Smart Checkout System

PROJECT REPORT

Submitted in fulfilment for the J-Component of CSE3002 – Internet and Web Programming

CAL COURSE

in

B.Tech. (Computer Science)

by

Mihir Mangesh Pavuskar 18BCE0159
Aashraya Singal 18BCE0171

Under the guidance of

Prof. Jayakumar Sadhasivam

SCOPE

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Fall Semester 2021-2022

Abstract: **3**

Introduction: **3**

Problem Statement: **4**

Process Flow Diagram **4**

Customers 4

For owner- 5

Module Description: **5**

1 For website 5

1.1) Login Page: 5

1.2) Dashboard Page: 5

1.3) Products Page: 6

1.4) Categories Page: 6

1.5)Profile Page 6

1.6) Transaction Page 6

2 Mobile App 6

2.1) Login Screen 6

2.2) Qrcode Scanner 6

2.3) Product details screen 6

2.4) Transaction Screen 6

ER Diagrams- **7**

Database Design: **9**

Conclusion: **11**

References: **11**

Snapshots : **12**

Web Interface (Seller) 12

1. Login Page 12

2. Dashboard 12

3. Products Page 13

4. Create New Product Page 13

5. Categories Page 14

6. Create New Category Page 14

7. Profile Page 15

Mobile Interface (Buyer) 16

QrCode Scanner Add to Cart 16

Cart Checkout 17

Profile 18

Source Code: **18**

Repository Link:

https://github.com/MeMihir/SmartCheckout

Abstract:

A fully functional checkout system which makes sure your visit to the local daily needs store is completely safe and hassle-free! Imagine if you could shop at your local supermarket, without having to queue at checkouts to pay for your groceries. Dealers and small store owners don't have checkouts where a customer can buy. It is all managed and maintained by owner. So, we have thought of a solution to minimize the workload and man-work. Whenever a user wants to buy a product, he can scan the QR code and get the product details, then add it to the cart and pay online.

Introduction:

Imagine if you could shop at your local supermarket, without having to queue at checkouts to pay for your groceries. Dealers and small store owners don't have checkouts where a customers can buy. It is all managed and maintained by owner.

We will be creating both mobile and web application.

The owner would have access to dashboard where he can manage his products, while adding a product QR Code would be generated and he can take printout to stick it on the counter. They will be following a canteen type system.

Whenever a user wants to buy a product, he can scan the code and get the details through a mobile app (Flutter). He will get an option to buy and if customer clicked yes, the amount of product will be added to his credits. Customer has minimum like 3 days to pay the price through online mode (UPI, debit card etc).

Products will be managed in real time in dashboard if a person buys and owner can see his paid status.

This will also allow the owner to understand the costumers and their needs and help recognise loyal and recurring customers. They can then use this to give them special discounts and offers to reward their loyalty. This will also allow them to retain customers by giving them comeback offers.

The inventory management system will allow the owner to monitor the trends of sale and buy supplies accordingly.

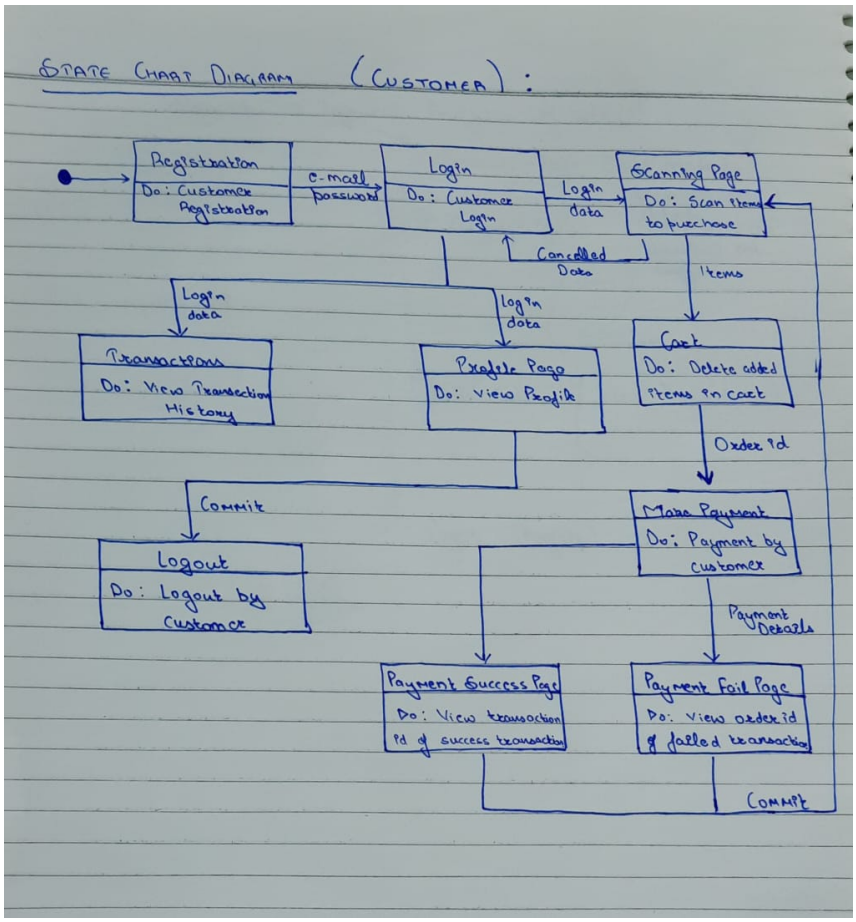
QR code will be secured through token authentication from API. So not anyone without login can buy the product.

Problem Statement:

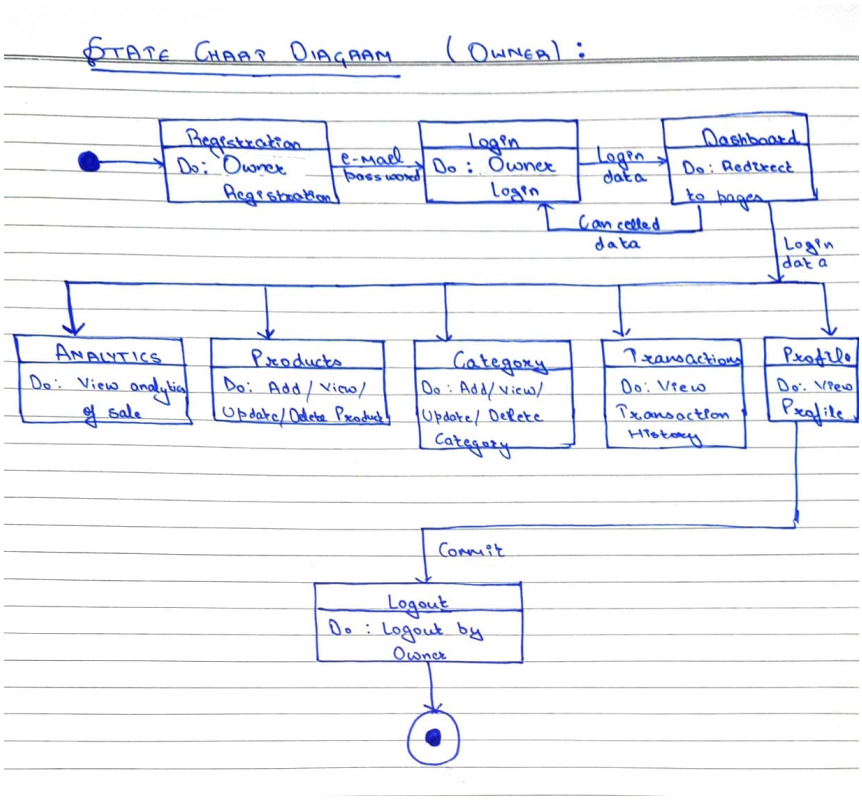
magine if you could shop at your local supermarket, without having to queue at checkouts to pay for your groceries. Dealers and small store owners don't have checkouts where a customer can buy. It is all managed and maintained by owner. So, we have thought of a solution to minimize the workload and man-work. This is useful for small scale business and local retail shops especially during the time of pandemic this will help in maintaining social distancing and contactless buying.

Process Flow Diagram

Customers



For owner-



Module Description:

1 For website

1.1) Login Page:

This login page is for the store administrator only. Its a simple form which validates the stored credentials from the database. The database also hashes and stores the password so the form sends credentials to backend and then hashes them and checks against the stores credentials.

1.2) Dashboard Page:

The dashboard page gives an insight to the store admin, the daily sale, the customers for the day, the possible profit,sale in different categories,etc. We have implemented role-based access control,thus only people with permissions can see certain pages.

1.3) Products Page:

This lets the admin search for a particular product and get its defined information like how many are left in the inventory and the current price of the product.

There is an option to create more products and thus while extending the product range in the store it could be added to the database as well.

1.4) Categories Page:

Lists out the different products in their respective categories for easy segregation. Also has the option to add newer products in different categories.

1.5)Profile Page

This is solely for the store admin to make their profiles and shop profiles on the website.

1.6) Transaction Page

Gives a detailed view of all transactions that takes place hence keeping in check of the accounts.

2 Mobile App

2.1) Login Screen

This screen is for the customers to login into the app. The login will be authenticated only when the customer is present in the store.

2.2) Qrcode Scanner

Used to scan the barcode on the products to scan the prices and to put it in the cart. Open source library pytesseract and pyzbar for optical character recognition and qrcode recognition.

2.3) Product details screen

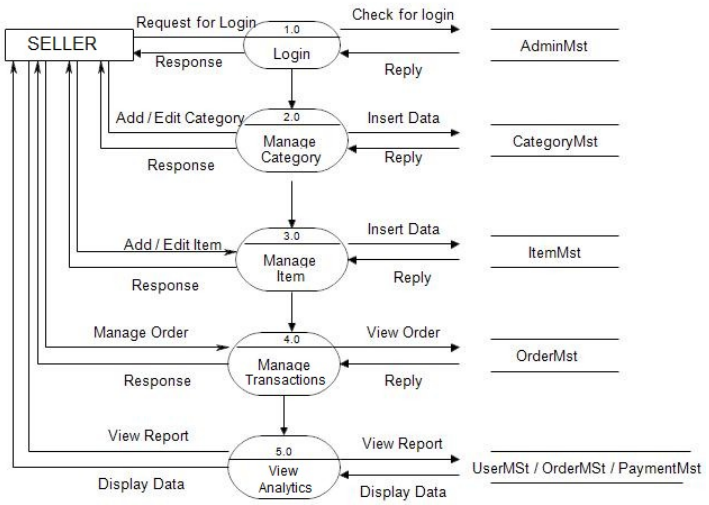
Right after scanning this screen pops up giving the product details.

2.4) Transaction Screen

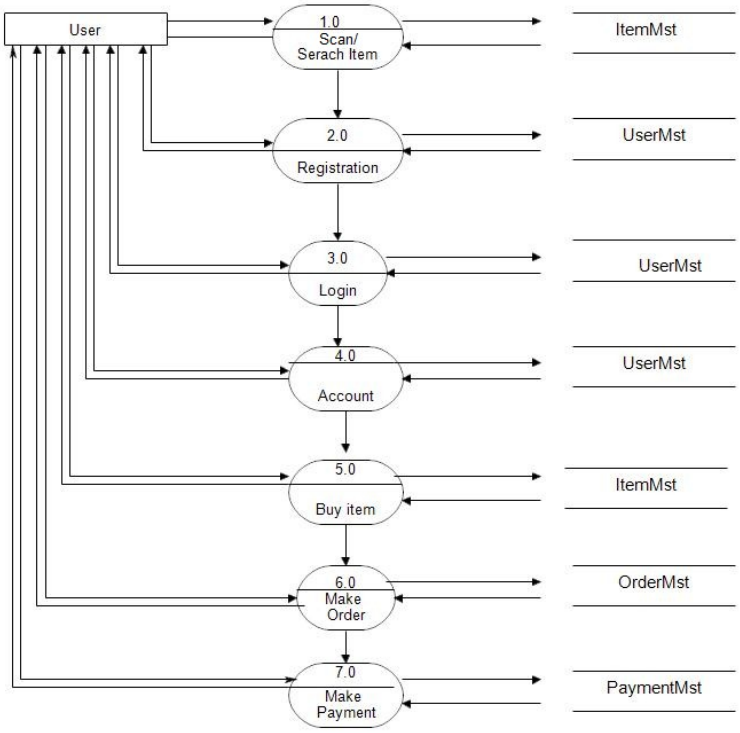
After buying the items which are in the cart. Checkout leads to payment portal (UPI) which gives a list of all transactions with seller information.

ER Diagrams-

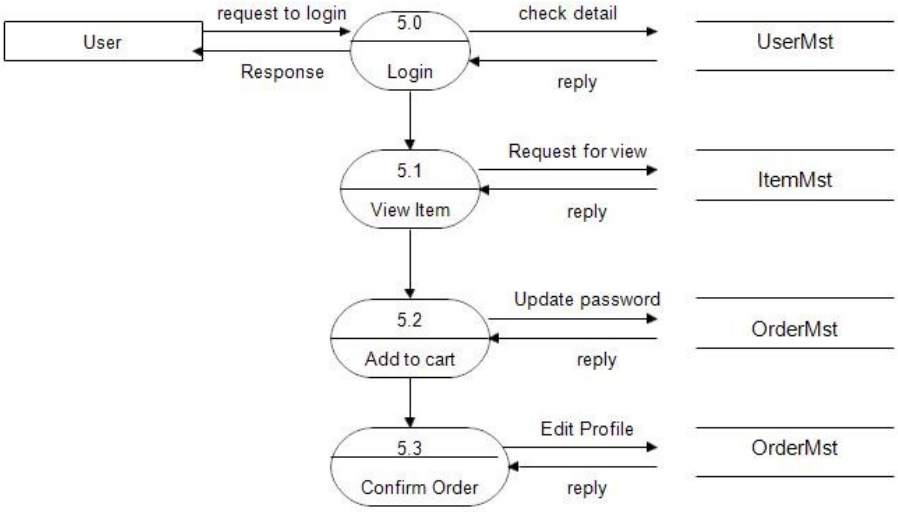
Seller Side DFD - 1st Level



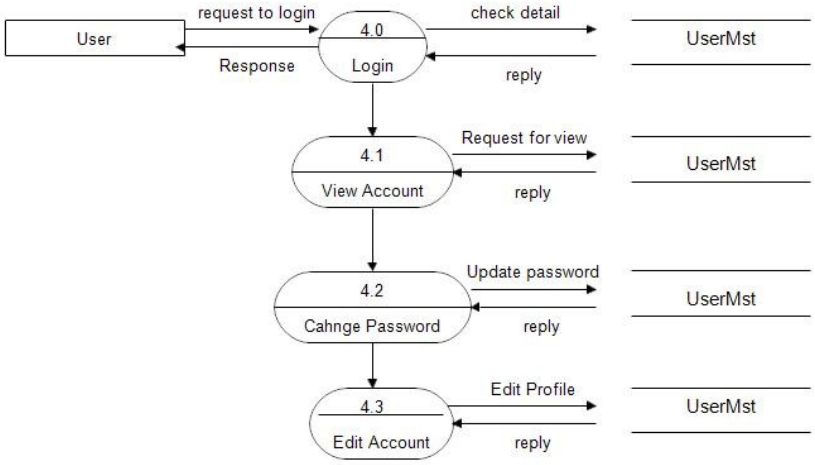
1st Level User side DFD



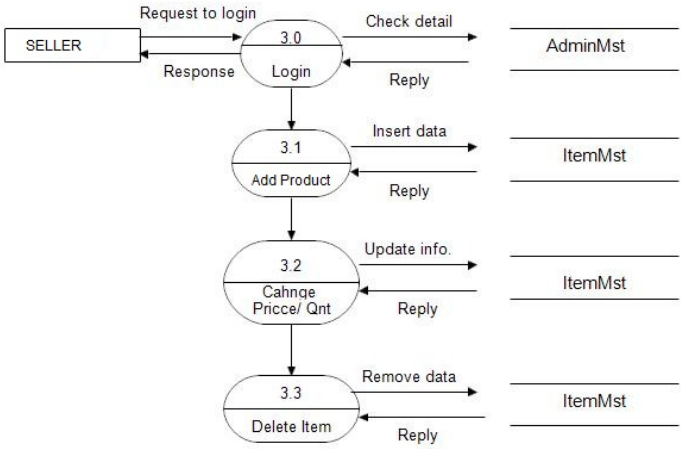
2st Level User DFD - (5.0)

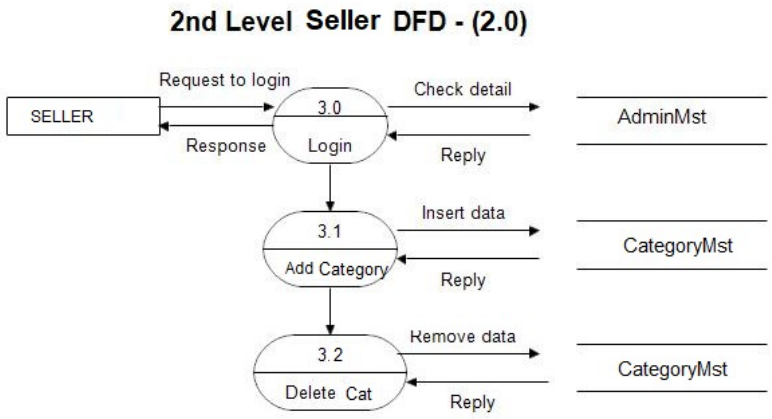


2st Level User DFD - (4.0)



2nd Level Seller DFD - (3.0)





Database Design:

1. **Users Table**

Name	Type	isUnique	isRequired	default
_id	ObjectID	true	true	
email	String	true	true	
name	String	false	true	
avatar	String	false	true	/public/defprofile.png
mobile	Number	true	true	
address	String	false	true	
shop	String	false	true	
password	String	false	true	

2. **Customer Table**

Name	Type	isUnique	isRequired	default
_id	ObjectID	true	true	
email	String	true	true	
name	String	false	true	

mobile	Number	true	true	
password	String	false	true	

3. **Categories Table**

Name	Type	isUnique	isRequired	default
_id	ObjectID	true	true	
owner	ObjectID (users)	true	true	
name	String	false	true	

4. **Products Table**

Name	Type	isUnique	isRequired	default
_id	ObjectID	true	true	
category	ObjectID (category)	true	true	
owner	ObjectID (users)	true	true	
name	String	false	true	
qrdata	String	true	true	
price	Number	true	true	
quantity	Number	true	true	

5. **Transaction Table**

Name	Type	isUnique	isRequired	default
_id	ObjectID	true	true	
buyerId	ObjectID (customer)	true	true	
sellerId	ObjectID (users)	true	true	
products	Array	false	true	
price	Number	false	true	

date	Date	false	true	

Conclusion:

The iterative life cycle model is followed as SDLC model as it starts by specifying and implementing a part of the software, which is then reviewed to identify further requirements or changes. Each new features like role based authentication, file upload, visualization and user management were implemented undergoing development, testing and optimization phase.

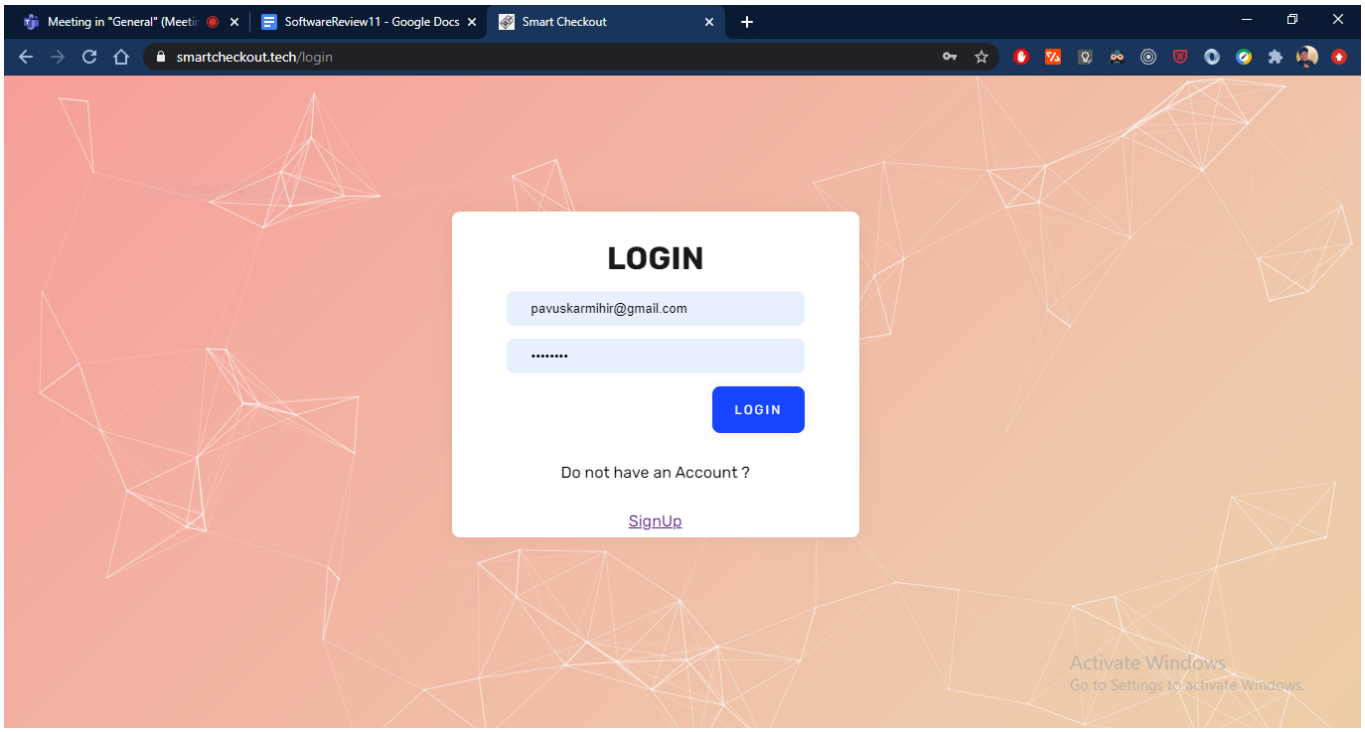
References:

1. https://docs.docker.com/
2. https://nodejs.org/en/docs/
3. https://reactjs.org/docs/getting-started.html
4. https://docs.flutter.dev/
5. https://developer.mozilla.org/en-US/docs/Web/HTML
6. https://developer.mozilla.org/en-US/docs/Web/CSS
7. https://developer.mozilla.org/en-US/docs/Web/JavaScript
8. https://developer.mozilla.org/en-US/docs/Web/API
9. https://razorpay.com/docs/
10. https://docs.mongodb.com/

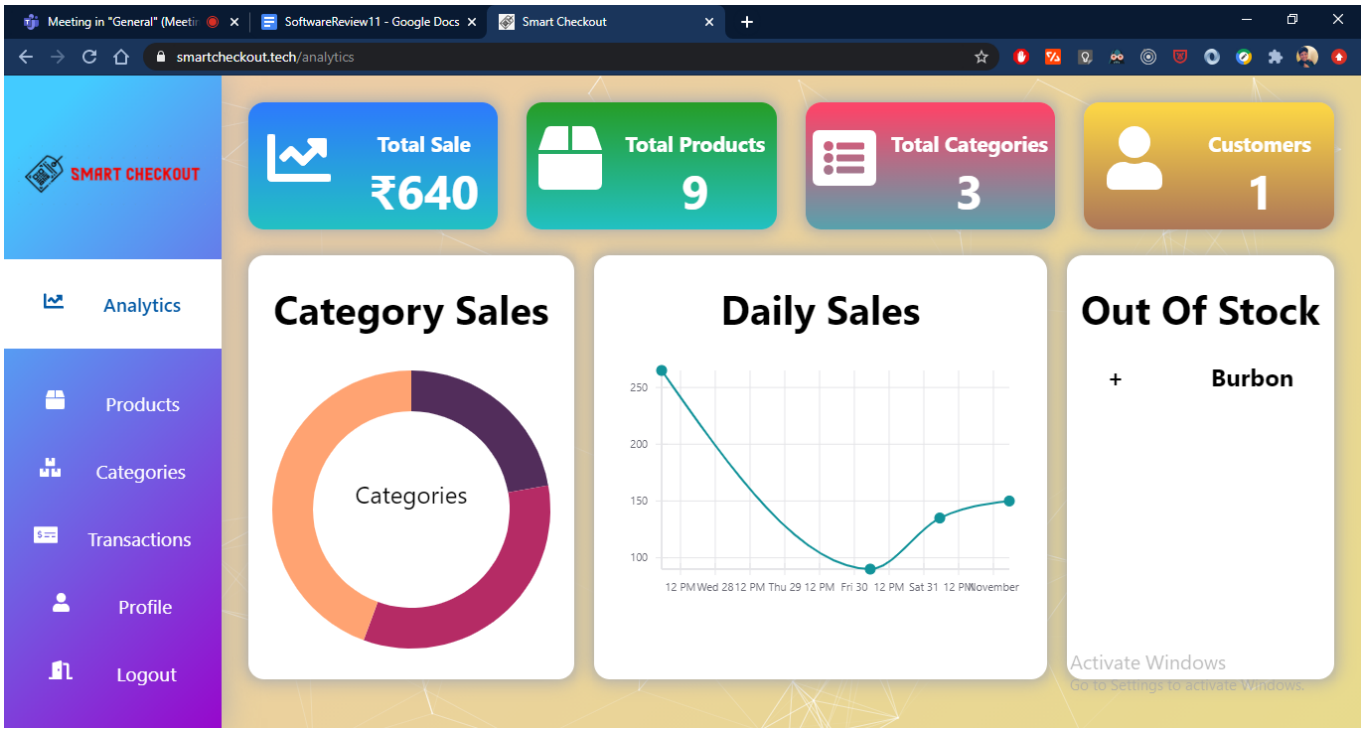
Snapshots :

Web Interface (Seller)

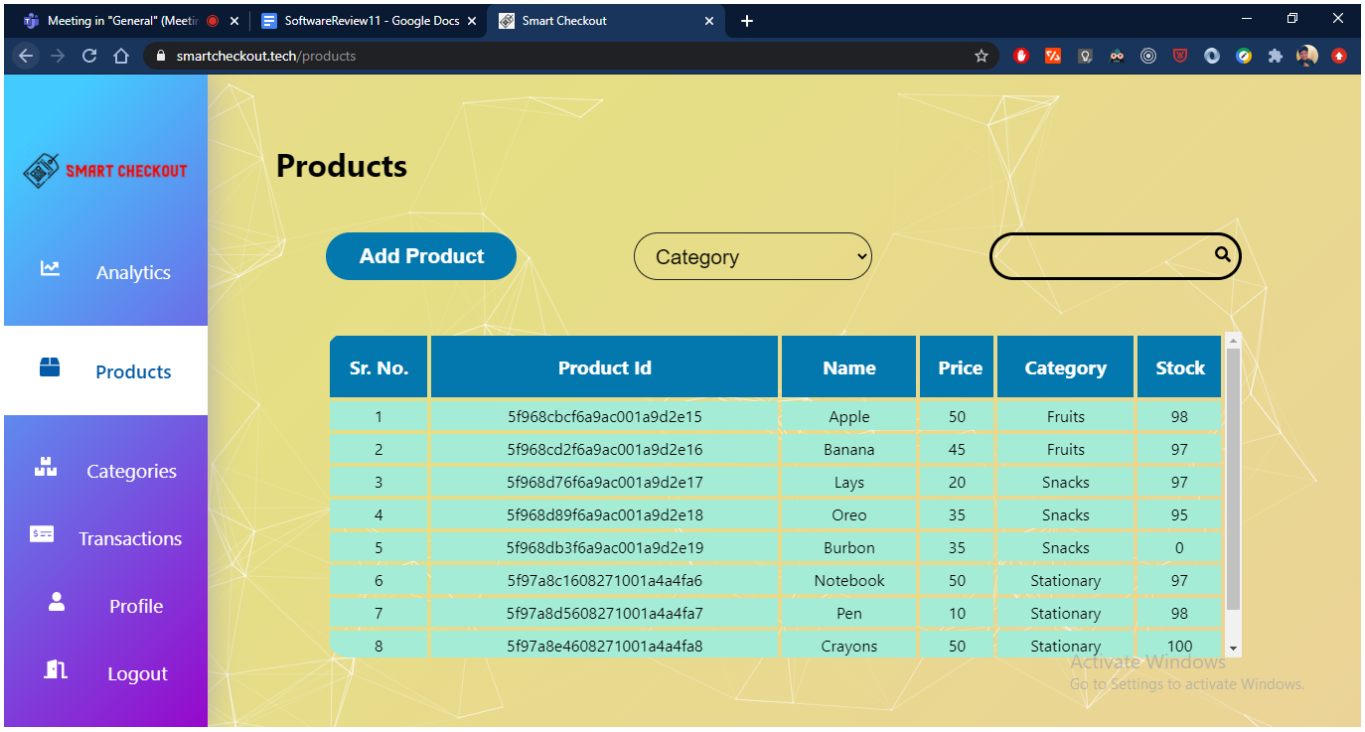
1. Login Page



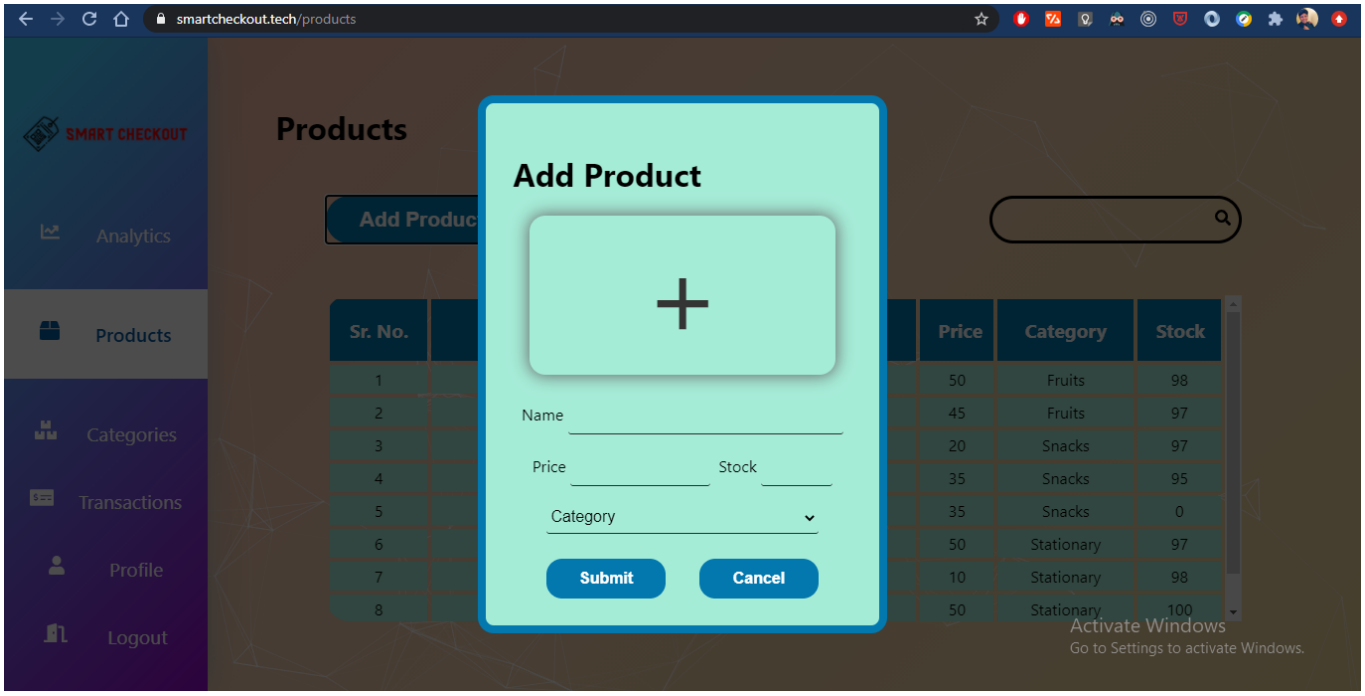
2. Dashboard



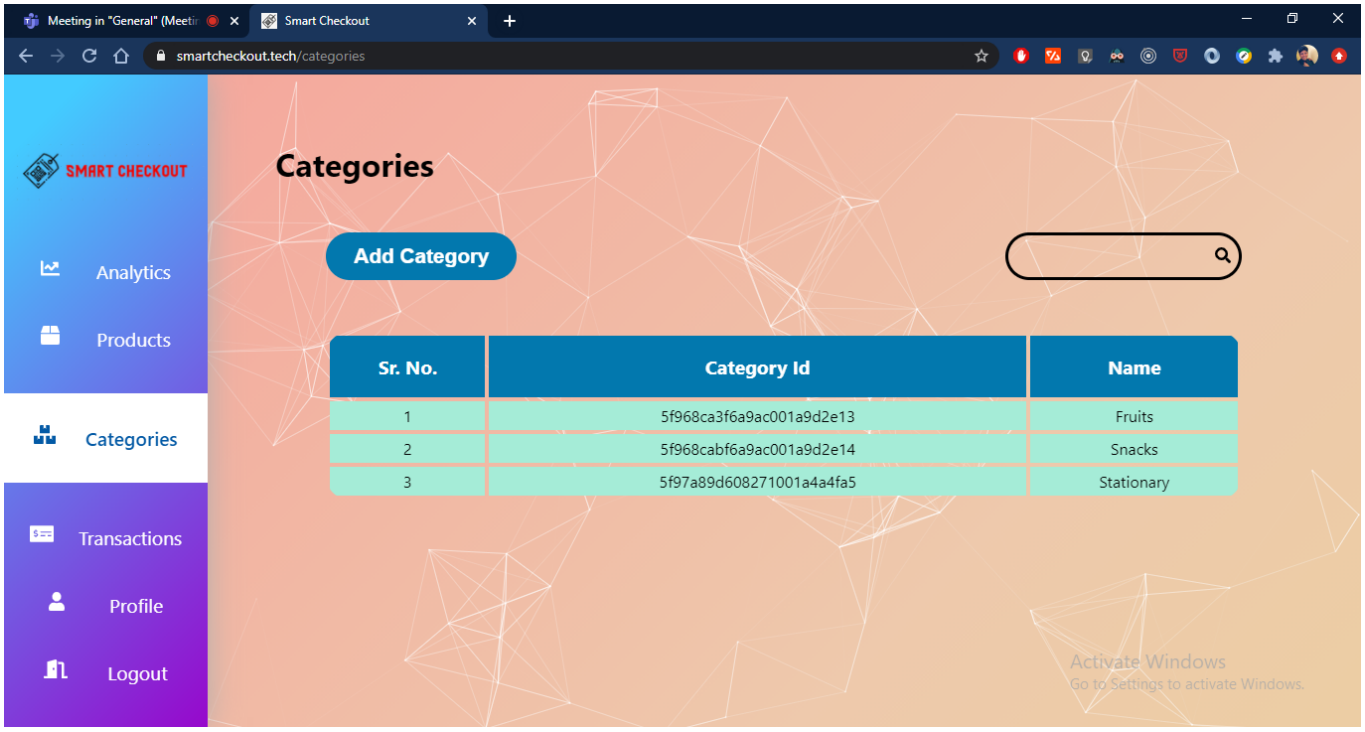
3. Products Page



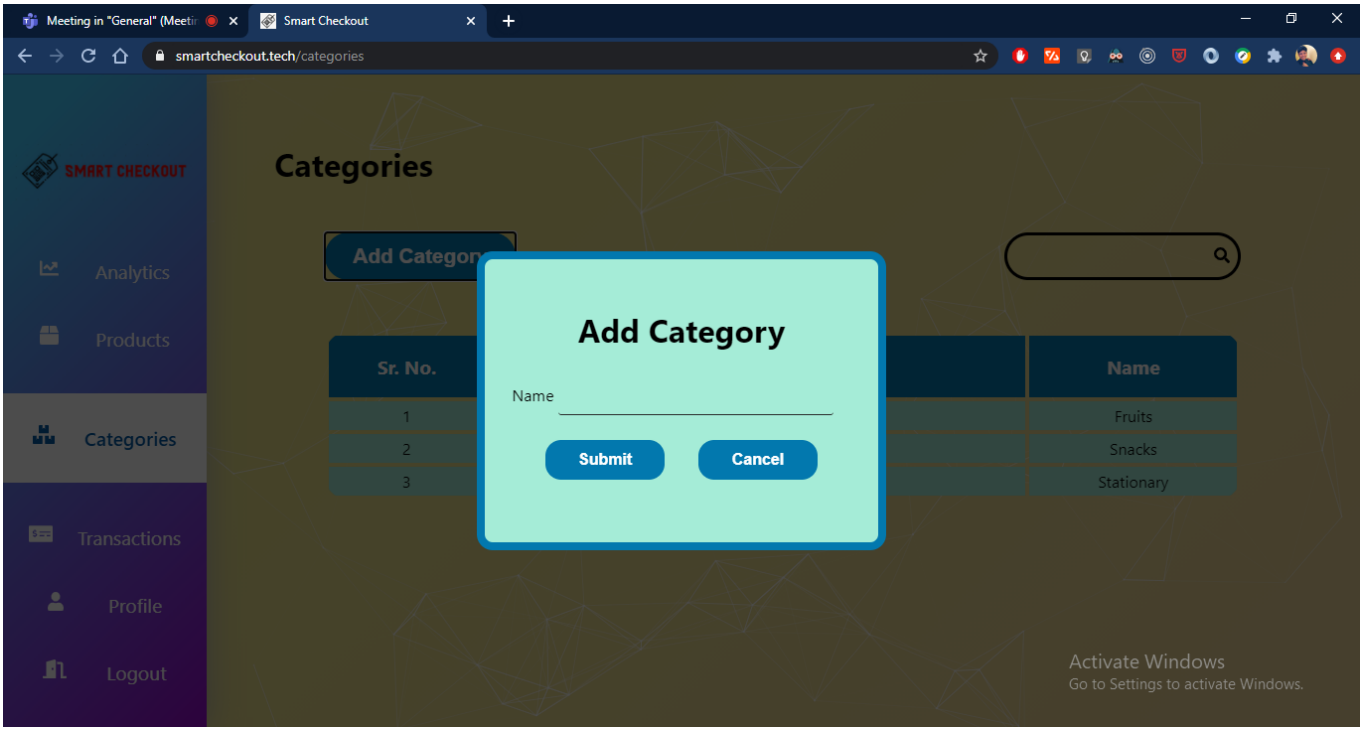
4. Create New Product Page



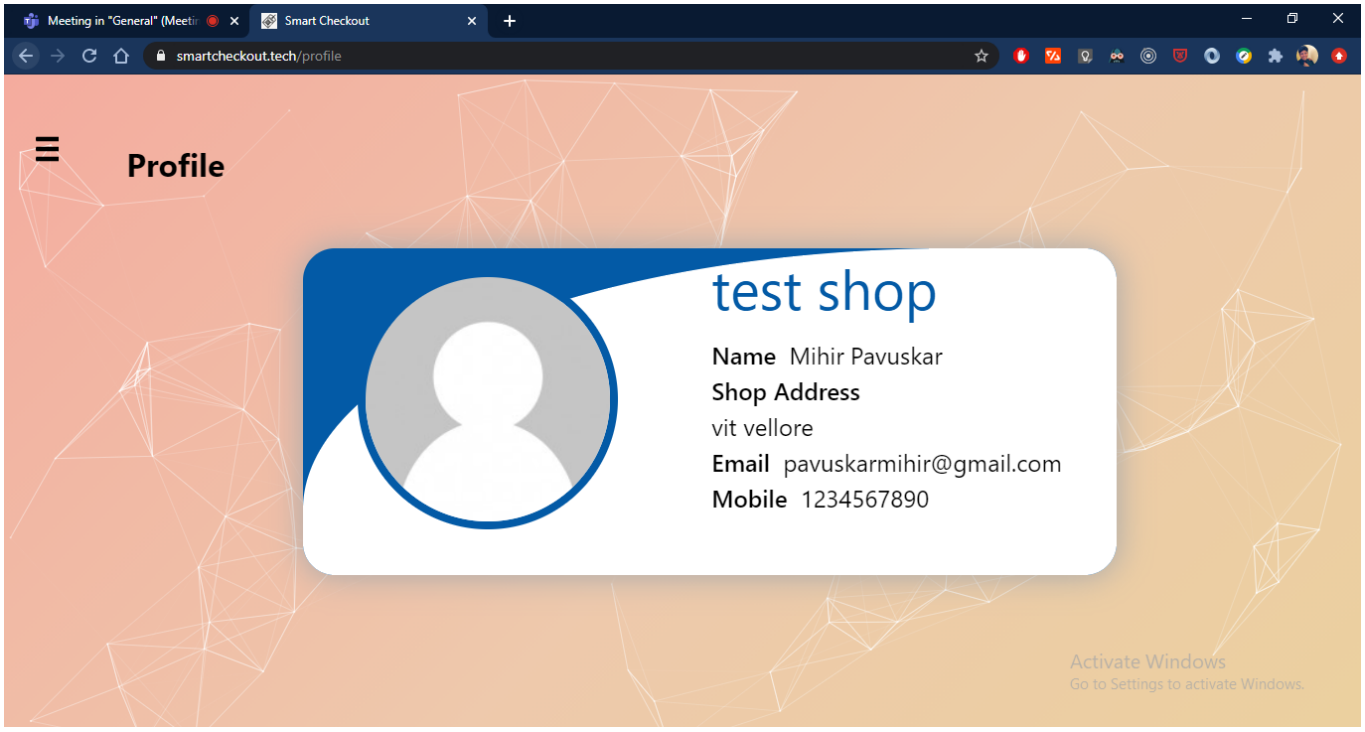
5. Categories Page



6. Create New Category Page

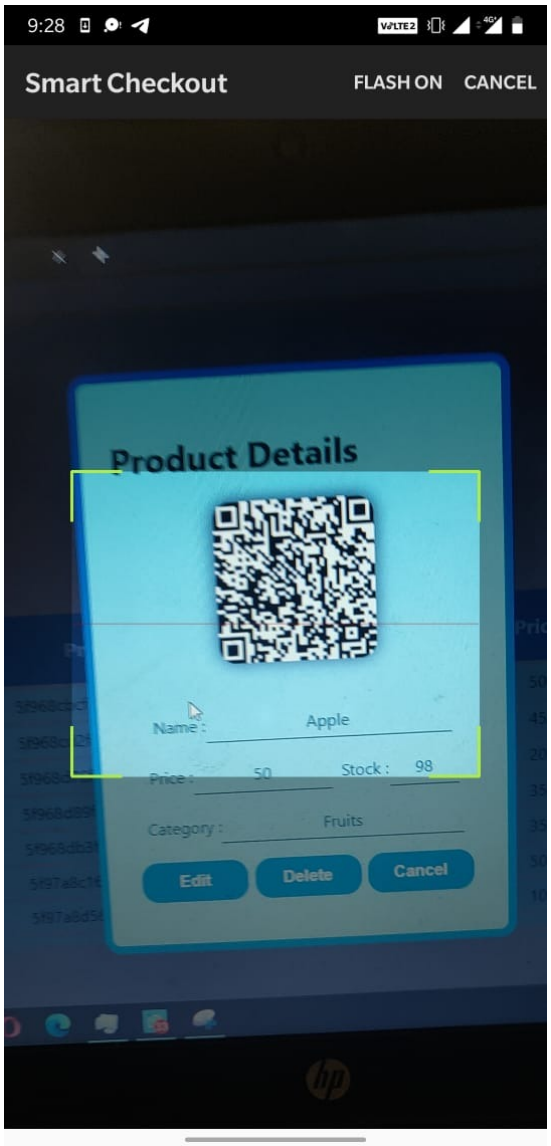


7. Profile Page

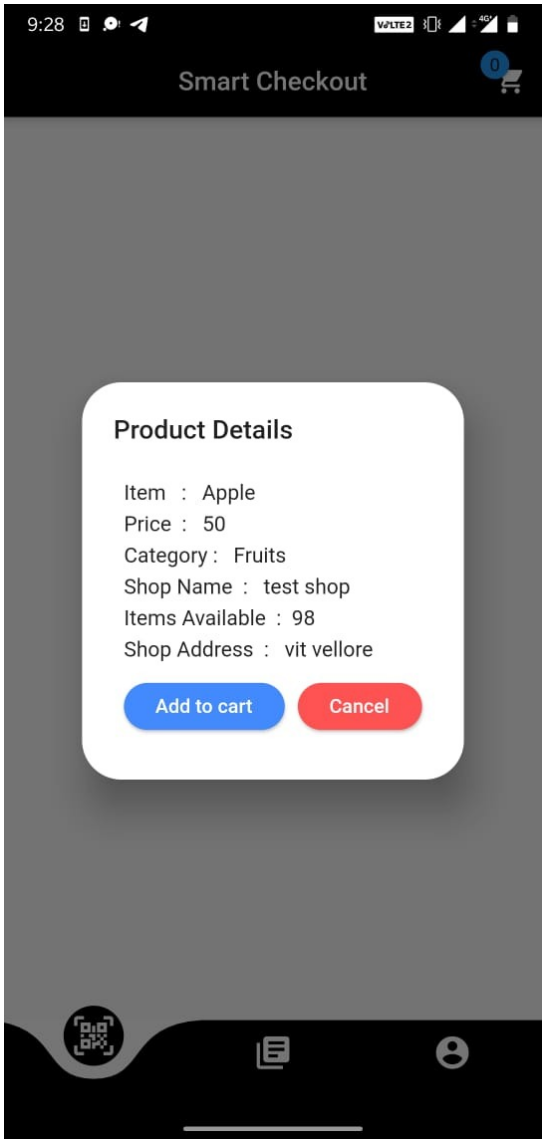


Mobile Interface (Buyer)

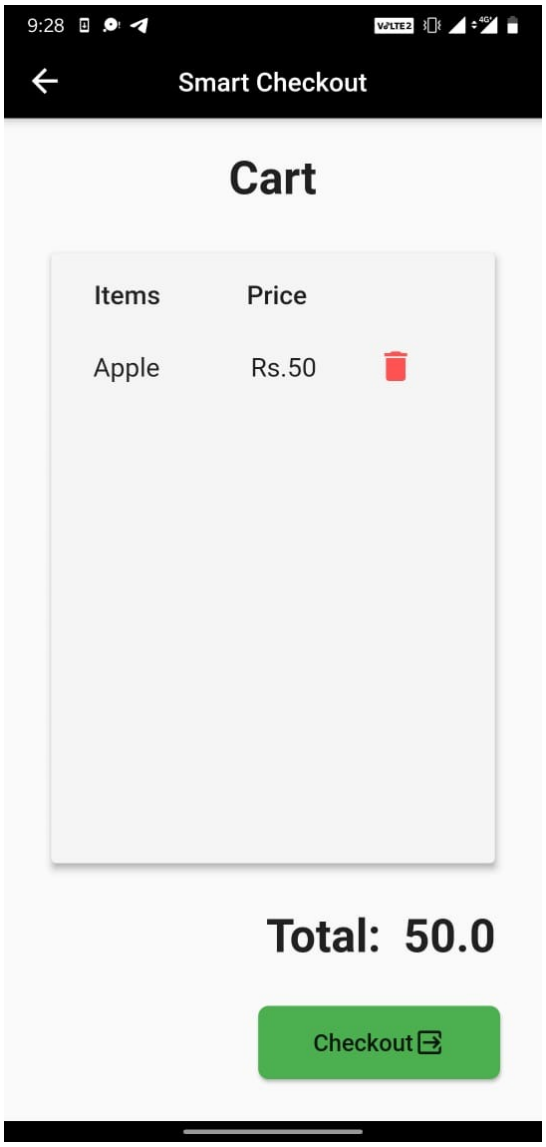
QrCode Scanner



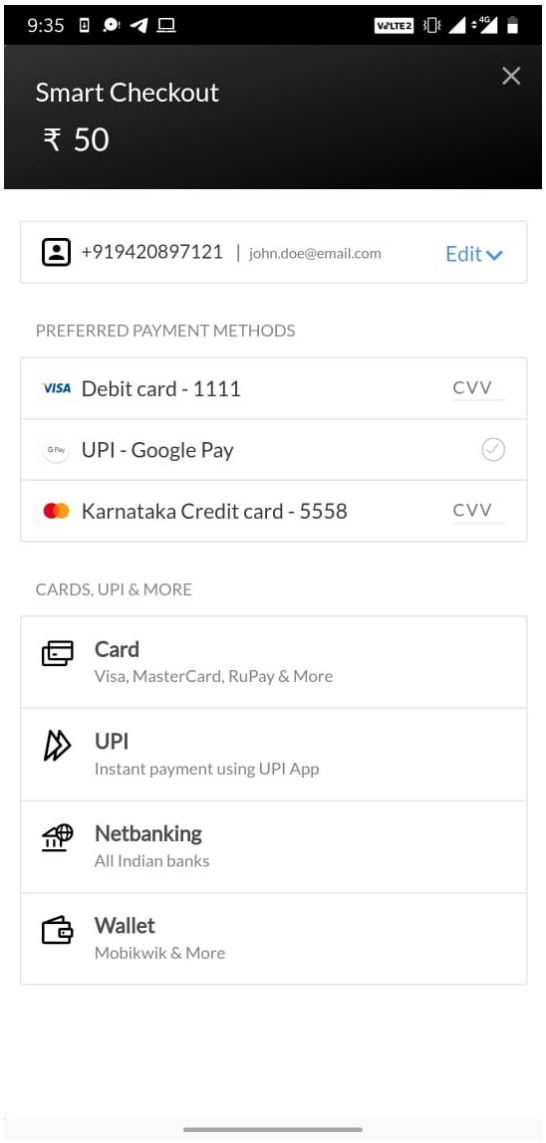
Add to Cart



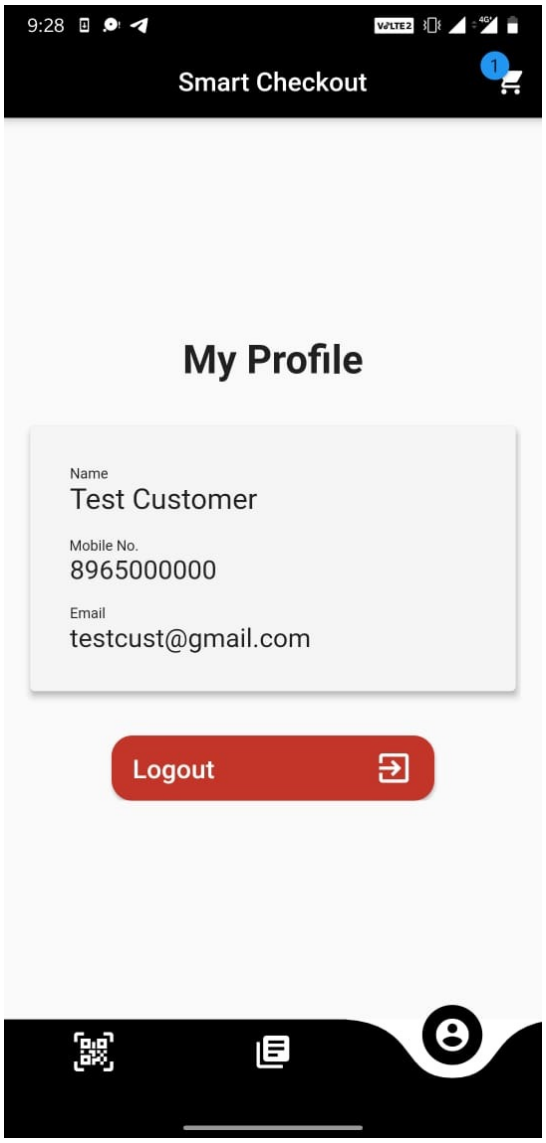
Cart



Checkout



Profile



Source Code:

https://github.com/MeMihir/SmartCheckout