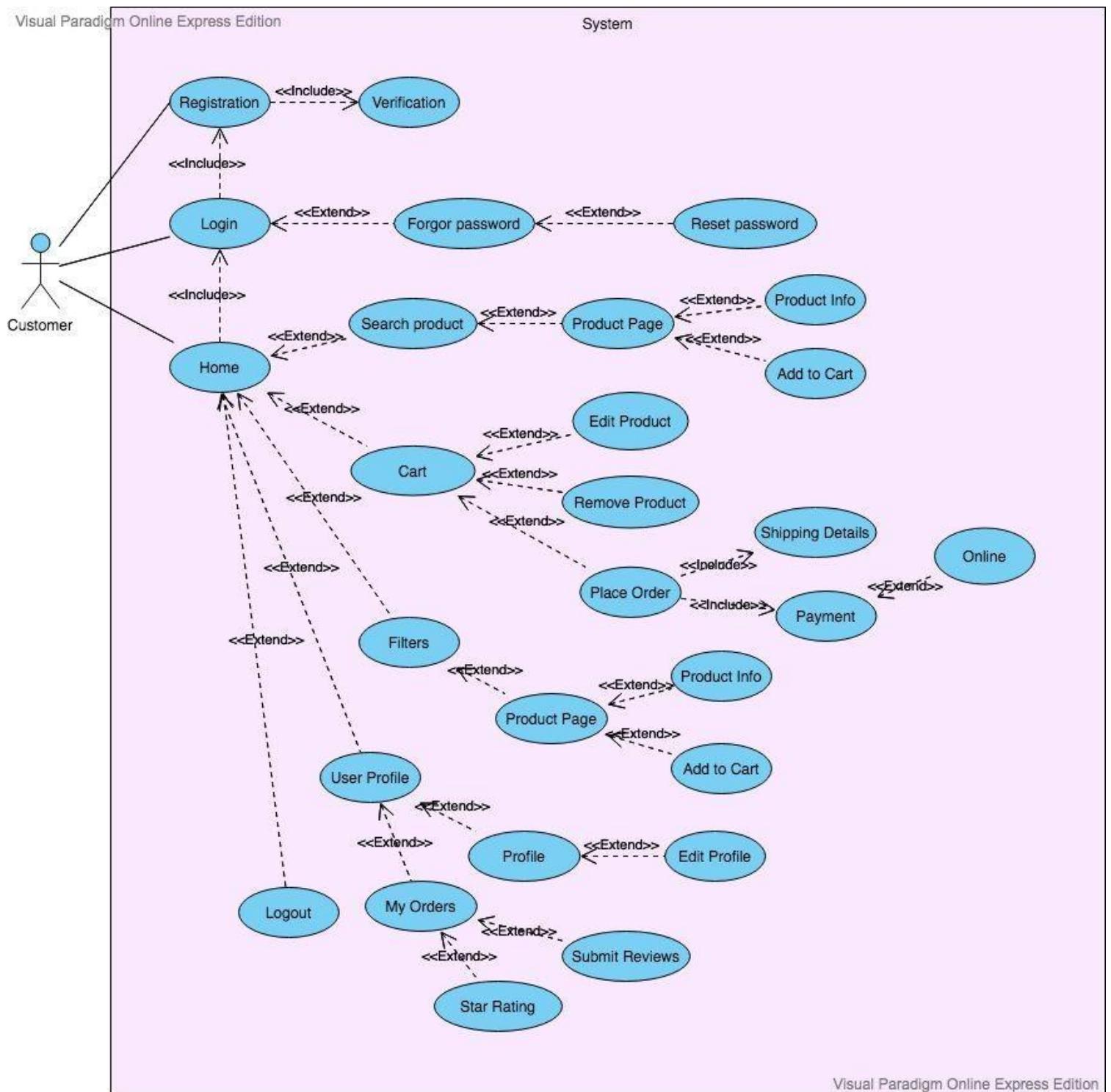
Fig 4.2 MongoDB DATABASE MODEL

## 5. SYSTEM DESIGN

### 5.1 USE CASE DIAGRAM

We created the use case diagrams using Visual Paradigm Online.

#### 5.1.1 CUSTOMER:



### 5.1.2 ADMIN:

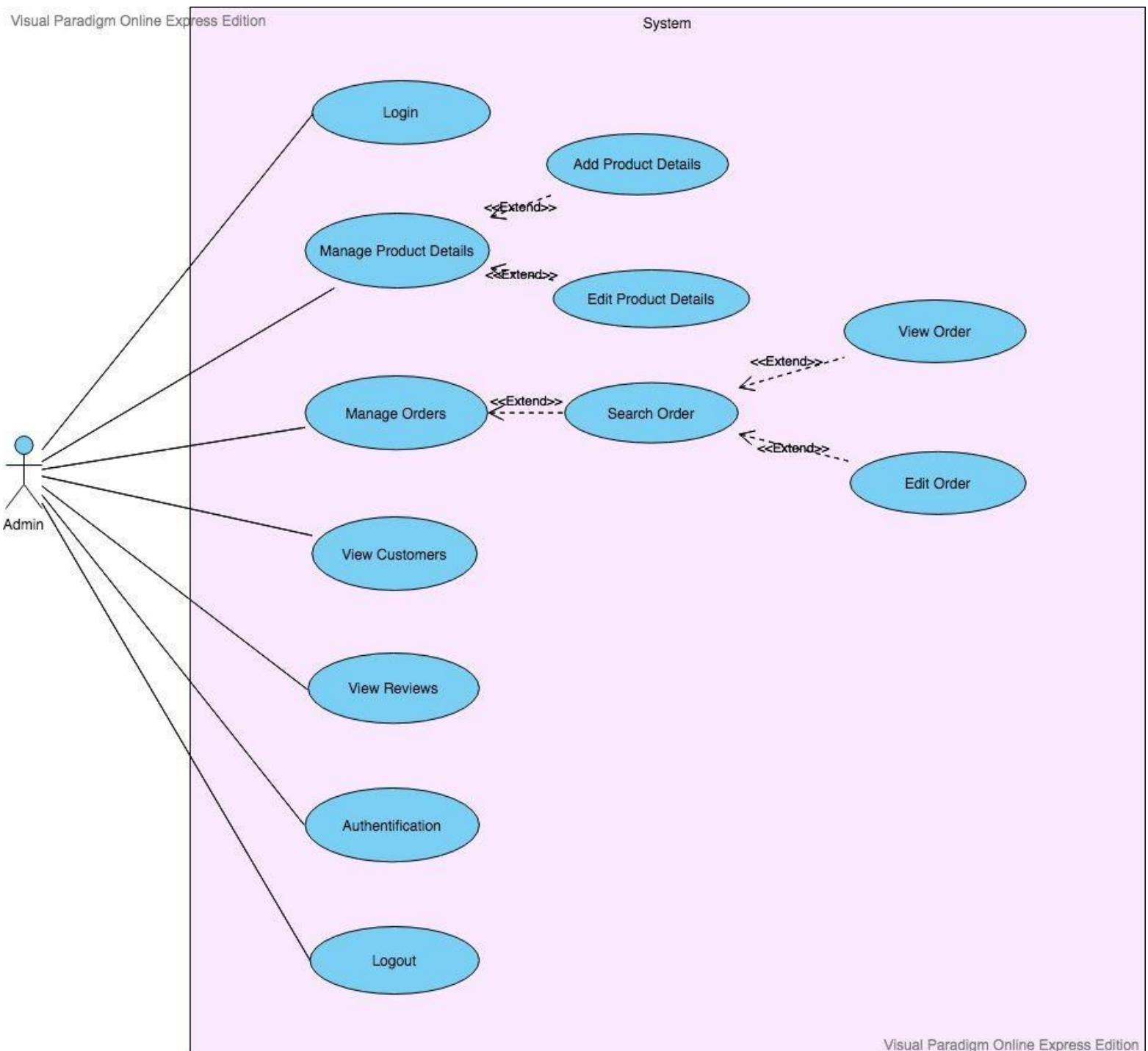


Fig 5.1 Use Case Diagram for Customer and Admin

## 5.2 CLASS DIAGRAM

We created the class diagram using Creately.

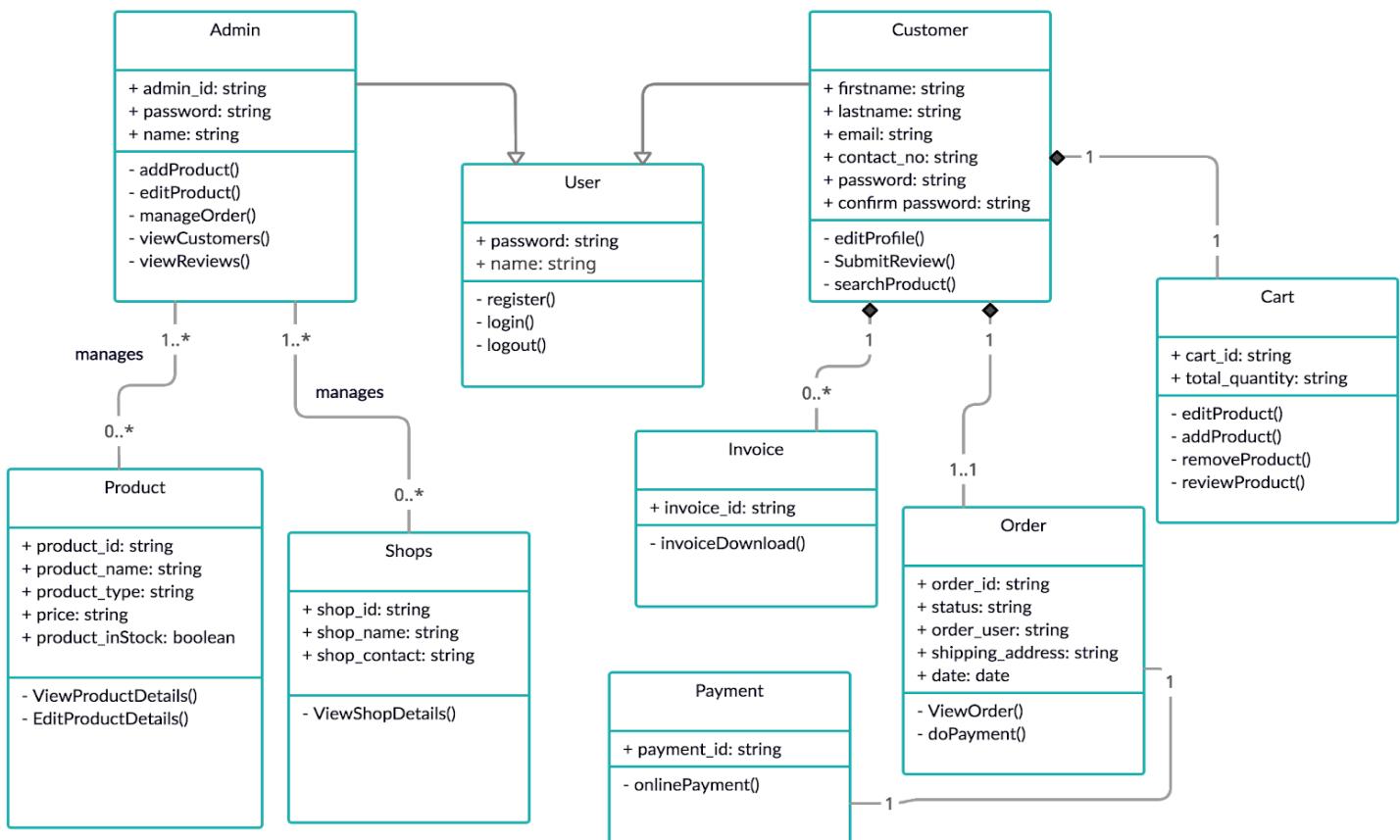


Fig 5.2 Class Diagram

## 5.3 SEQUENCE DIAGRAM

### 5.3.1 CUSTOMER SEQUENCE DIAGRAM

#### 1. Registration and Verification

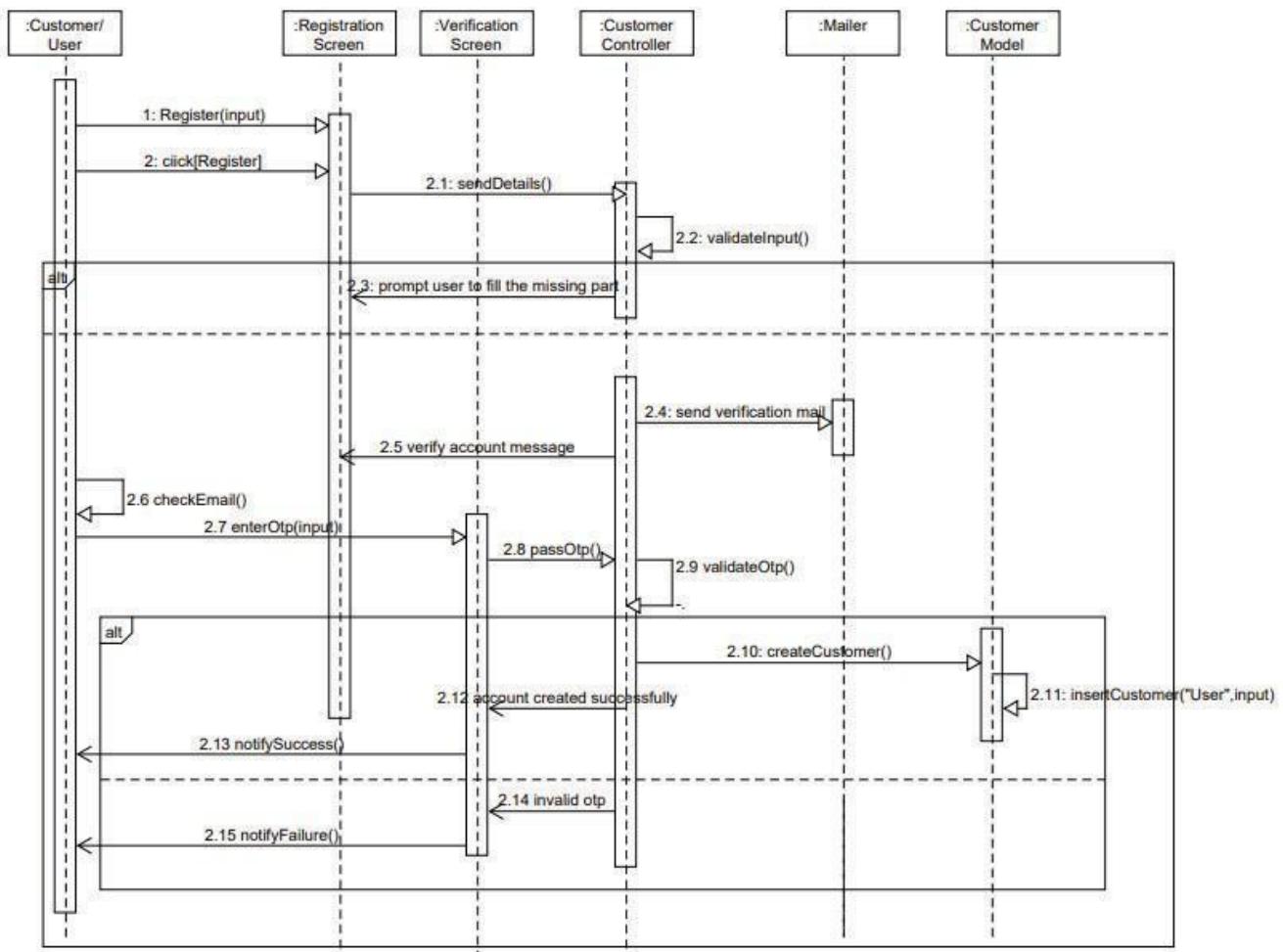


Fig 5.3.1.1 Registration and Verification

## 2. Login

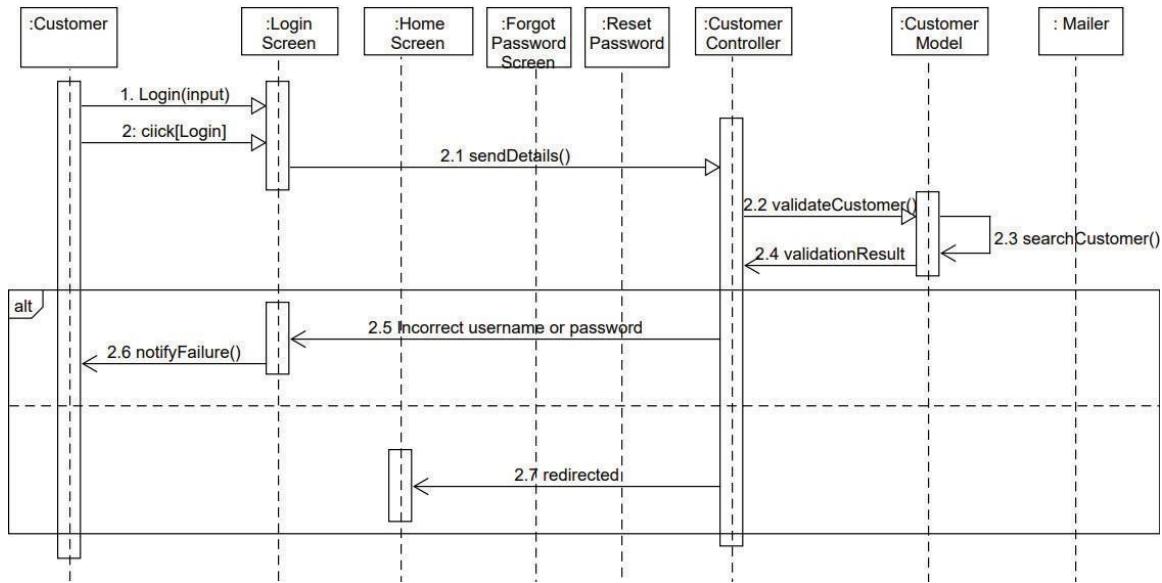


Fig 5.3.1.2 Login

## 3. Forgot Password and Reset Password

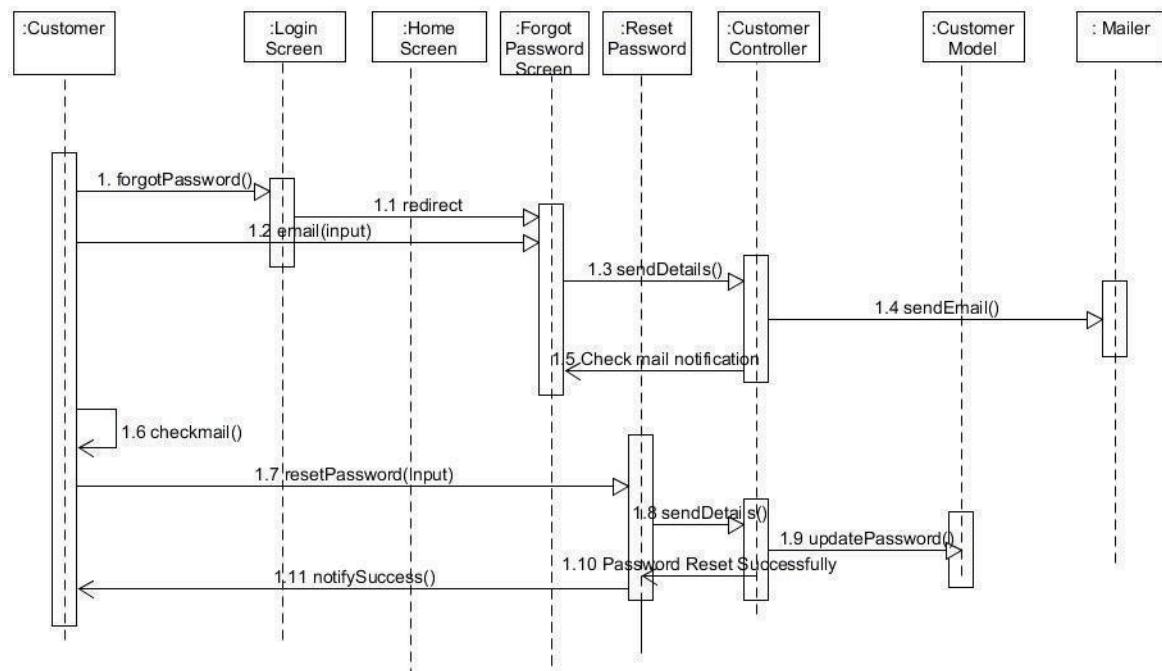


Fig 5.3.1.3 Forgot and Reset password

#### 4. Search Product

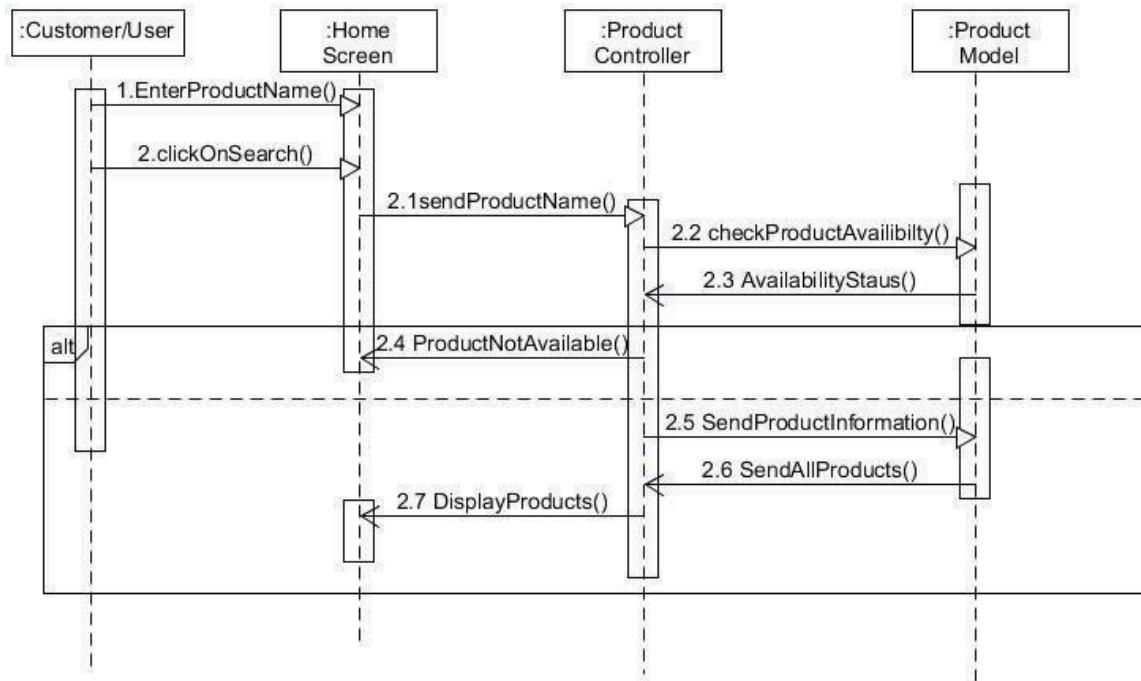


Fig 5.3.1.4 Search Product

#### 5. Add to Cart

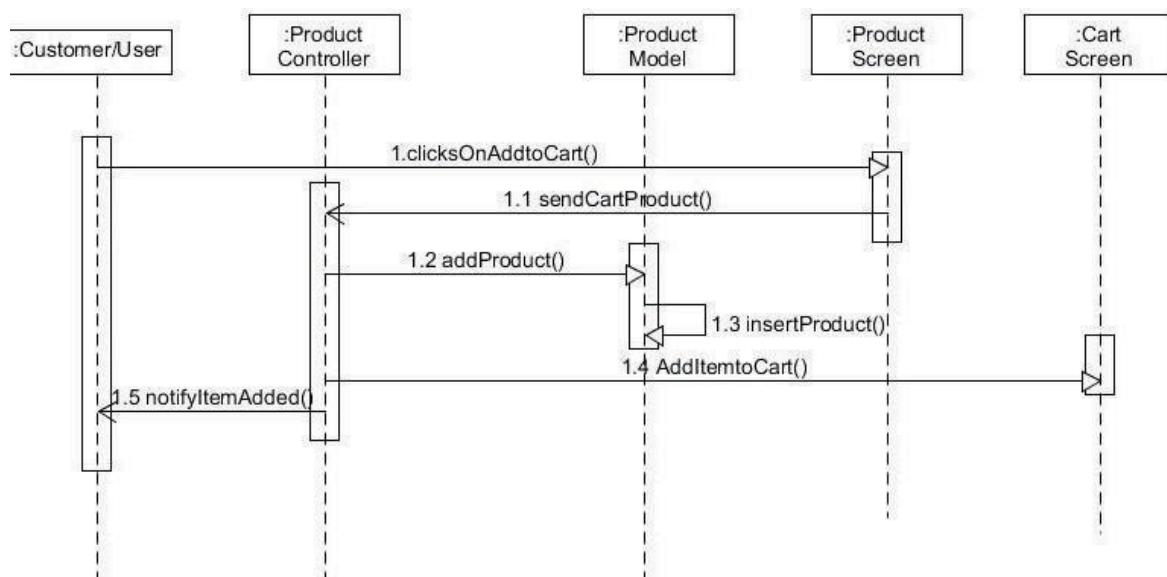


Fig 5.3.1.5 Add to Cart

## 6. Shopping Cart- Edit / Remove Product

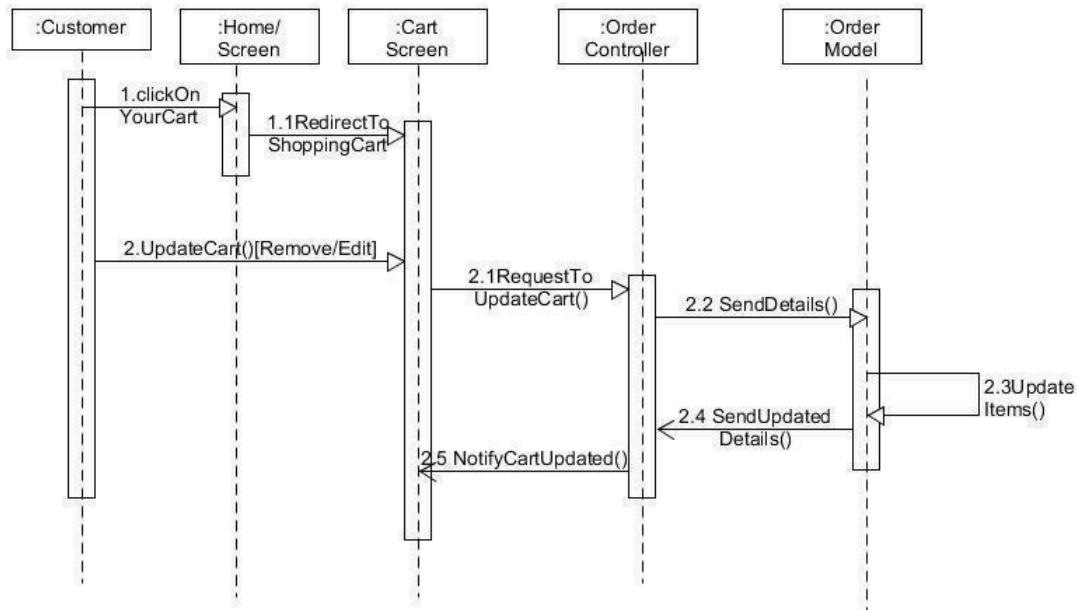


Fig 5.3.1.6 Cart -Edit/ Remove Product

## 7. Place Order

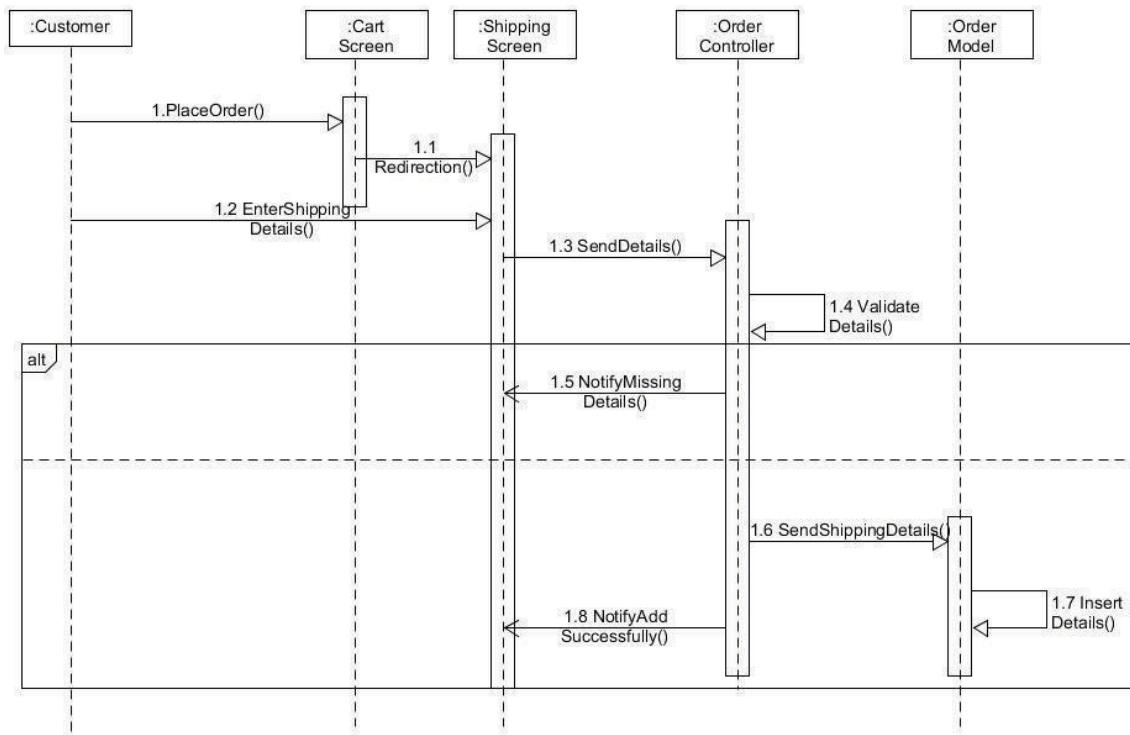


Fig 5.3.1.7 Place Order

## 8. Payment

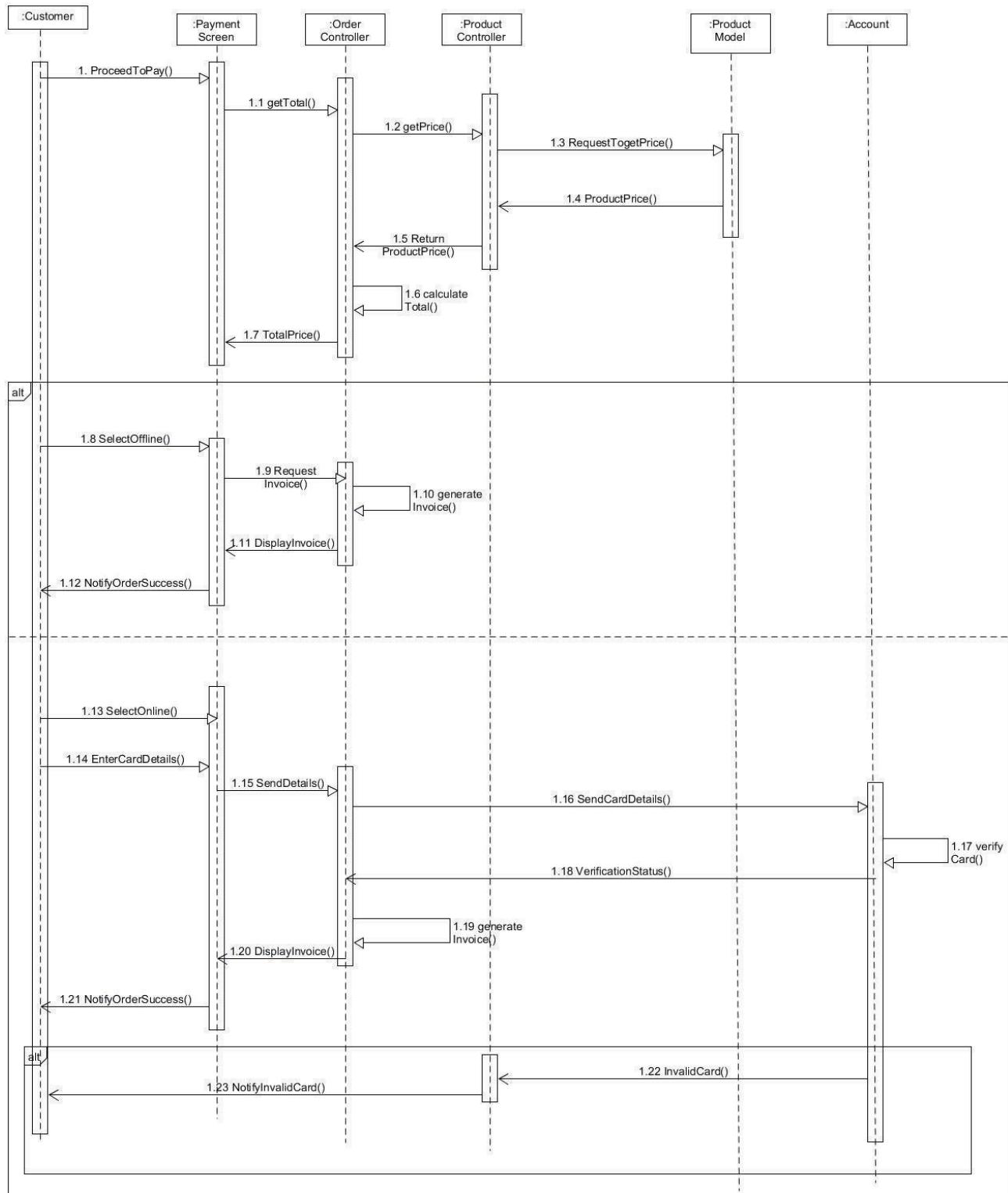


Fig 5.3.1.8 Payment

## 9. Logout

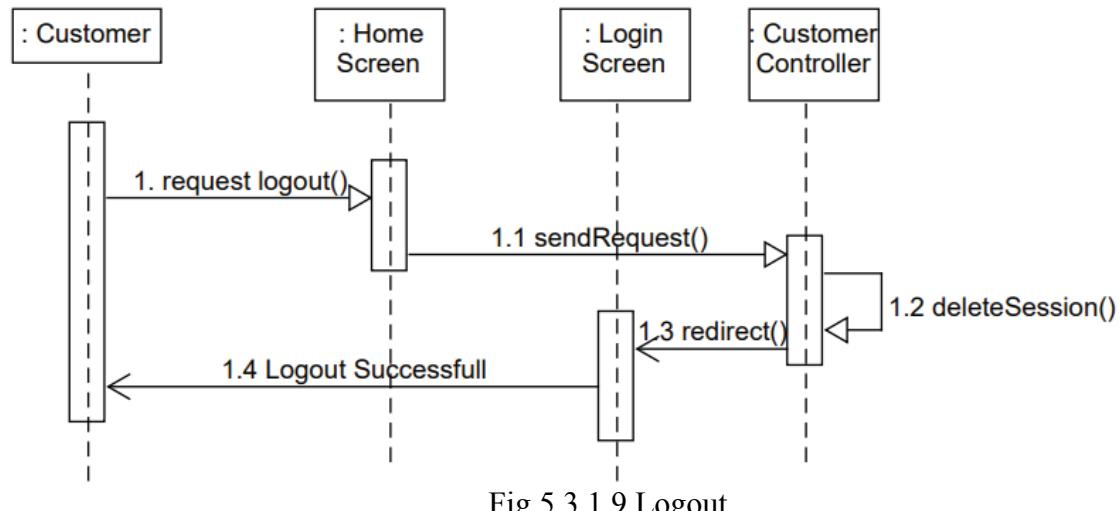


Fig 5.3.1.9 Logout

### 5.3.2 ADMIN SEQUENCE DIAGRAM

#### 1. Edit Shop Details

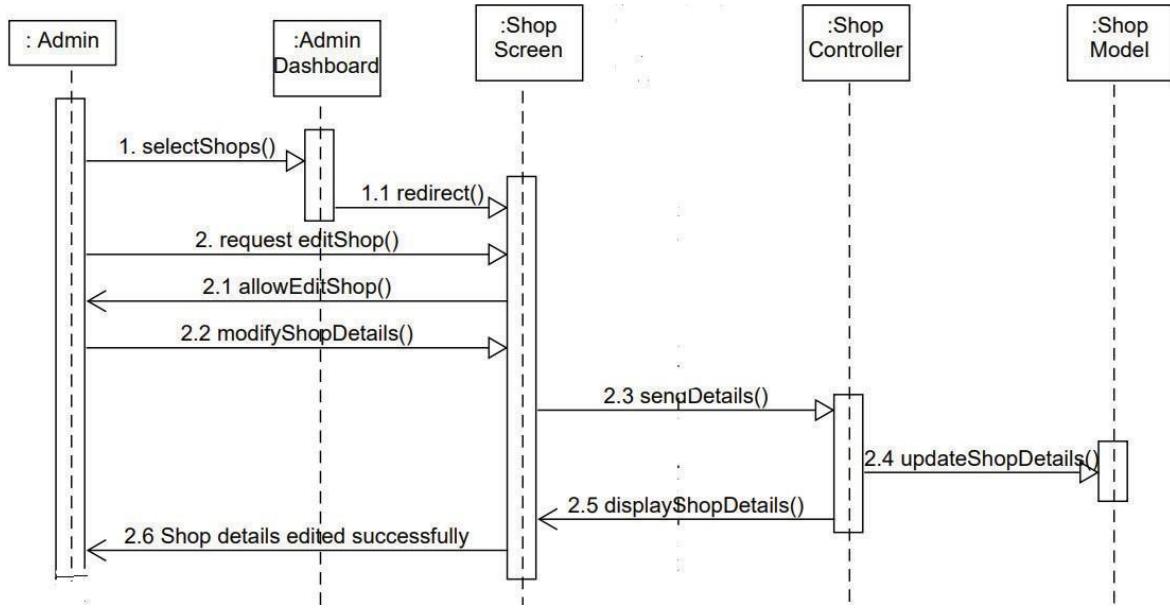


Fig 5.3.2.1 Edit Shop details

## 2. Edit Product Details

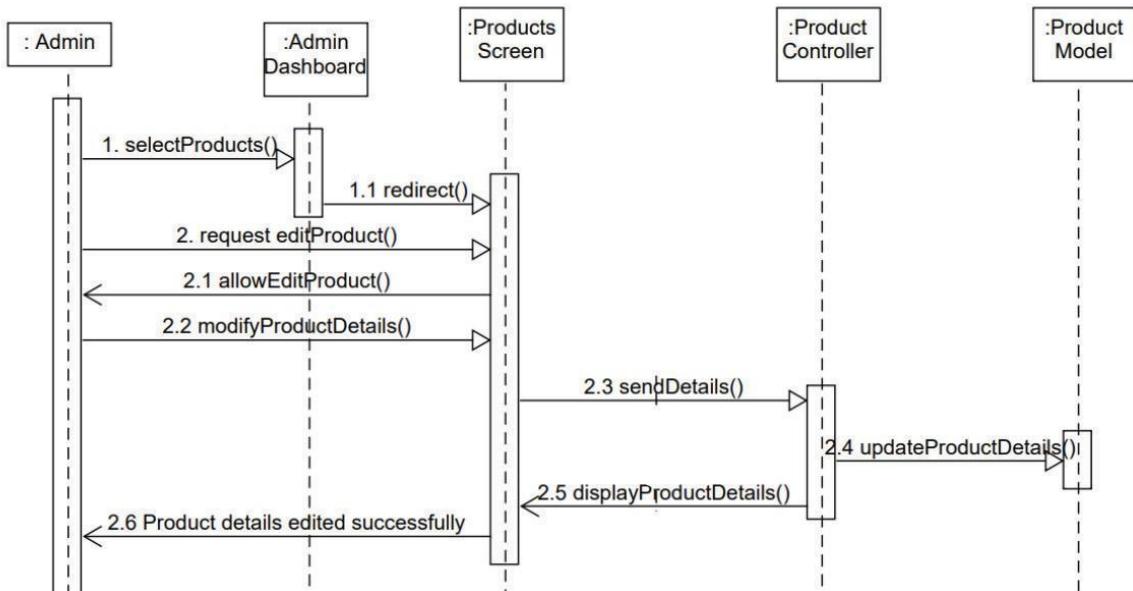


Fig 5.3.2.2 Edit Product details

## 3. Add Product Details

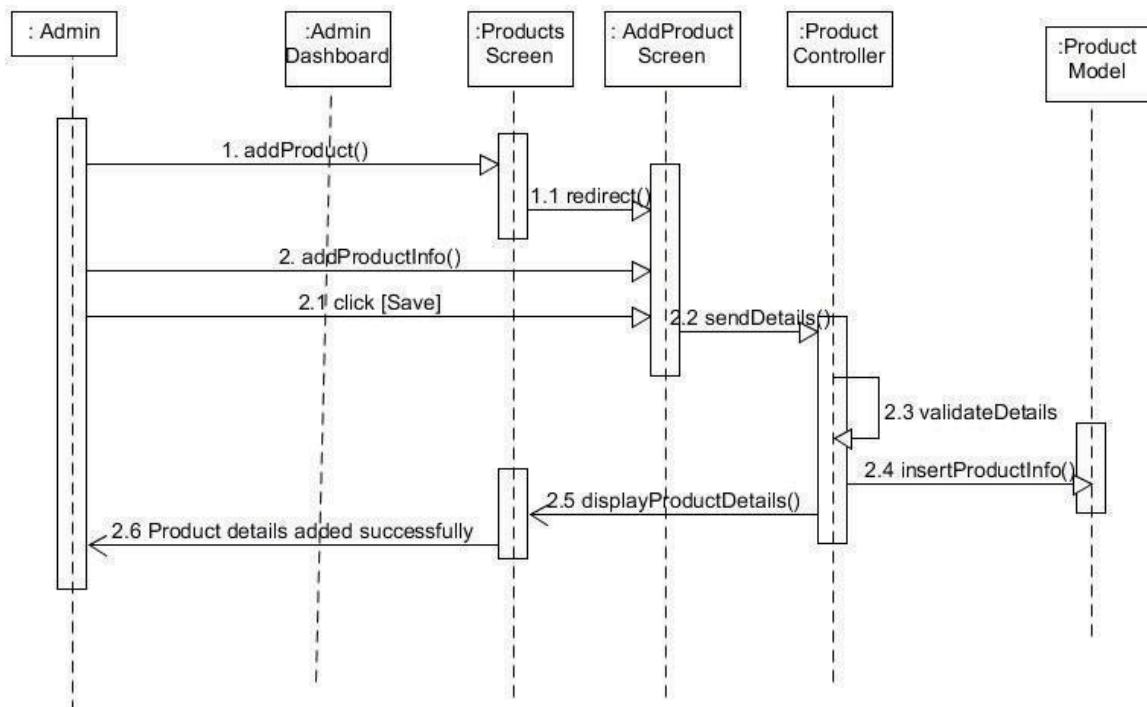


Fig 5.3.2.3 Add Product details

#### 4. Search Orders

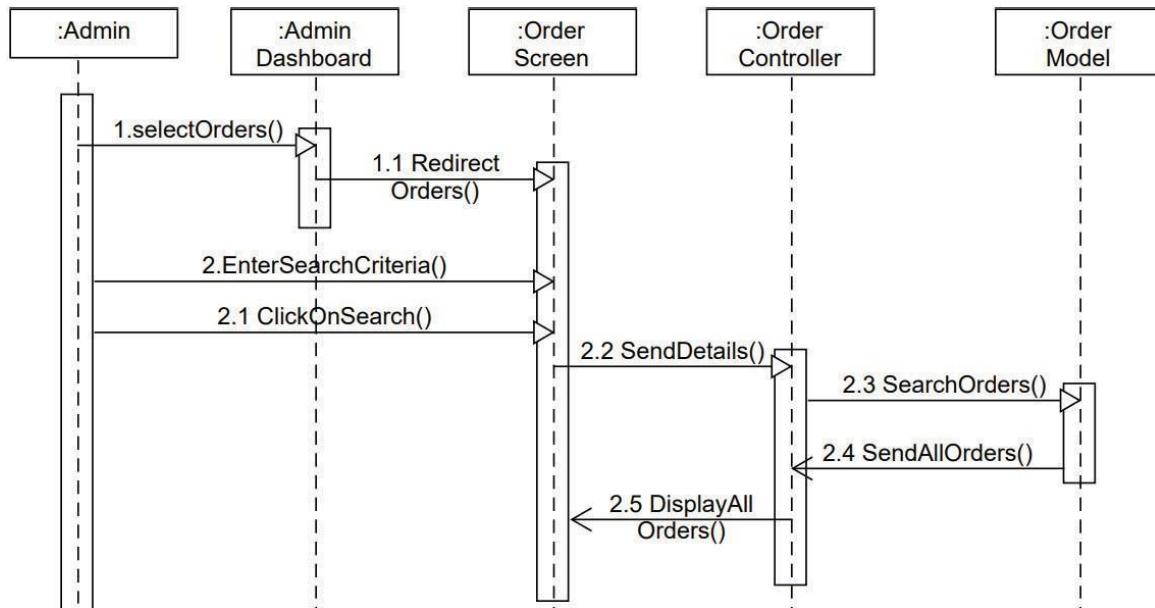


Fig 5.3.2.4 Search Orders

#### 5. Edit Order

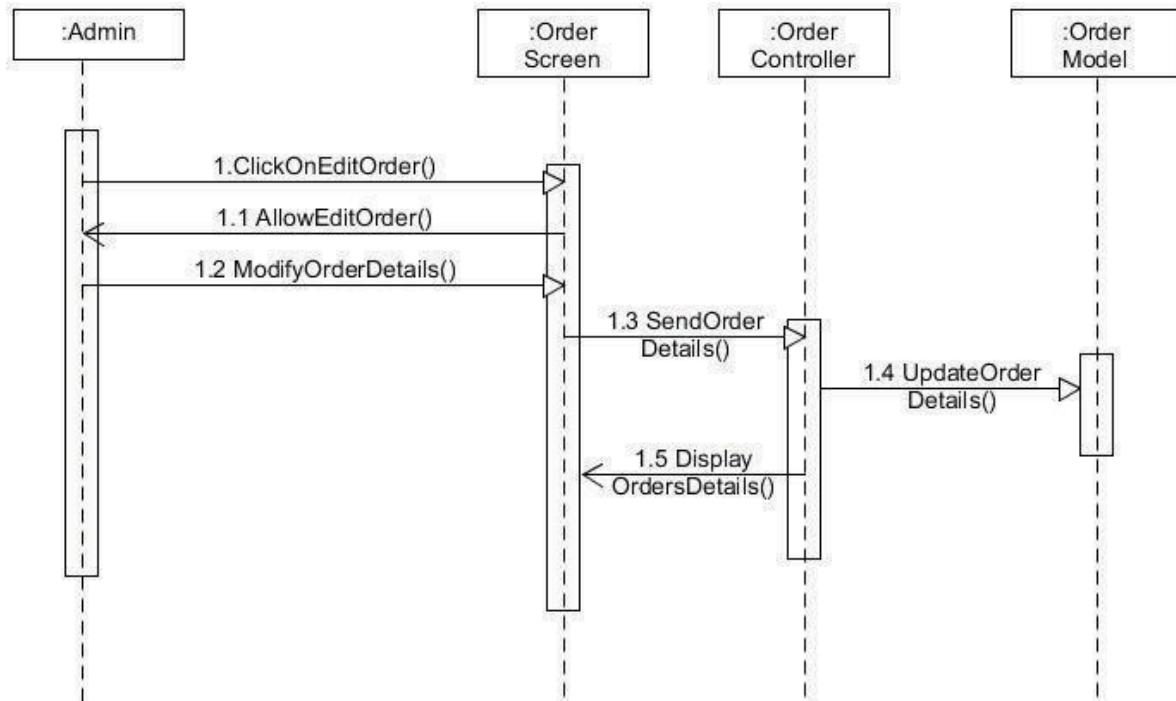


Fig 5.3.2.5 Edit Orders

## 6. View Reviews

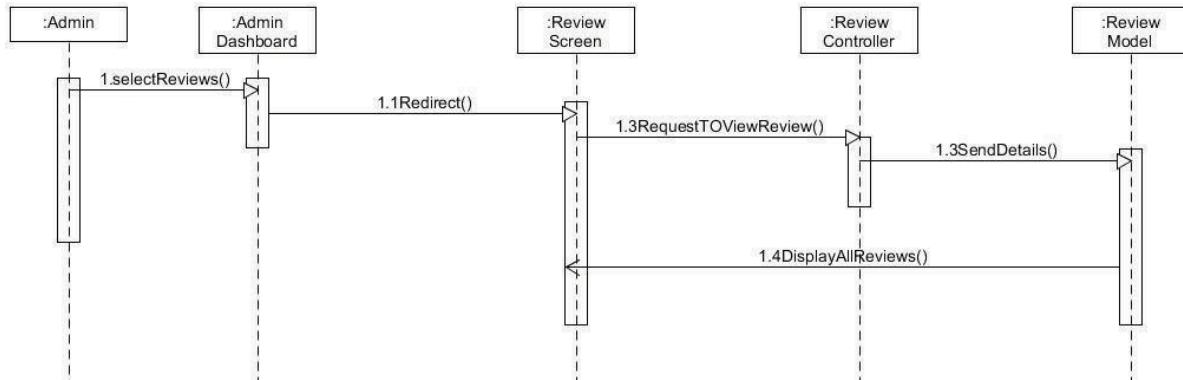


Fig 5.3.2.6 View Orders

## 7. View Customers

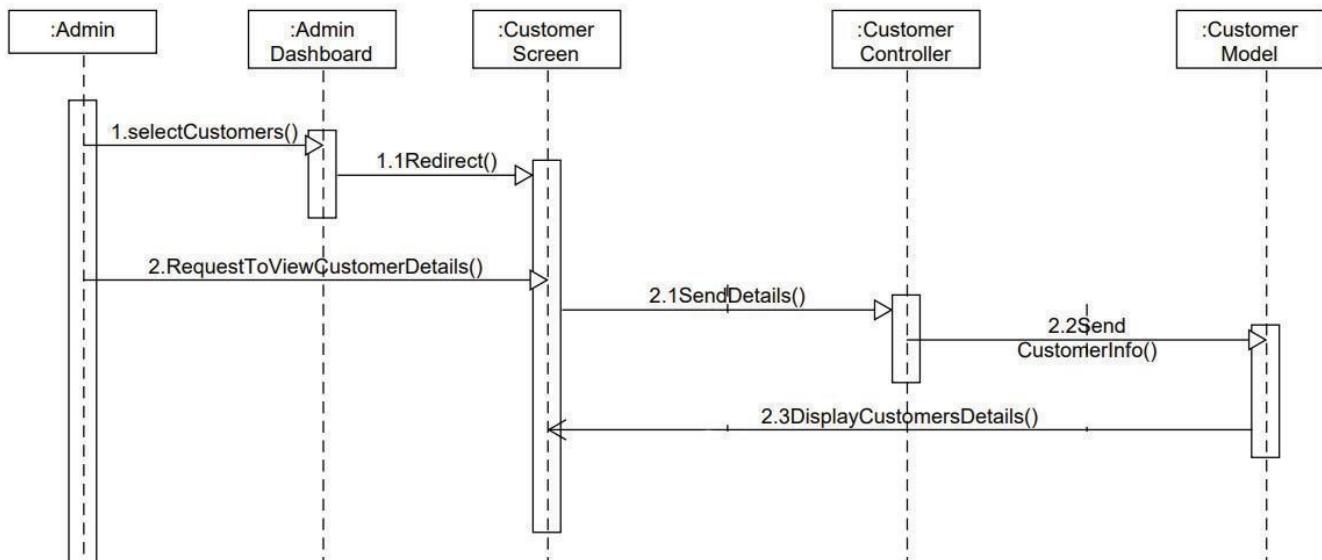


Fig 5.3.2.7 View Customers

## 5.4 COMPONENT DIAGRAM

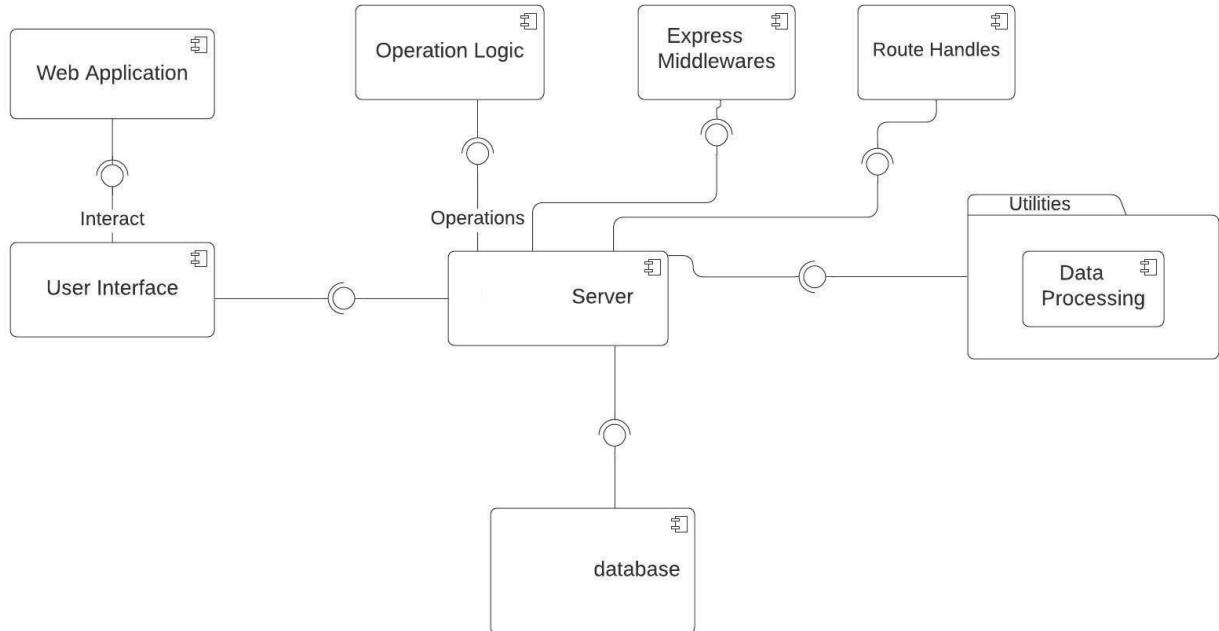


Fig 5.4 Component Diagram

## 5.5 DEPLOYMENT DIAGRAM

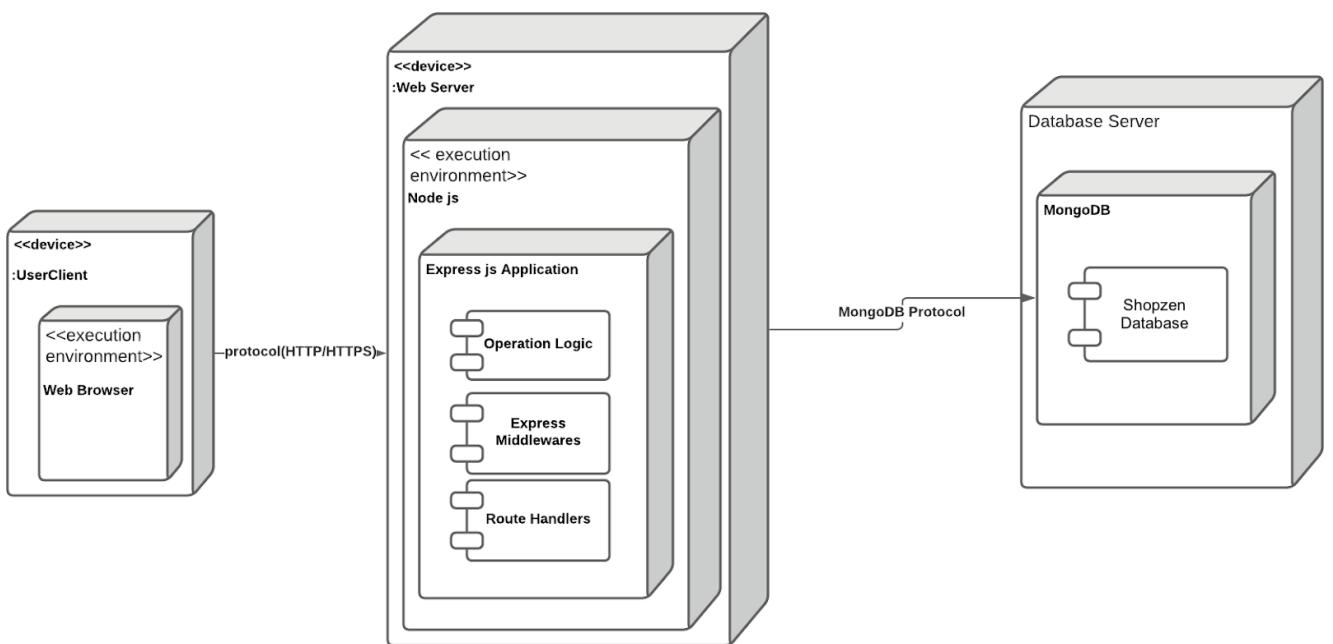


Fig 5.5 Deployment Diagram

## **6. IMPLEMENTATION PLANNING**

### **6.1 Implementation Environment (Single vs Multi User, GUI vs Non GUI)**

For implementation we have used:

1. Visual Studio Code
2. MongoDB Atlas

Our project is built using MERN stack for the backend and frontend purposes. And for the database purpose we have used online MongoDB- Atlas, it is NoSQL (aka "not only SQL"), databases are non tabular, and store data differently than relational tables, they provide flexible schemas it does not require a fixed schema and scale easily with large amounts of data and high user loads.

### **6.2 Program/Modules Specification**

Our application is based on MVC architecture so we have created different repository which provides different methods to access data. The controller is responsible for responding to the user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model, and passes it to the view which renders and update the user's view. In this way model, view and classes together form a system which separates all three of them.

### **6.3 Coding Standards**

To make the system coding easy, easy to remember and reducing the chances of errors some techniques are used at the time of coding of the application which is called coding standard. The coding standard which we adopted during the coding is explained as follows:

- Each nested block should be properly indented and spaced.
- The code should be properly commented for understanding easily. Comments regarding the statements increase the understandability of the code.
- Better to avoid use of digits in variable names.
- The names of the function should be written in camel case starting with small letters.
- The name of the function must describe the reason for using the function clearly and briefly.

## 7.0 TESTING

### 7.1 Testing Plan

The testing technique that is going to be used in the project is White box testing. In White box testing the Tester has knowledge about the internal structure of the code or the program of the software. If a third party checks our application and tests it then it is known as black box testing because internal details are hidden.

### 7.2 Testing Strategy

The development process repeats this testing subprocess a number of times for the following phases.

- a) Unit Testing.
- b) Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed.

Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

### 7.3 Testing Methods

Black Box and White Box Testing:

In **black-box** testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In **white-box** testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

Out of the 2 methods for testing, black box testing and white box testing, we would be using the white box testing as we are well aware of the internal functionalities of our application unlike in the black box testing, where we require a 3rd party to test our cases and the internal details are hidden from him.

## 7.4 Test Cases

Purpose	Input	Test Case	Expected output	Actual output	Test output
Login	Registered Username and password (correct credentials)	Validation	Redirect to Product page	Redirect to Product page	PASS
Login	Wrong credentials	Validation	Redirect to Product page	Display error	FAIL
Search Product	Search desired product	Search criteria	Display product(s) according to user's search	Display product(s) according to user's search	PASS
Search Product	Search desired product	Search criteria	Display product(s) according to user's search	Product not found	FAIL
Filter Product	Filter desired product	Filter criteria	Display product(s) according to filters applied	Display product(s) according to filters applied	PASS
Filter Product	Filter desired product	Filter criteria	Display product(s) according to filters applied	Product not found	FAIL
Add To Cart	Add product in to cart	Cart information	Display appropriate Cart according to User's choice	Display appropriate Cart according to User's choice	PASS

Add To Wishlist	Add product in to User's wishlist	Wishlist	Add product to wishlist and Display user's wishlist	Add product to wishlist and Display user's wishlist	PASS
Checkout	Address, pincode, State, city, phone number, payment method	Delivery Address information	Display Summary of Order and send order Details to server & database once user clicks to "place order"	Display Summary of Order and send order Details to server & database once user clicks to "place order"	PASS
Payment	Card details /bank-account details (correct credentials)	Validation	Redirect to "Payment History" page	Redirect to "Payment History" page	PASS
Payment	Card details /bank-account details (wrong credentials)	Validation	Redirect to "Payment History" page	Payment Failed, Display error	FAIL
Purchase History	Go to User Purchase History	User's Purchase History	Display all orders placed by customer	Display all orders placed by customer	PASS
Invoice Download	Click download pdf	Invoice	Display order summary, order date, order id.	Display order summary, order date, order id.	PASS
Rating (with Averaging)	Gives Rating to specific product	Customer's Rating (with Averaging)	Display Average Rating on that Product.	Display Average Rating on that Product.	PASS

Table 7.4 Test Cases

## **8.0 USER MANUAL**

User Manuals are manuals that enable the user of a system or application to understand the working of the system and help them to use them efficiently.

It is usually written by a technical writer, although user guides are written by programmers, productor project managers, or other technical staff, particularly in smaller companies.

Our user guides contain both a written guide and the associated images. In the case of our application, it is usual to include screenshots of how the program should look. The language use is mismatched to the intended audience.

## 1. Registration page.

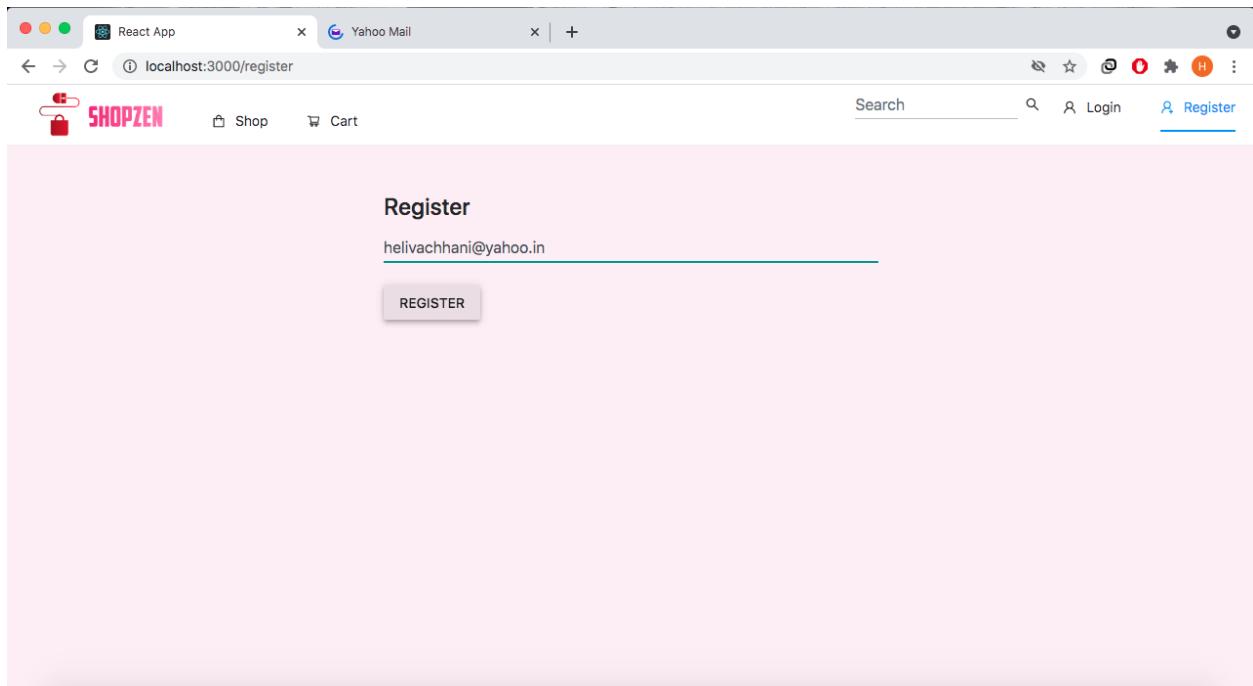


Fig 8.1 Registration Page

- **Input:** Valid email address.
- **Output:** User email verification Succeeded if valid and link sent to that email.

## 2. If the email is valid it will send a sign-up link to the registered email address.

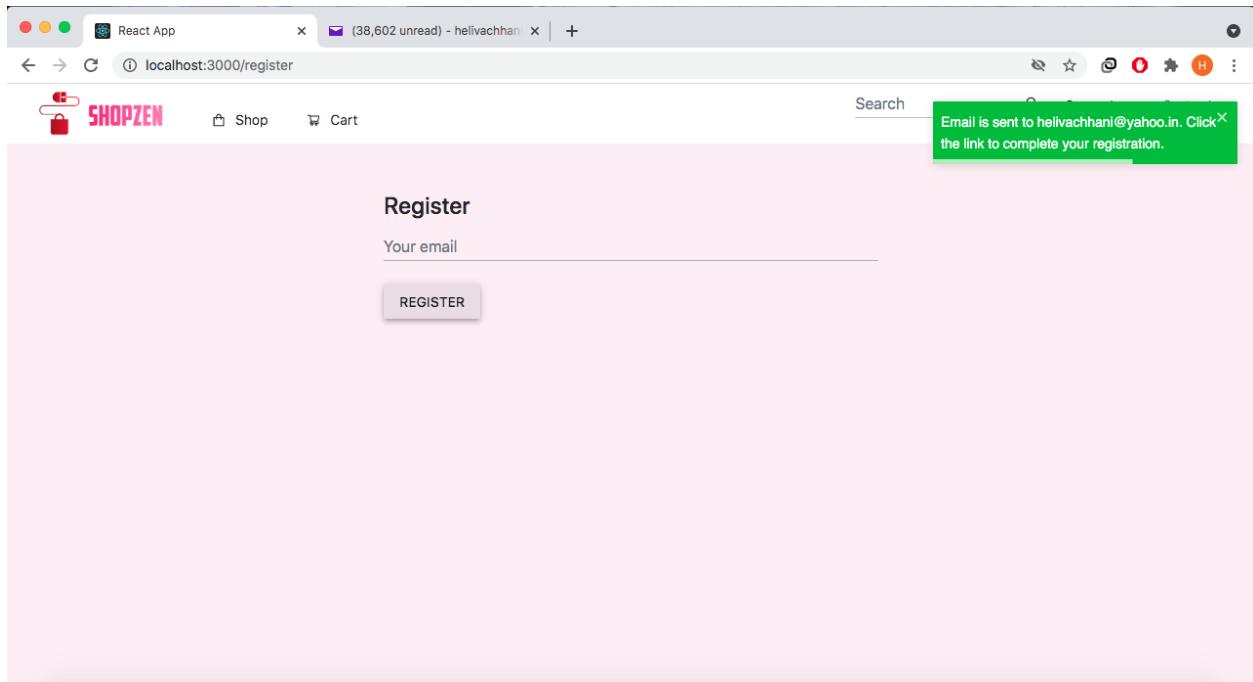


Fig 8.2 Registration Page Popup

### 3. Click on the link to proceed.

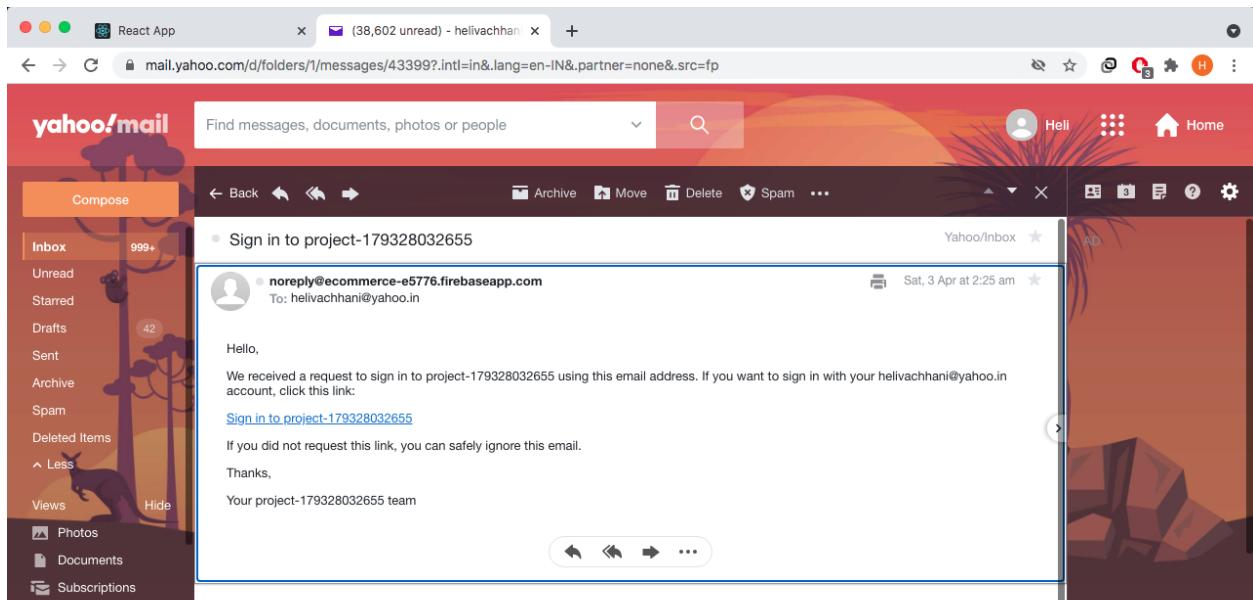


Fig 8.3 User Mail

- Input:** Click hyperlink to proceed to the completed registration page.
- Output:** User password set page.

### 4. Set password to complete registration.

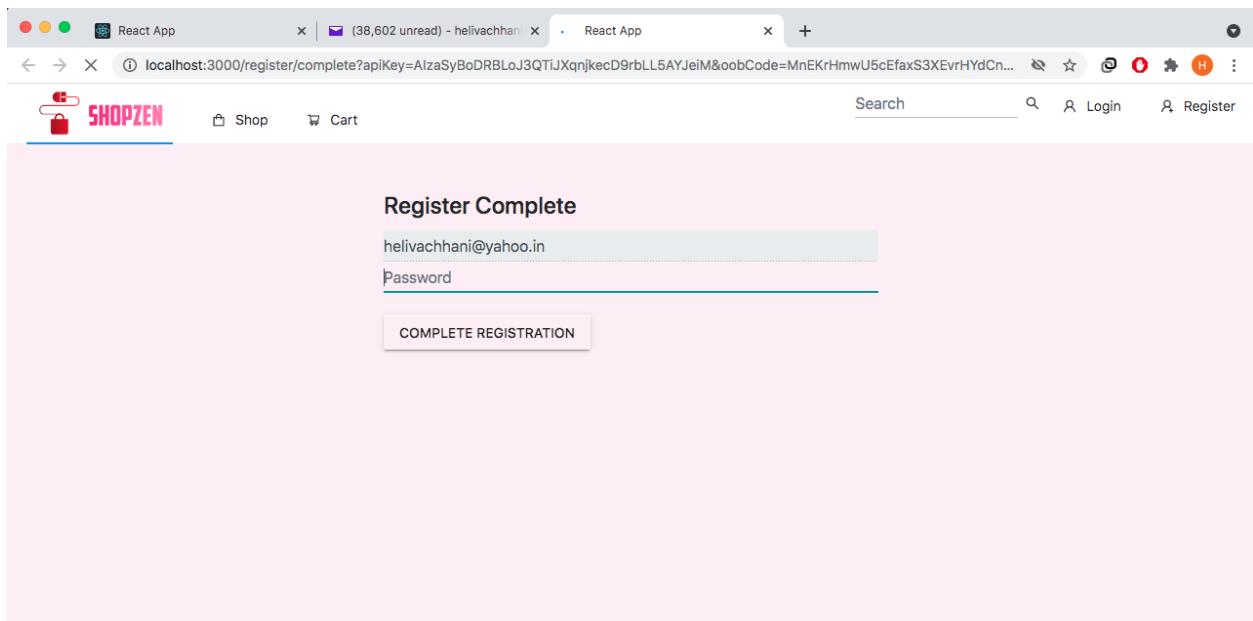


Fig 8.4 Set Password to complete registration

- Input:** Type to set ur password and click complete registration.
- Output:** User Registration Succeeded.

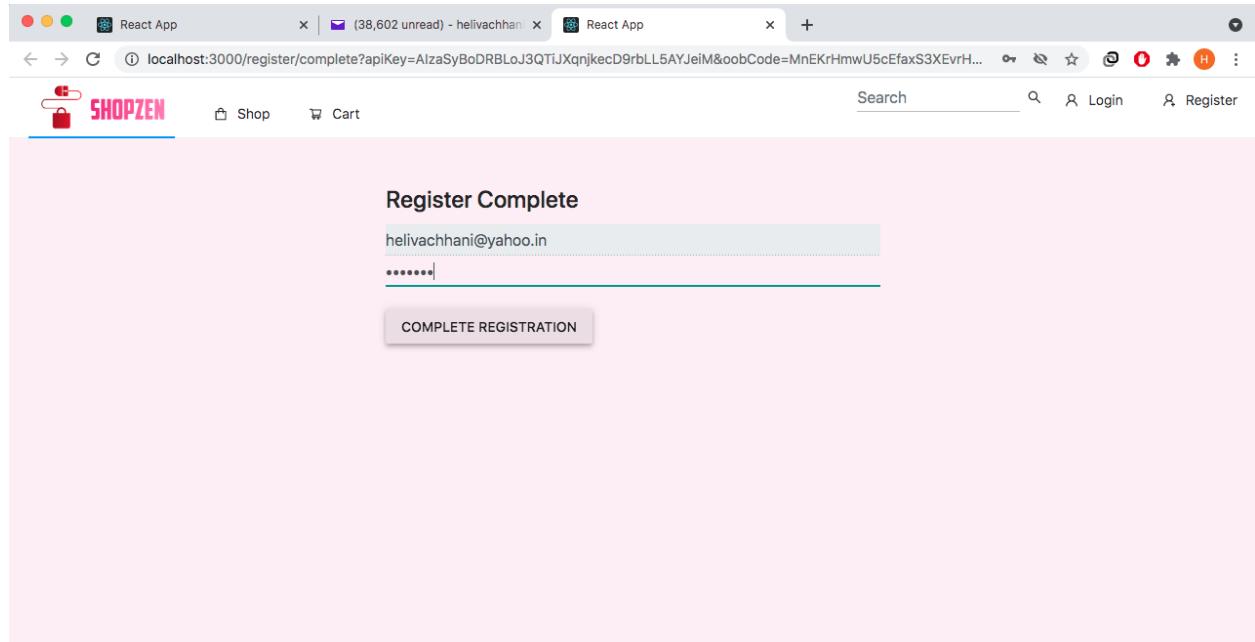
**5.**

Fig 8.5 Complete registration

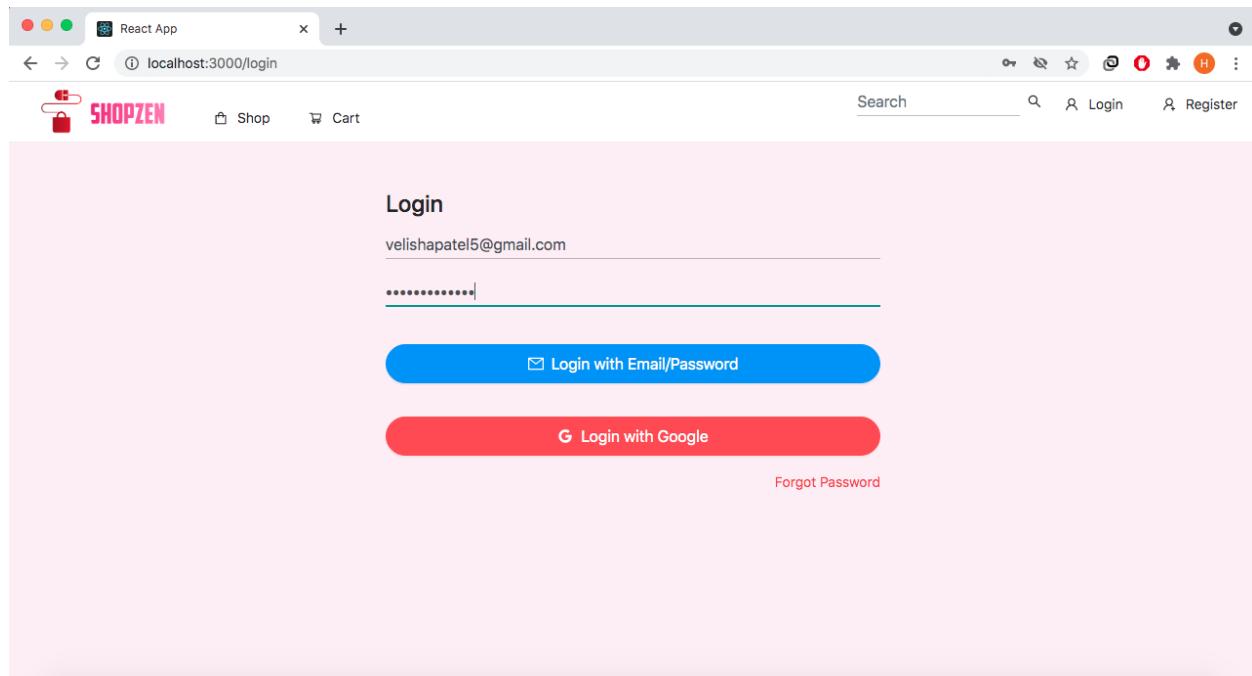
**6. Login. (email login)**

Fig 8.6 Login Page

- **Input:** Correct Credentials.
- **Output:** Login Success.

## 7. Login fails if credentials don't match.

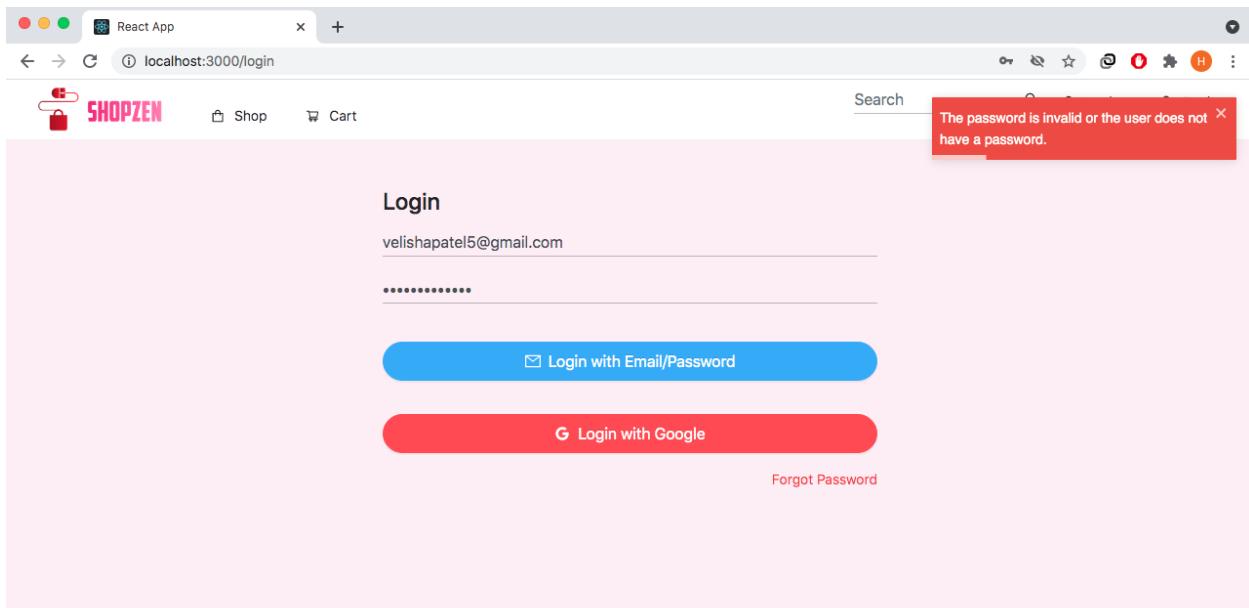


Fig 8.7 Login Failed Page

- **Input:** Incorrect Credentials
- **Output:** Login Failed.

## 8. Login. (direct login with Google)

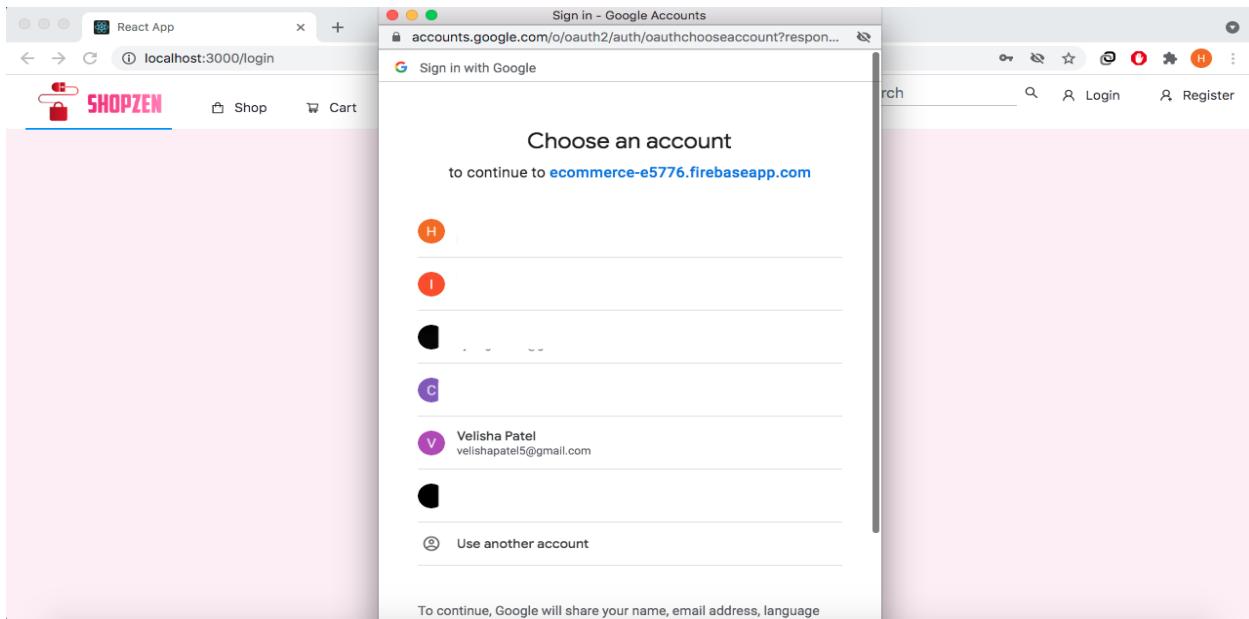


Fig 8.8 Login with Google

- **Input:** Choose account.
- **Output:** Login Success with Google.

## 9. On login success, users are directed to the Home Page.

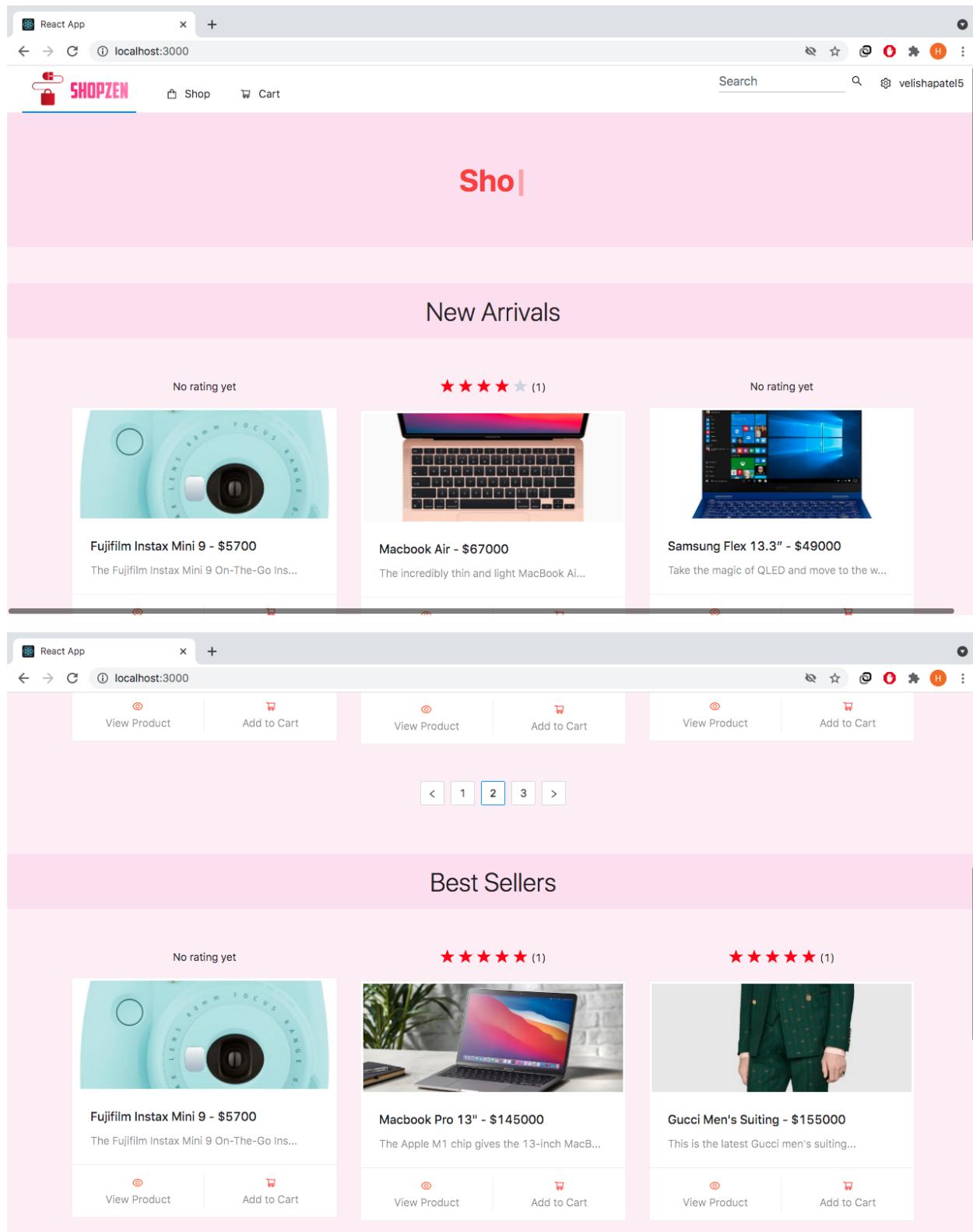
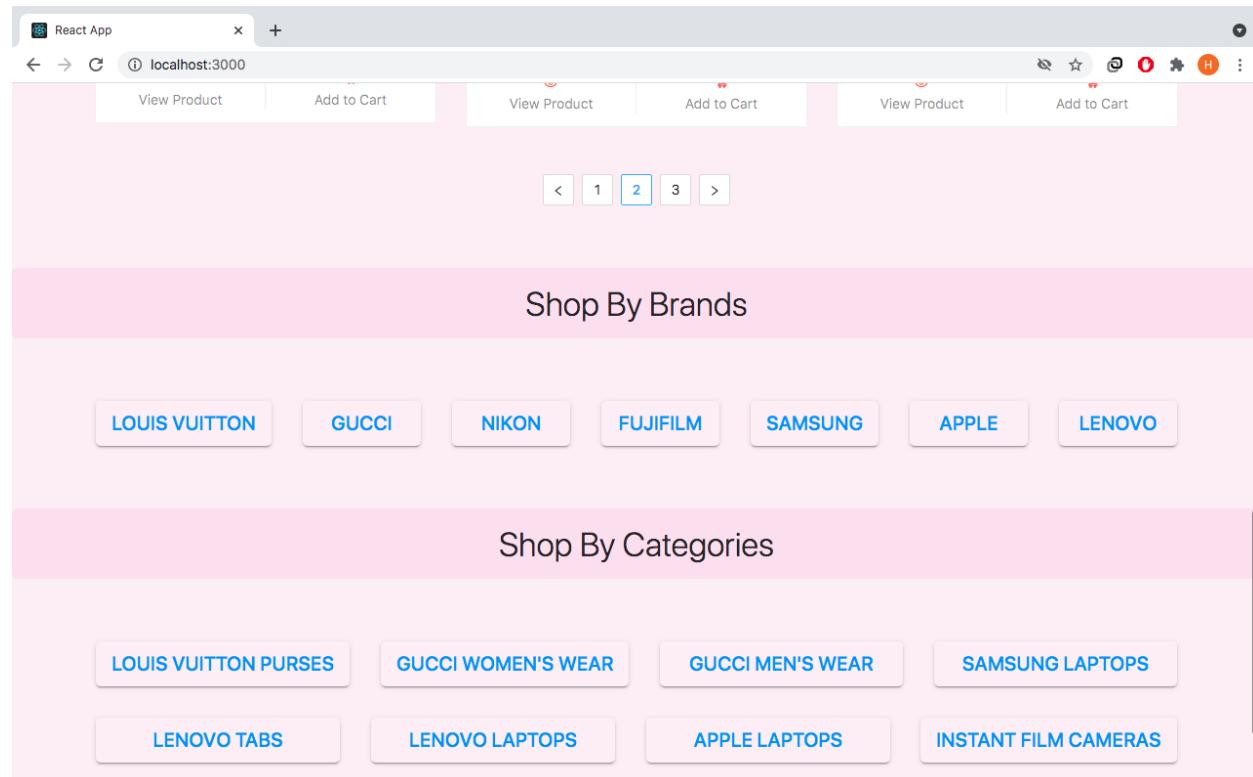


Fig 8.9 Home Page



## 10. Forgot Password? Reset it

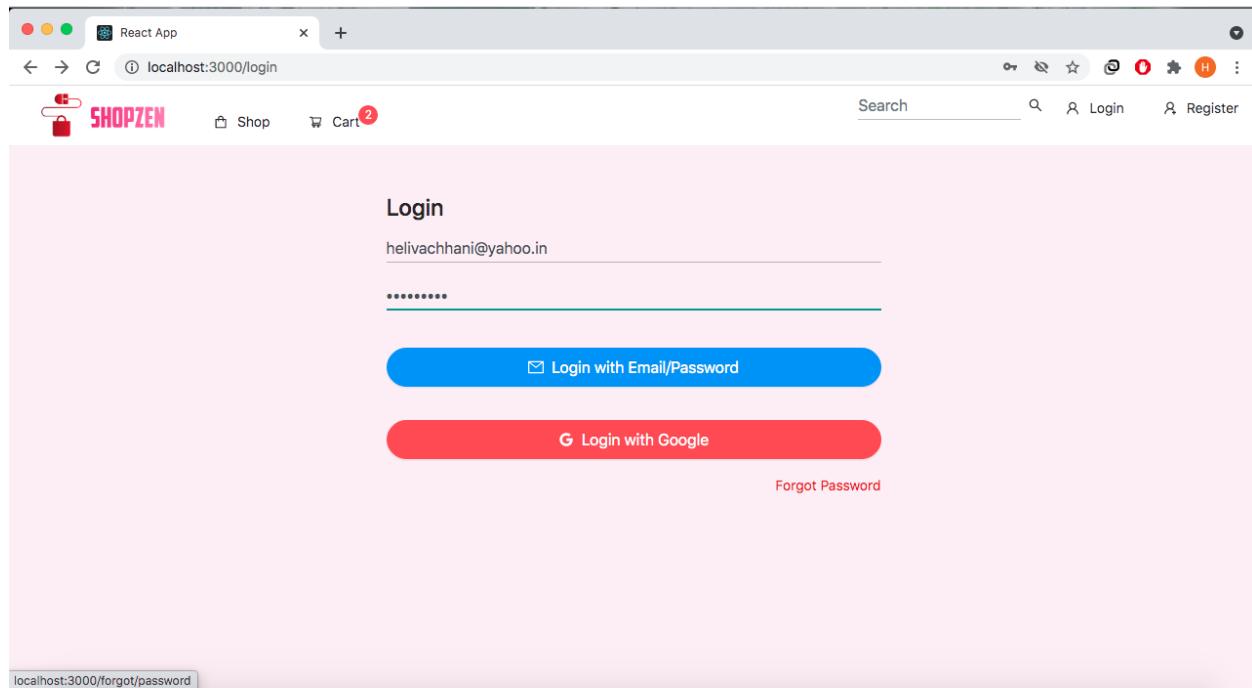


Fig 8.10 Forgot Password

## 11. Type your email address and submit.

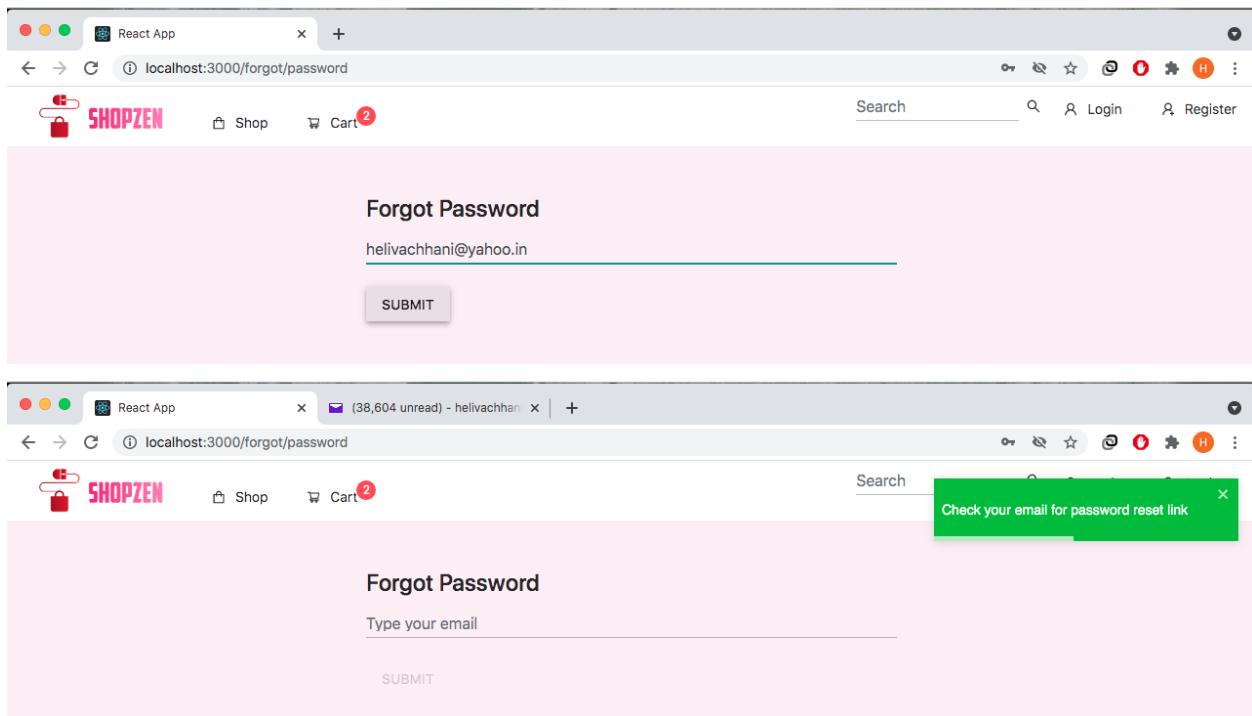


Fig 8.11 Forgot Password Continue

- **Input:** Enter mail.
- **Output:** Password Reset link sent to mail.

## 12. Follow the instructions sent to the registered and entered mail.

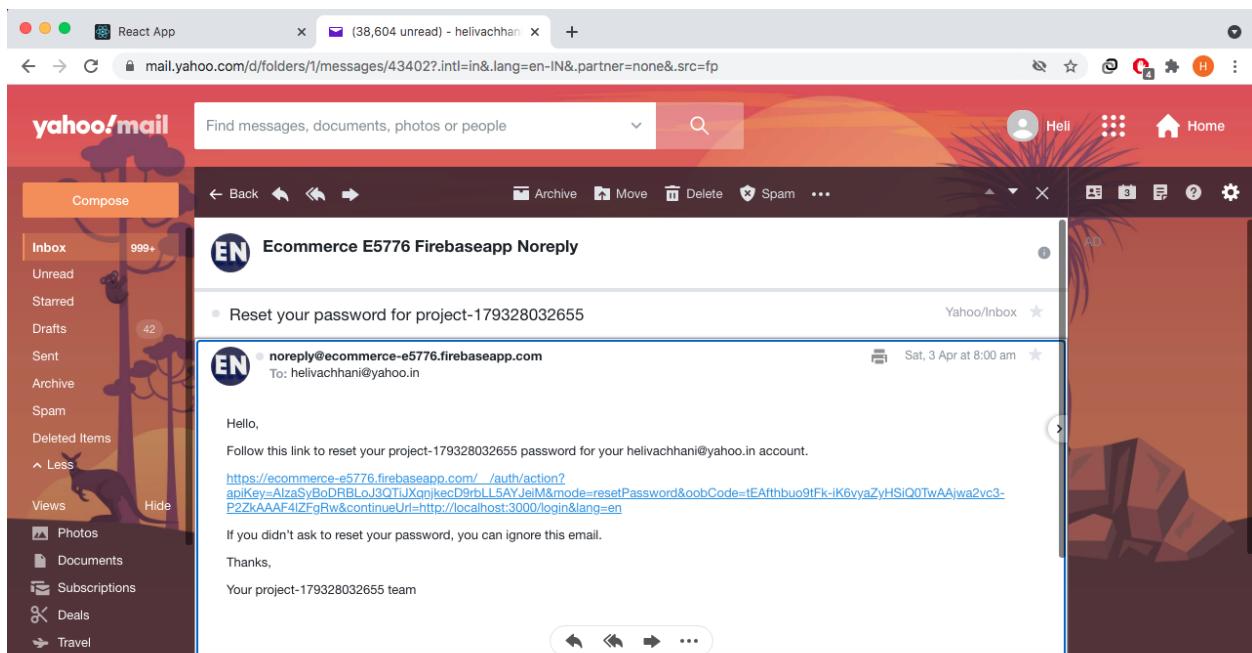


Fig 8.12 Forgot Password Continue

### 13. On clicking the link above, it will prompt to set a new Password.

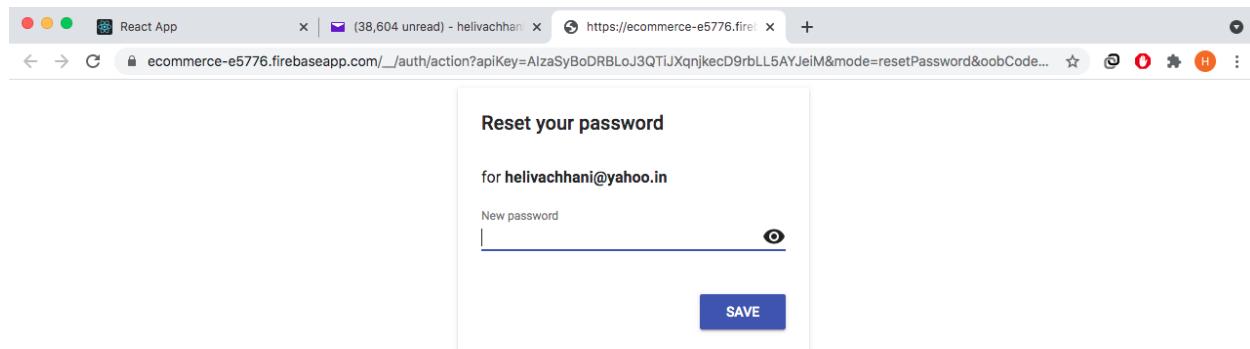


Fig 8.13 Set New Password

- **Input:** Enter new password.
- **Output:** Resets Password.

### 14. Login with New Password.

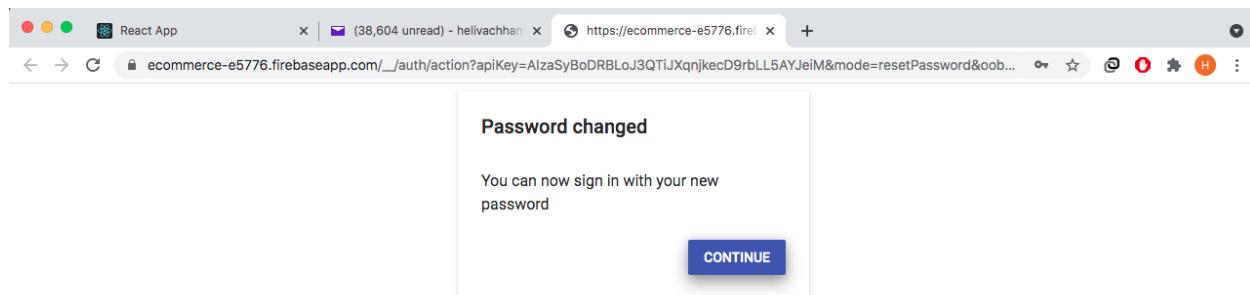


Fig 8.14 Login with New Password

## 15. Shop Page.

The screenshot shows a web browser window for 'SHOPZEN' at 'localhost:3000/shop'. The interface includes a top navigation bar with tabs for 'Shop' and 'Cart', and a search bar. On the left, there's a 'Search/Filter' sidebar with sections for 'Price', 'Categories' (listing Louis Vuitton, Gucci, Nikon, Fujifilm, Samsung, Apple, Lenovo), 'Rating' (5-star scale), and 'Sub Categories' (Louis Vuitton Purses, Gucci Women's Wear, Gucci Men's Wear, Samsung Laptops, Lenovo Tabs). The main area is titled 'Products' and displays a grid of items. Each item has a thumbnail, name, price, a short description, and 'View Product' and 'Add to Cart' buttons. The products shown are: a Nikon Coolpix camera (4 stars, \$5400), a Gucci Women's Suiting outfit (4 stars, \$18...), a Gucci Men's Suiting outfit (5 stars, \$155000), a blue camera (No rating yet), a gold laptop (4 stars, \$18...), and a blue laptop (No rating yet).

Fig 8.15 Shop Page

## 16. Search Product in top Search Bar.

The screenshot shows a web browser window for 'SHOPZEN' at 'localhost:3000/shop?Louis%20Vuitton%20Purse'. The search term 'Louis Vuitton Purse' is entered in the search bar. The rest of the interface is identical to Fig 8.15, showing the 'Search/Filter' sidebar and a grid of products. The first product in the grid is a 'LV Purse Beige Brown - \$450...' with a brown monogram pattern, described as a style for outfitting.

Fig 8.16 Search

- Input:** User searches for Louis vuitton Purse.
- Output:** According to matching Products found, they are displayed.

## 17. Filters

**Search/Filter**

⌚ Price

⌚ Categories

- Louis Vuitton
- Gucci
- Nikon
- Fujifilm
- Samsung
- Apple
- Lenovo

☆ Rating

★★★★★  
★★★★  
★★★  
★★  
★

⌚ Sub Categories

**Louis Vuitton Purses**

**Gucci Women's Wear** **Gucci Men's Wear**

**Samsung Laptops** **Lenovo Tabs**

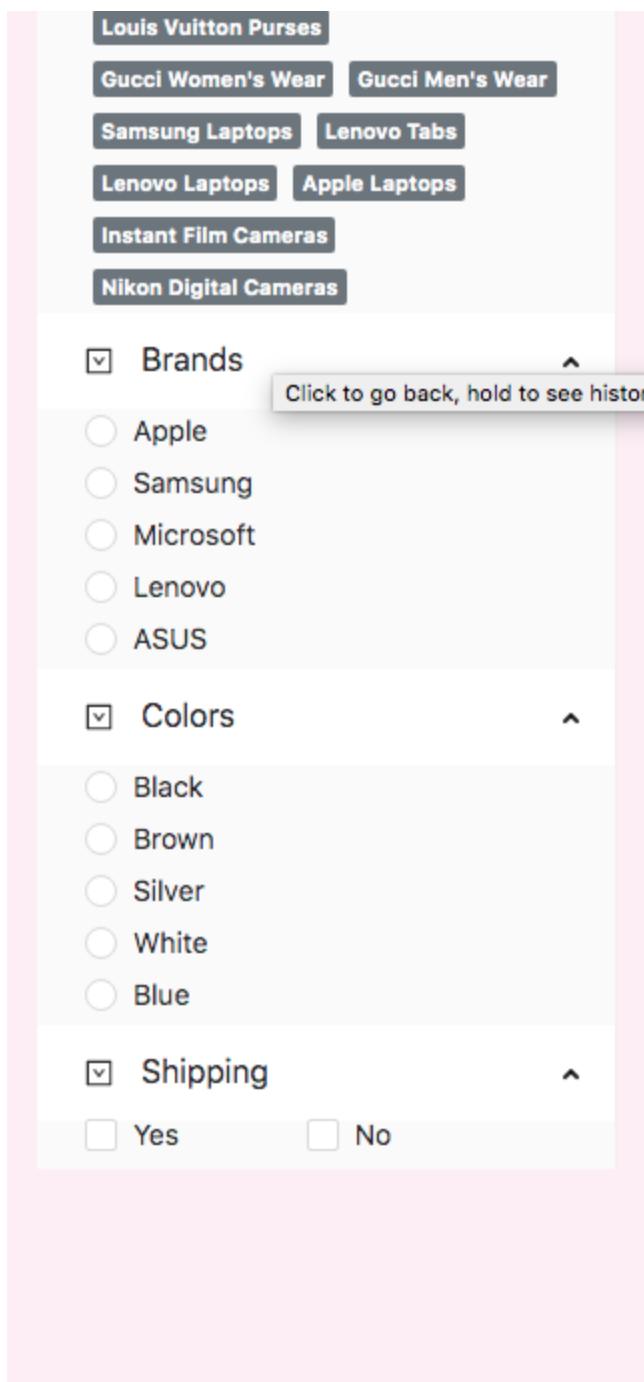


Fig 8.17 Filter List (list to filters that can be applied)

- **Input:** User selects a Filter to filter products based on various categories.
- **Output:** According to the user's selected filters, matching products found are displayed.

## 18. When filters are used(filter by categories applied below)

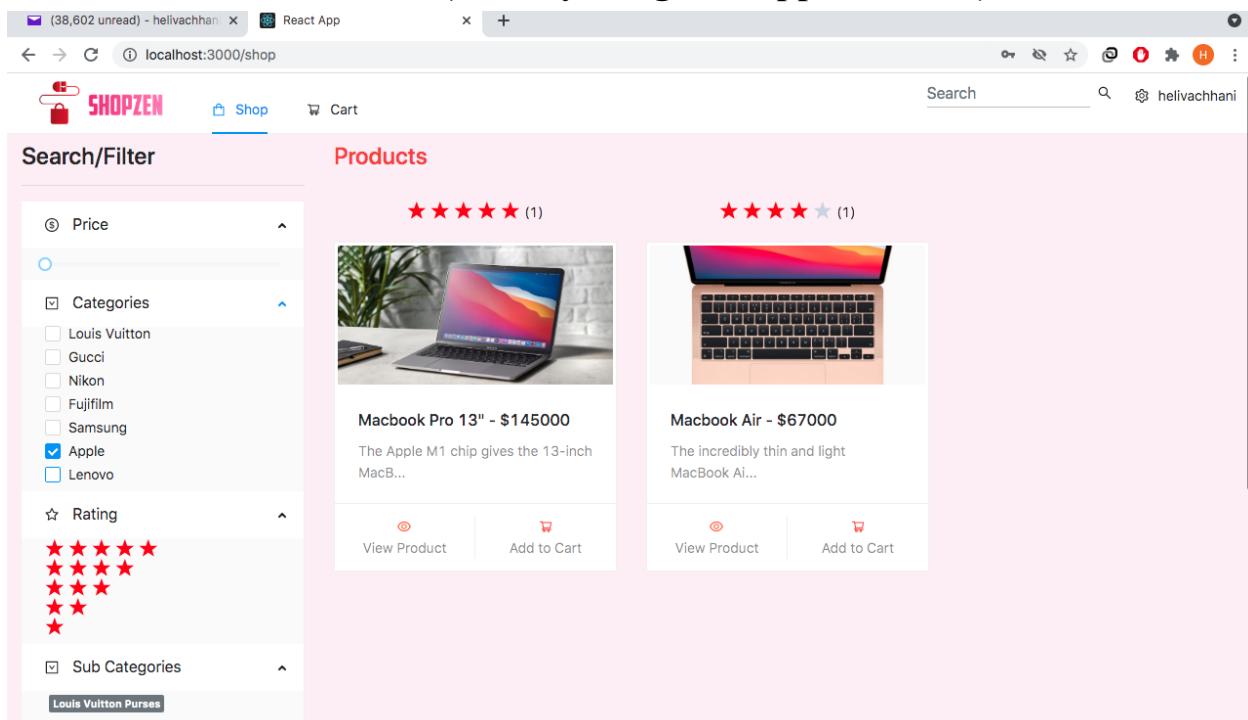


Fig 8.18 Filter applied

## 19. To add products in cart, click on 'Add to Cart'

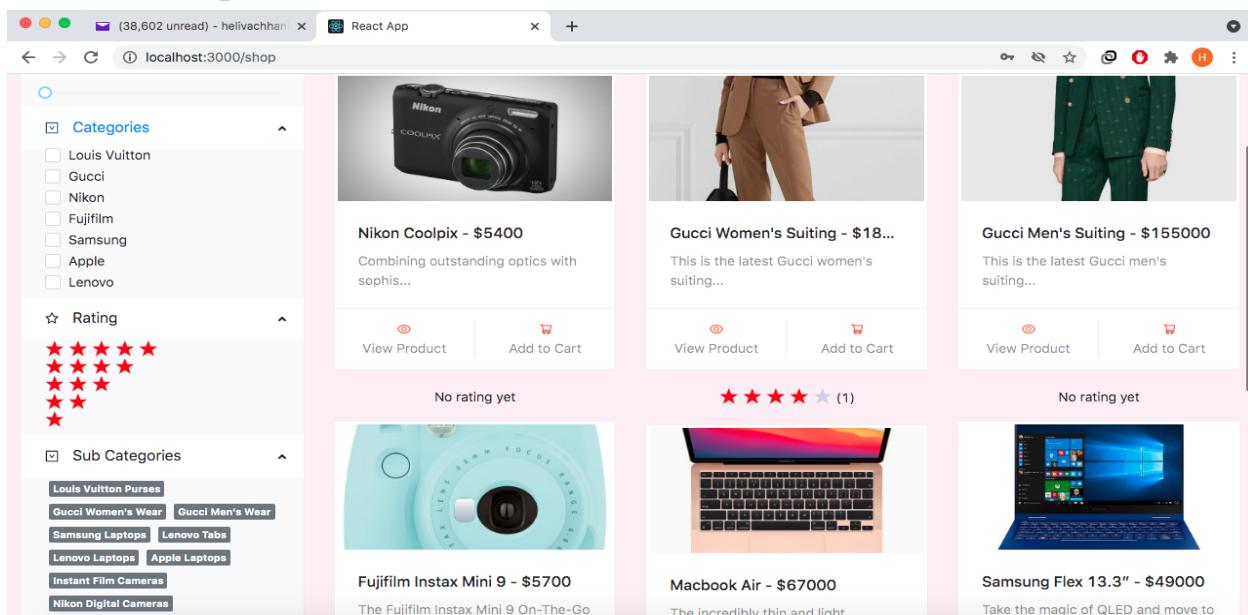


Fig 8.19 Add to Cart

- **Input:** User clicks on “Add to Cart” button and adds that product in the cart.
- **Output:** Product is added to Cart.

## 20. After adding products you want to buy

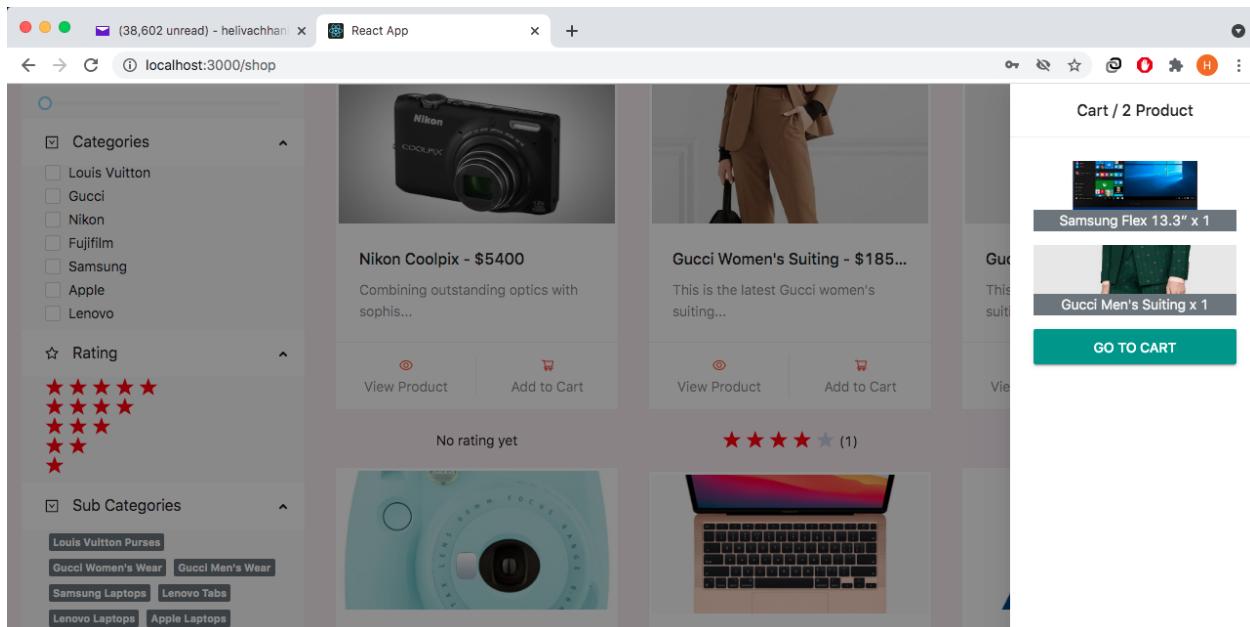


Fig 8.20 Add to Cart sidebar popup

## 21. Go to Cart

The screenshot shows the 'Cart / 2 Product' page. The table lists two items: 'Samsung Flex 13.3" x 1' and 'Gucci Men's Suiting'. The 'Order Summary' section shows a total of \$204000. At the bottom, there are buttons for 'PROCEED TO CHECKOUT' and 'PAY CASH ON DELIVERY'.

Image	Title	Price	Brand	Color	Count	Shipping	Remove
	Samsung Flex 13.3"	\$49000	Samsung	Blue	1		
	Gucci Men's Suiting	\$155000	Gucci	Greer	1		

Fig 8.21 Cart Page

- **Input:** User can remove, change color, increase/ decrease count of product, empty cart and proceed to pay by card method or cash on delivery.
- **Output:** Product in Cart modified according to User's choice.

## 22. Proceed to Pay by Checking out (using card-Real Time Payment)

Fill in the checkout details and apply a coupon code to get a discount.

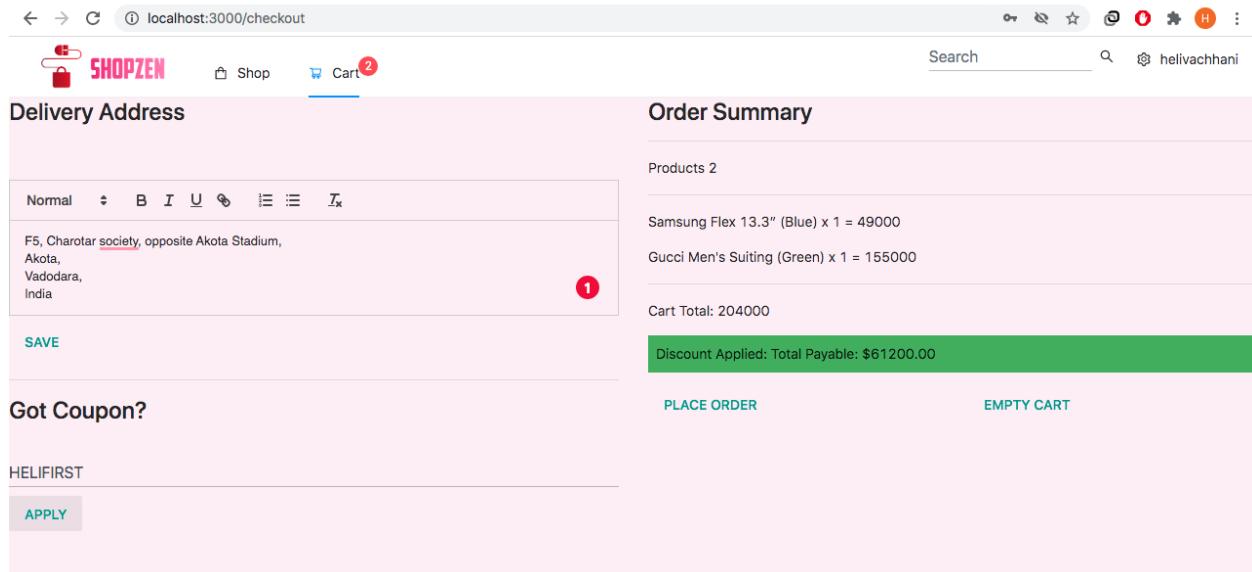


Fig 8.22 On Click Proceed to Checkout

- Input:** Fill Checkout page, where Order summary is displayed. Add Coupon also.
- Output:** Shipping address saved to database, discount applied & net pay is calculated.

## 23. Click on place Order to place the order and Empty cart to cancel.

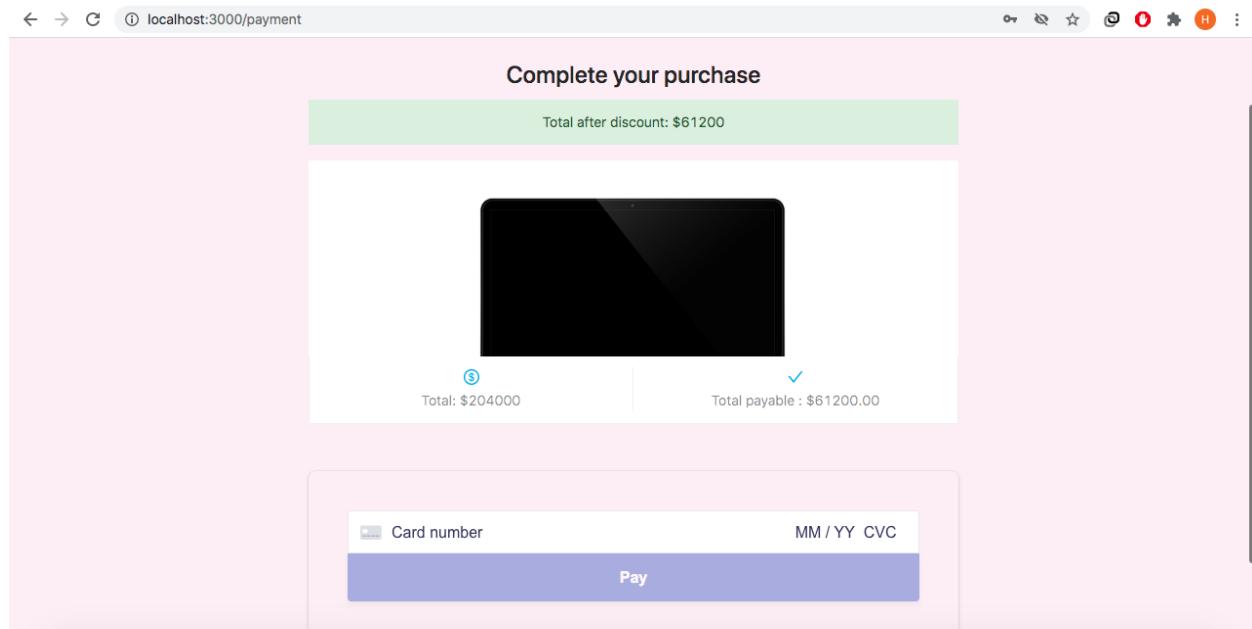


Fig 8.23 Complete purchase, Enter card details

- Input:** Provide Card no. and click "Pay" button.
- Output:** Payment done and User History Page Link is displayed.

## 24. Enter card details to pay. Click on pay.

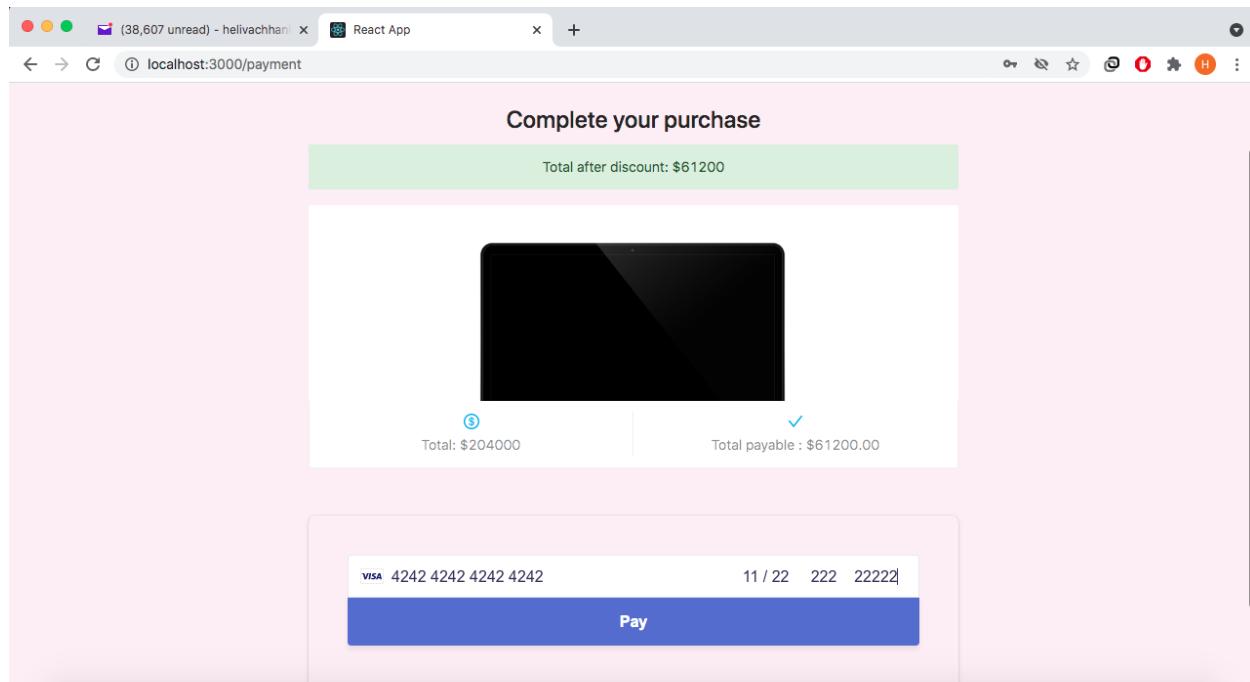


Fig 8.24 Pay

## 25. Click purchase history to see your history of purchases.

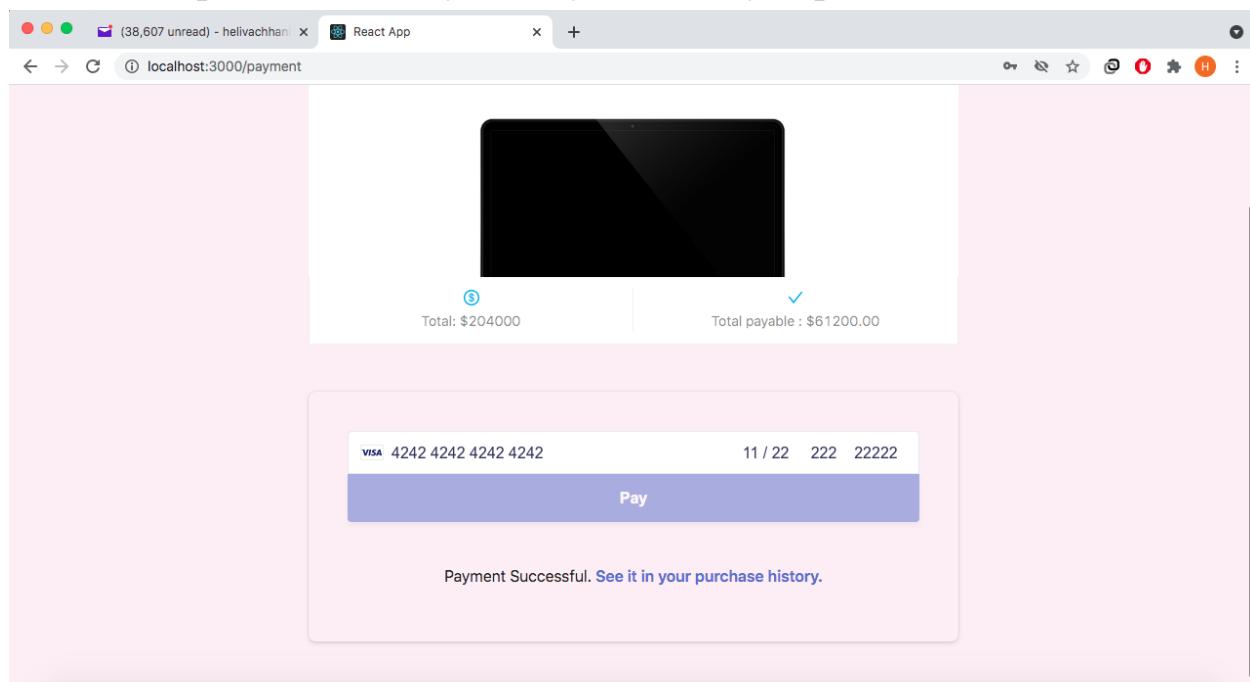


Fig 8.25 See Purchase History

- **Input:** User clicks on “See it in your purchase history”.
- **Output:** directed to Purchase history page.

## 26. Payment by Cashless Order (cash on delivery)

**Click pay cash on delivery. And fill checkout details (same process now onwards like that of Card- Real Time Payment)**

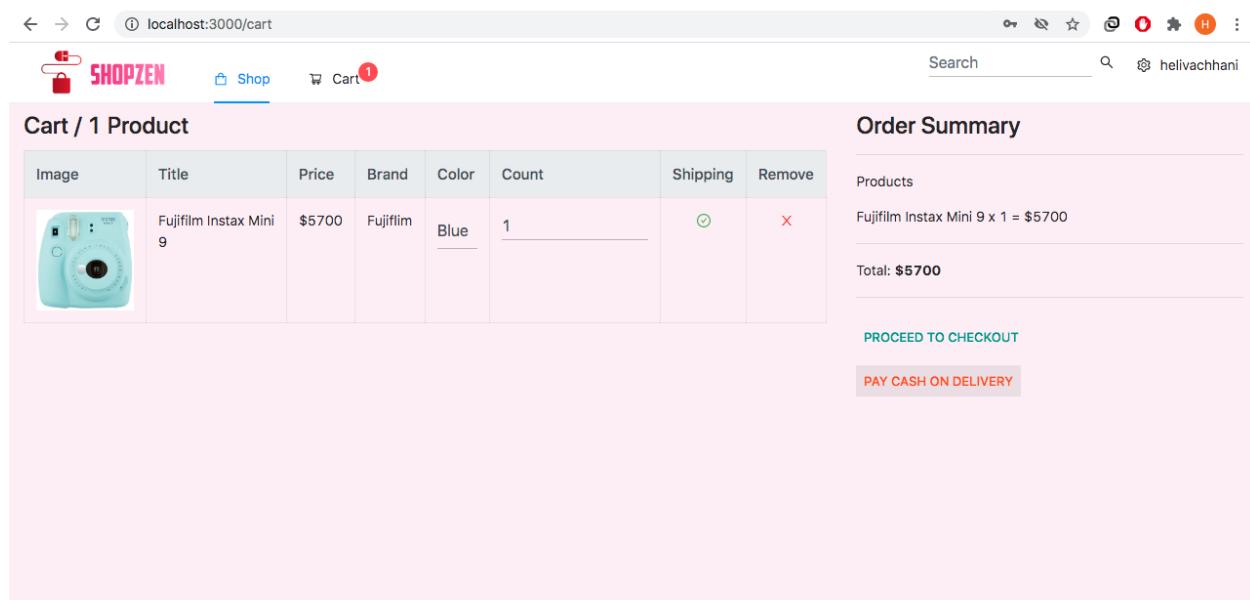


Fig 8.26 On Click Pay Cash On Delivery

## 27. Users cannot buy more than Max. quantity of Product currently available. (here if the user tries to buy more than 13(max. quantity available) camera, he/she will get error)

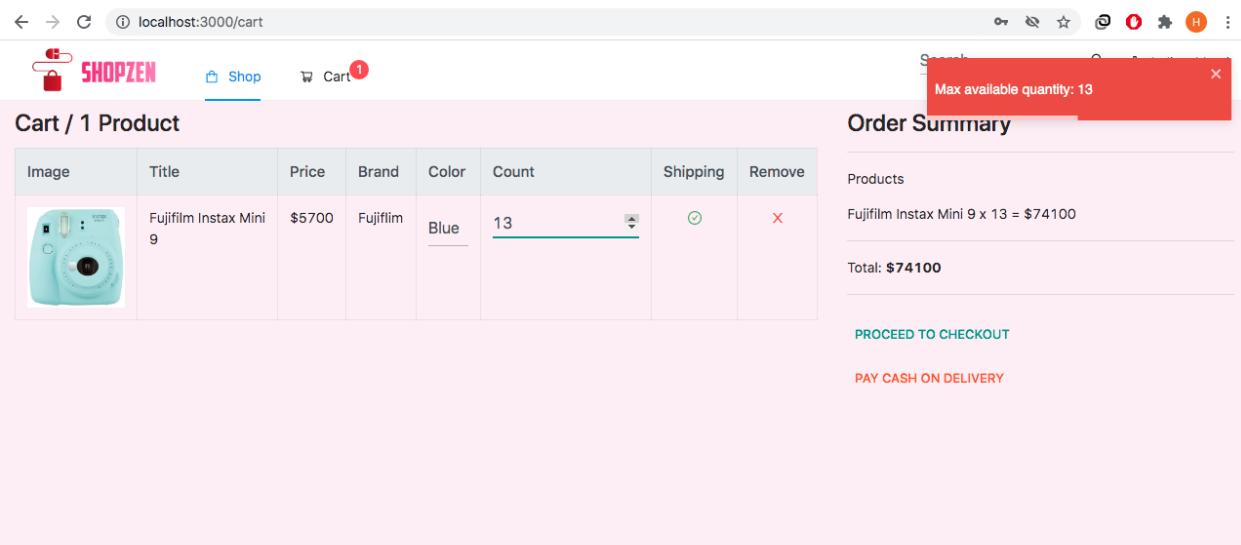
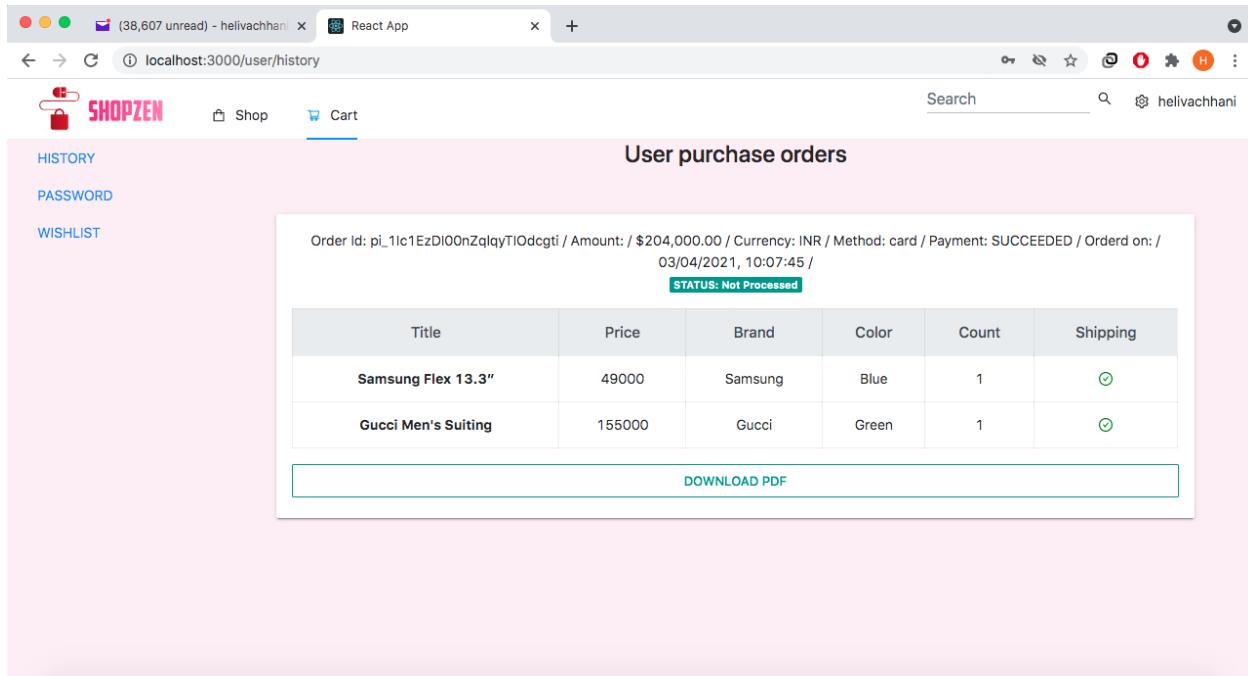


Fig 8.27 On Click Proceed to Checkout

- **Input:** User wants to buy more than available quantity of product.
- **Output:** Error Displayed.

## 28. History of purchases of user [helivachhani@yahoo.in](mailto:helivachhani@yahoo.in) who made single purchase till now.



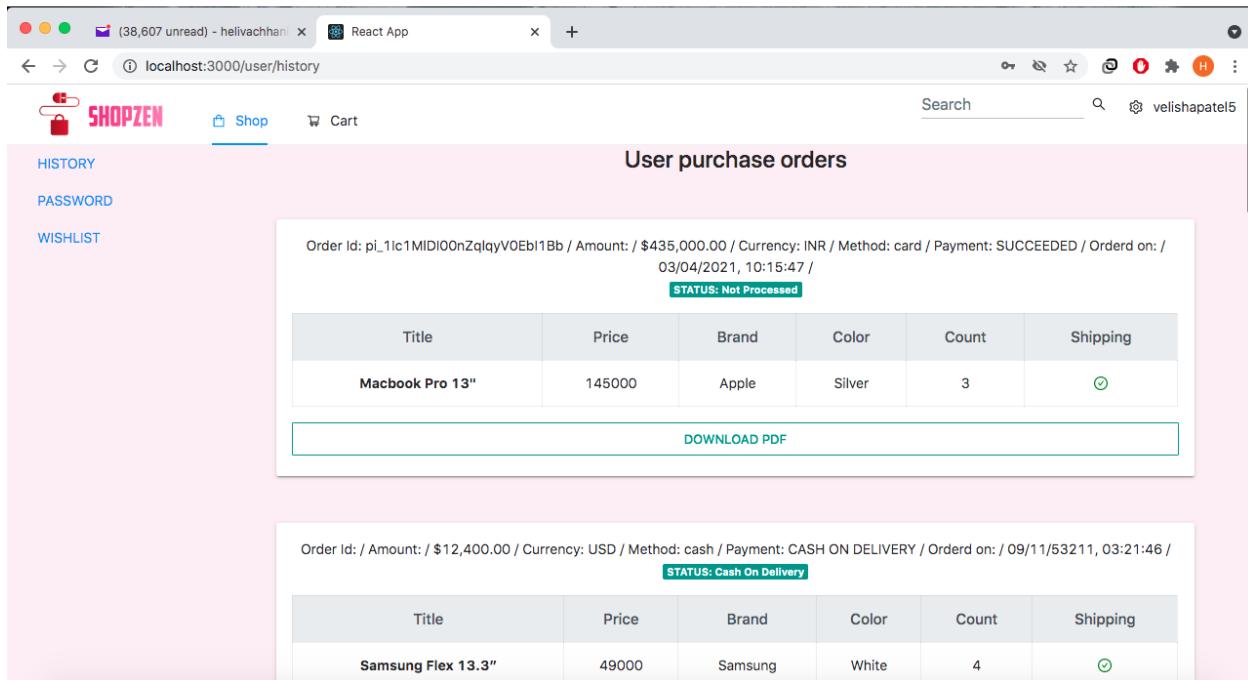
The screenshot shows a web browser window with the URL [localhost:3000/user/history](http://localhost:3000/user/history). The page title is "User purchase orders". On the left, there are navigation links for "HISTORY", "PASSWORD", and "WISHLIST". The main content area displays a single purchase order with the following details:

Title	Price	Brand	Color	Count	Shipping
Samsung Flex 13.3"	49000	Samsung	Blue	1	<input checked="" type="checkbox"/>
Gucci Men's Suiting	155000	Gucci	Green	1	<input checked="" type="checkbox"/>

Below the table is a green button labeled "DOWNLOAD PDF". Above the table, the order ID and amount are listed: Order Id: pi\_1lc1EzDl00nZqlqyTlOdcgti / Amount: \$204,000.00. The status is shown as "STATUS: Not Processed".

Fig 8.28 User Purchase History Page with single order purchase

## 29. History of purchases of another user who made multiple purchases from Shopzen.



The screenshot shows a web browser window with the URL [localhost:3000/user/history](http://localhost:3000/user/history). The page title is "User purchase orders". On the left, there are navigation links for "HISTORY", "PASSWORD", and "WISHLIST". The main content area displays two purchase orders. The first order is for a Macbook Pro 13" with the following details:

Title	Price	Brand	Color	Count	Shipping
Macbook Pro 13"	145000	Apple	Silver	3	<input checked="" type="checkbox"/>

Below the table is a green button labeled "DOWNLOAD PDF". Above the table, the order ID and amount are listed: Order Id: pi\_1lc1MIDI00nZalqyV0EbI1Bb / Amount: \$435,000.00. The status is shown as "STATUS: Not Processed".

The second order is for a Samsung Flex 13.3" with the following details:

Title	Price	Brand	Color	Count	Shipping
Samsung Flex 13.3"	49000	Samsung	White	4	<input checked="" type="checkbox"/>

Below the table, the order ID and amount are listed: Order Id: / Amount: \$12,400.00. The status is shown as "STATUS: Cash On Delivery".

Fig 8.29 User Purchase History Page with Multiple order purchases

### 30. To download the invoice, click on Download PDF.

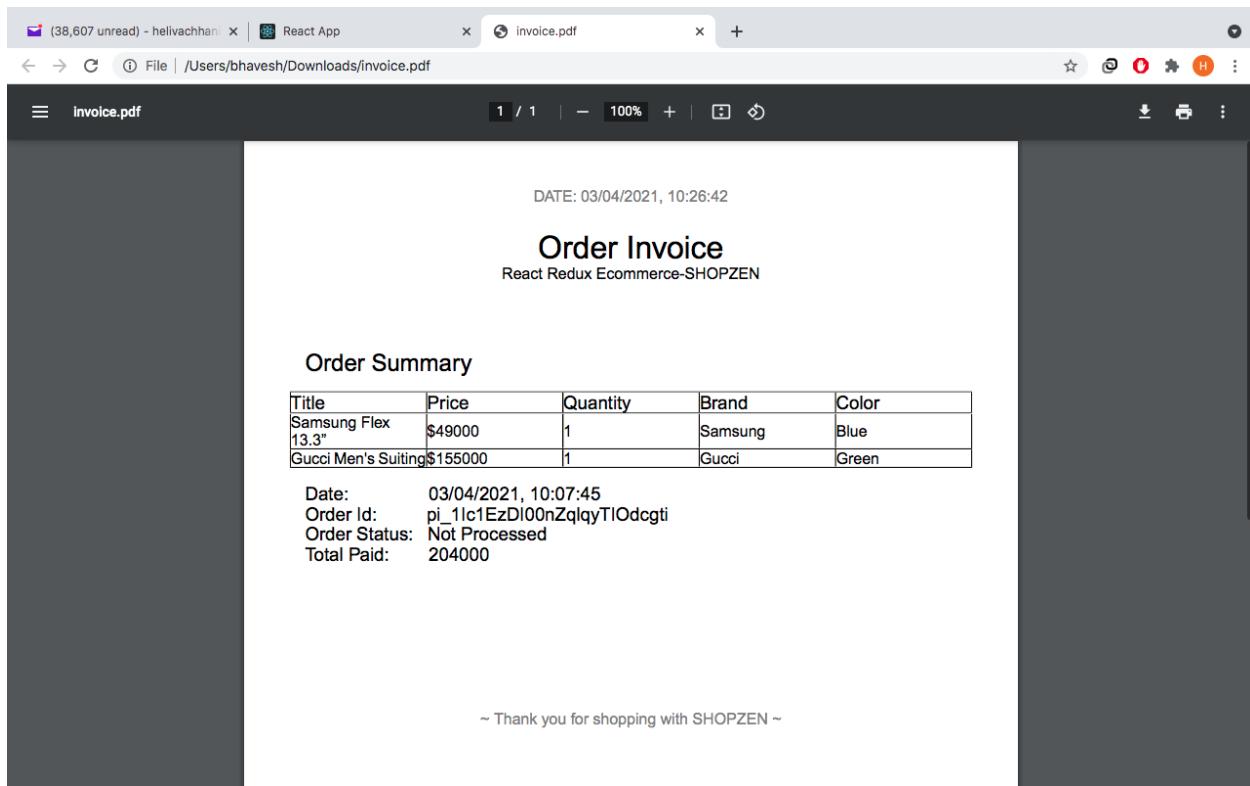


Fig 8.30 Invoice Download

- **Input:** When user clicks on “download pdf”.
- **Output:** Order Invoice in pdf form gets downloaded.

### 31. Password Reset.

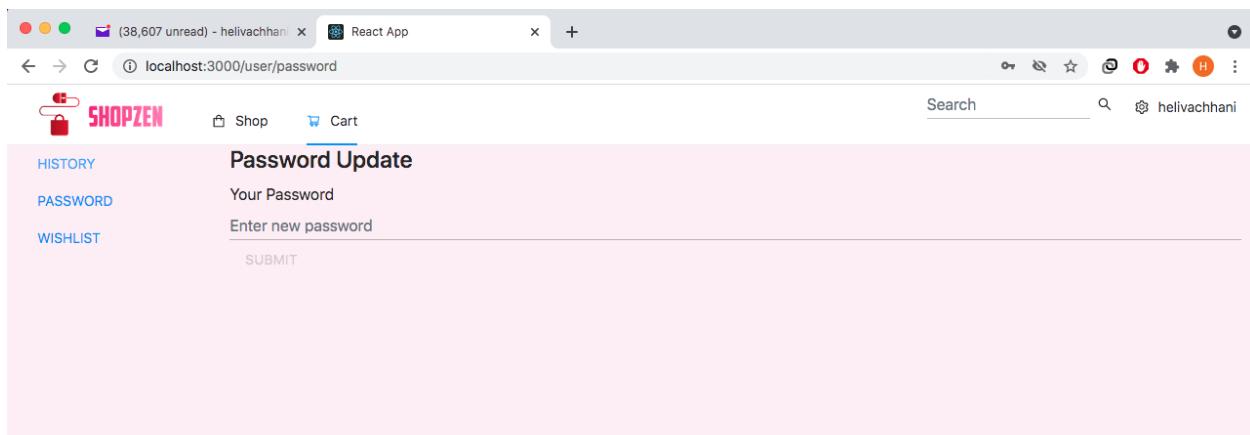


Fig 8.31 Reset Password

- **Input:** Enter new password and click “Submit”.
- **Output:** Resets Password.

### 32. Rating a Product (with Averaging)

Suppose if one user rated a product with 5 stars and another with 3, then an average of 4, i.e. 4 star rating will be displayed with user count 2.

The screenshots illustrate the product rating process:

**Screenshot 1: Rating Input**

A modal window titled "Leave your rating" is open, showing a 5-star rating scale with the first four stars filled (red) and the fifth star unfilled (gray). Below the scale are "Cancel" and "OK" buttons.

**Screenshot 2: Confirmation**

The modal window has closed, and a green notification message "Thanks for your review. It will appear soon" is displayed in the top right corner of the page.

(average rating displayed)

The screenshot shows the product page for the Fujifilm Instax Mini 9. The product image is visible on the left, and the product title "Fujifilm Instax Mini 9" is prominently displayed in a blue header box. Below the title, the average rating is shown as "★★★★☆ (2)".

Fig 8.32 Product Rating (and display Average Rating)

- **Input:** User gives ratings.
- **Output:** Average Rating Displayed to every user.

### 33. Wishlist a product.

This is the latest Gucci men's suiting

**Related Products**

★★★★★ (1)

Added to wishlist

Fig 8.33 Wishlist (and how to wishlist a Product)

- **Input:** User clicks on “Add to wishlist”.
- **Output:** That product gets added to their personal wishlist and remains there until the user deletes it.

## 34. Related Products

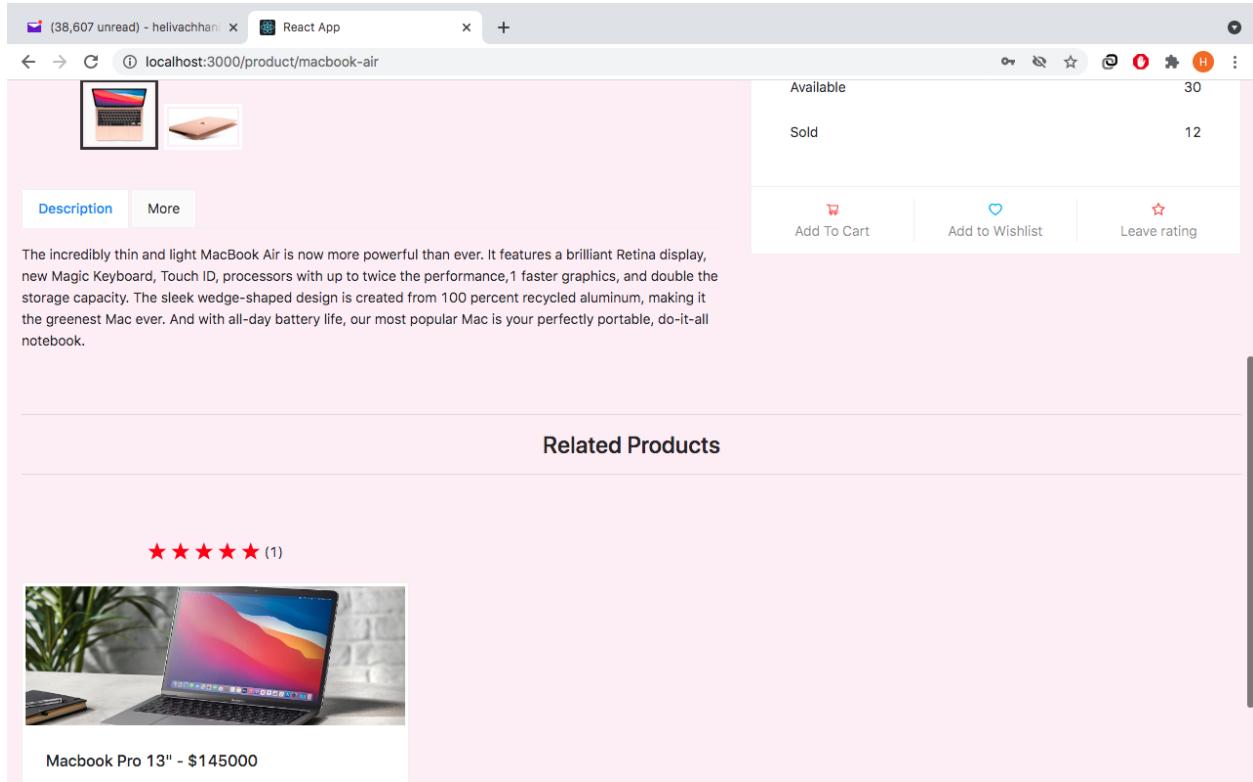


Fig 8.34 Related Products

- **Input:** When a user scrolls down a product under view product, if products related to the currently viewed product are found, they are displayed.
- **Output:** Related Products are displayed.

## ~ Extra Controls on Admin Side ~

### 35. Add Product

DASHBOARD      **Product create**

PRODUCT       [CHOOSE FILE](#)

PRODUCTS

CATEGORY

SUB CATEGORY

COUPON

PASSWORD

Title: LV Sling Purse Brown

Description: LV x Signature Monogram

Price: 35000

Shipping: Yes

Quantity: 50

Color: Brown

Brand:

Fig 8.35 Add Product

### 36. Update Product

DASHBOARD      **All Products**

PRODUCT

PRODUCTS

CATEGORY

SUB CATEGORY

COUPON

PASSWORD

Shopzen

Cart 1

Search velishapatel5

 LV Purse Beigh Brown Style every outfit with this LV purse wh... <a href="#">Edit</a> <a href="#">Delete</a>	 Nikon Coolpix Combining outstanding optics with sophis... <a href="#">Edit</a> <a href="#">Delete</a>	 Gucci Women's Suiting This is the latest Gucci women's suiting... <a href="#">Edit</a> <a href="#">Delete</a>
 Gucci Men's Suiting	 Fujiifilm Instax Mini 9	 Macbook Air

Fig 8.36 Update Product

### 37. Add Brand category

The screenshot shows the 'Create category' page of the ShopZen admin interface. On the left, a sidebar lists navigation options: DASHBOARD, PRODUCT, PRODUCTS, CATEGORY, SUB CATEGORY, COUPON, and PASSWORD. The 'CATEGORY' option is currently selected. The main area has a title 'Create category' and a 'Name' input field which is empty. Below the input field is a 'SAVE' button. To the right of the input field is a 'Filter' placeholder. A scrollable list of brand names is displayed, each with edit and delete icons. The brands listed are Louis Vuitton, Gucci, Nikon, Fujifilm, and Samsung.

Fig 8.37 Add Brand Category

### 38. Edit/ Update Brand category

This screenshot shows the 'Update category' page for the brand 'louis-vuitton'. The URL in the browser bar is 'localhost:3000/admin/category/louis-vuitton'. The left sidebar and the main form structure are identical to Fig 8.37. The 'Name' input field now contains the value 'louis-vuitton'. The 'SAVE' button is located below the input field. The rest of the interface, including the list of brands and the sidebar, remains the same.

Fig 8.38 Edit/Update Brand Category

### 39. Add Subcategories under Brand

Create sub category

Parent category

Gucci

Name

Gucci Purses

**SAVE**

Filter

- Louis Vuitton Purses
- Gucci Women's Wear
- Gucci Men's Wear
- Samsung Laptops

Fig 8.39 Add Sub Categories under Brand

### 40. Edit/ Update Subcategories under Brand

Filter

- Louis Vuitton Purses
- Gucci Women's Wear
- Gucci Men's Wear

localhost:3000/admin/sub/gucci-men's-wear

ShopZen

Shop Cart 1

Search velishapate15

Update sub category

Parent category

Gucci

Name

new/ updated name goes here

**SAVE**

Fig 8.40 Edit/Update Subcategories under Brand

## 41. Set/Update Delivery status (from Admin dashboard)

The screenshot shows the Admin Dashboard interface. On the left, there's a sidebar with links: DASHBOARD, PRODUCT, PRODUCTS, CATEGORY, SUB CATEGORY, COUPON, and PASSWORD. The main area has a heading "Admin Dashboard". Below it, there are two sections. The first section displays an order with details: ORDER ID: / AMOUNT: / \$74,100.00 / CURRENCY: USD / METHOD: CASH / PAYMENT: CASH ON DELIVERY / ORDERD ON: / 08/03/53224, 05:48:01 / Not Processed. It includes a "DELIVERY STATUS" dropdown menu with options: Not Processed, ✓ Cash On Delivery, Processing, Dispatched (which is highlighted in blue), Cancelled, and Completed. The second section displays another order with details: ORDER ID: PI\_1IC1MLDIOONZQLQYV0EBI1BB / AMOUNT: / \$435,000.00 / CURRENCY: INR / METHOD: CARD / PAYMENT: SUCCEEDED / ORDERD ON: / 03/04/. It also includes a "DELIVERY STATUS" dropdown menu with the option Not Processed.

Title	Price	Brand	Color	Count	Shipping
Fujifilm Instax Mini 9	5700	Fujifilm	Blue	13	<input checked="" type="checkbox"/>

(admin changed/ set the delivery status to product being dispatched)

This screenshot is similar to the previous one but includes a green toastify-style notification bar at the top right with the text "Status updated". The rest of the interface and data tables are identical to the first screenshot.

Fig 8.41 Set/Update Delivery Status

- **Input:** Admin sets or updates delivery status from Admin dashboard.
- **Output:** Delivery Status updated/ set by admin. 'Status Updated' toastify notification.

## 42. Create Discount Coupons (set expiry date, coupon name, discount %)

The screenshot shows the ShopZen admin interface for creating a new coupon. The left sidebar has links for DASHBOARD, PRODUCT, PRODUCTS, CATEGORY, SUB CATEGORY, COUPON, and PASSWORD. The main form is for creating a 'Coupon' with fields for Name (NEWCUST), Discount % (45), and Expiry (04/03/2021). A 'SAVE' button is at the bottom. Below the form, a table titled '3 Coupons' lists existing coupons: WINTER (Expiry 23/06/2021, Discount 30%), NEWUSER (Expiry 02/06/2021, Discount 75%), and HELIFIRST (Expiry 12/03/2021, Discount 70%).

**Success Message:** "NEWCUST" is created

Name	Expiry	Discount	Action
WINTER	23/06/2021	30%	
NEWUSER	02/06/2021	75%	
HELIFIRST	12/03/2021	70%	

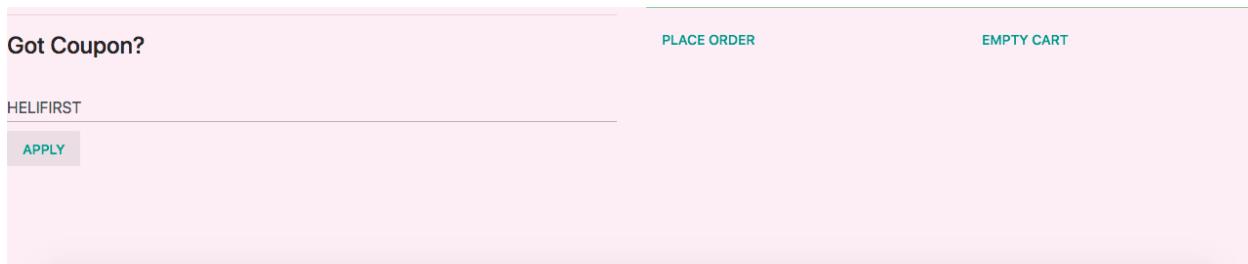
  

The screenshot shows the ShopZen admin interface after creating a new coupon. The left sidebar has links for DASHBOARD, PRODUCT, PRODUCTS, CATEGORY, SUB CATEGORY, COUPON, and PASSWORD. The main form is for creating a 'Coupon' with fields for Name (NEWCUST), Discount % (45), and Expiry (04/03/2021). A 'SAVE' button is at the bottom. Below the form, a table titled '4 Coupons' lists existing coupons: NEWCUST (Expiry 25/06/2021, Discount 45%), WINTER (Expiry 23/06/2021, Discount 30%), NEWUSER (Expiry 02/06/2021, Discount 75%), and HELIFIRST (Expiry 12/03/2021, Discount 70%).

Fig 8.42 Create Coupon

- **Input:** Admin creates coupon by specifying its Name, Discount %, Expiry date (a future date). After the coupon is created, when the coupon expires, the admin can delete the coupon.
- **Output:** Coupon is generated. And when the admin clicks on delete action, the coupon gets deleted.

### 43. Use coupons at the checkout page to get discounts.



If coupons invalid, display an error message.



Fig 8.43 Apply Coupon, if Invalid display Error

- **Input:** User Enter's coupon name.
- **Output:** 'Invalid coupon' message displayed if wrong coupon name is entered, else discount coupon is applied successfully and net payable amount is calculated.

## 44. FIREBASE AUTHENTICATION (users can be deleted from here)

The screenshot shows the Firebase Authentication console for a project named 'ECOMMERCE'. The left sidebar includes options for Project Overview, Build (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release & Monitor, and Extensions. The main area is titled 'Authentication' and shows the 'Users' tab selected. A banner at the top right says 'Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication' with a 'Get started' button. Below is a table of users with columns for Identifier, Providers, Created, Signed In, and User UID. Two users are listed: 'velishapatel5@gmail.com' and 'helivachhani@yahoo.in'. At the bottom, there are pagination controls for 'Rows per page: 50' and '1 - 2 of 2'.

Fig 8.44 Firebase Authentication

## 45. REAL TIME PAYMENT SUCCESS AND FAILURE (when user does not complete payment procedure) STATISTICS

The screenshot shows the Stripe dashboard under 'New Business'. The left sidebar lists 'Home', 'Activate your account', 'Payments' (Reviews, Disputes, Payouts, All transactions), 'Balances', 'Customers', 'Connected accounts', 'Products', 'Reports', 'Developers', 'Viewing test data', and 'Settings'. The main area is titled 'Payments' and shows a table of transactions. The table has columns for 'AMOUNT', 'DESCRIPTION', 'CUSTOMER', and 'DATE'. The transactions are categorized by status: 'Succeeded' (green), 'Incomplete' (grey), and 'Refunded' (light blue). Some descriptions mention 'Software development services'. The table includes rows for multiple transactions on different dates, such as 3 Apr, 26 Mar, and 13 Mar.

Fig 8.45 Real Time Payment Statistic

## **9.0 LIMITATION AND FUTURE ENHANCEMENT**

### **9.1 LIMITATIONS:**

- Users cannot add products.
- Only one card gateway for Real time payment is available but one can always integrate more gateways.

### **9.2 FUTURE ENHANCEMENT**

- Trial room facilities can be incorporated.
- Chatbot Recommendation (AI assistant) system can be incorporated.
- GUI can be improved.
- Another scope can be to expand the number of places/ locations/ states/ country the product can be delivered to the buyer.
- Increase the range of variety of products under each category and increase the number of different types of categories by making more associations with concerned companies/ sellers.
- Image search can be incorporated.

## **10. CONCLUSION AND DISCUSSION**

### **10.1 CONCLUSION**

According to us, this project gave us the confidence to believe in ourselves and a great experience of how to work as a team. It also boosted our technical coding as well as time management skills.

### **10.2 DISCUSSION**

#### **10.2.1 Self-Analysis of Project Viabilities**

According to us, this project is absolutely a good start for gaining hands-on experience on projects. It is useful if it is managed according to the goal for which it is made.

#### **10.2.2 Problems Encountered and Possible Solutions**

There are so many problems encountered during this project.

1. Problem to maintain databases.
2. Need to change some functionality fully which lead to do the whole work again

#### **10.2.3 Summary of Project Work**

It is a great achievement to successfully complete the project. The knowledge of software engineering has helped immensely in overcoming the various road blocks. We have done work with pre-planned scheduling related with time constraints and result oriented progress in project development.

## 11. BIBLIOGRAPHY

- **Reference docs:**

React Official docs

<https://reactjs.org/docs/getting-started.html>

MongoDB Official docs

<https://docs.mongodb.com/>

NodeJS Official docs

<https://nodejs.org/en/docs/>

Firebase Authentication

<https://firebase.google.com/docs/auth/web/start>

Cloudinary Official docs

[https://cloudinary.com/documentation/image\\_video\\_and\\_file\\_upload](https://cloudinary.com/documentation/image_video_and_file_upload)

Antd

<https://ant.design/docs/react/introduce>

Toastify docs

<https://www.npmjs.com/package/react-toastify>

- **Reference sites for debugging:**

Stack Overflow, Medium