Project Report

Android Application Development

Snack Squad: A Customizable Snack Ordering and Delivery App

Team ID: 593379

1. INTRODUCTION

1.1 Project Overview:

A project that demonstrates the use of Android Jetpack Compose to build a UI for a snack squad app. Snack Squad is a sample project built using the Android Compose UI toolkit. It demonstrates how to create a simple e-commerce app for snacks using the Compose libraries. The user can see a list of snacks, and by tapping on a snack, and by tapping on the "Add to Cart" button, the snack will be added to the cart. The user can also see the list of items in the cart and can proceed to checkout to make the purchase.

1.2 Purpose:

Building a snack ordering app in Kotlin serves the fundamental purpose of providing users with a streamlined and convenient means to order snacks, fostering efficiency and user satisfaction. Kotlin's versatility in Android development allows for the implementation of essential features such as menu browsing, order customization, secure payment options, and real-time order tracking. Beyond the basics, the app enhances user experience through personalization, leveraging Kotlin's capabilities to offer tailored recommendations based on user preferences. The language's support for secure transactions ensures the protection of user financial information during payments. Additionally, Kotlin facilitates the integration of real-time updates and notifications, keeping users informed about order status and promotions. The app can capitalize on Kotlin's scalability and cross-platform compatibility, reaching a wider audience by functioning seamlessly on both Android and iOS devices. Developers can harness Kotlin's analytical capabilities to gain insights into user behaviour and preferences, contributing to informed decision-making for business growth. Furthermore, the language's vibrant community support provides resources and assistance throughout the development process. By utilizing Kotlin's expressive syntax, developers can introduce innovative features, such as augmented reality or smart device integration, elevating the snack ordering experience. Overall, building a snack ordering app in Kotlin ensures a robust, scalable, and feature-rich application that caters to user needs while fostering a dynamic and engaging platform.

2. LITERATURE SURVEY

A literature survey for your snack ordering and delivery app could encompass various aspects related to mobile applications, food delivery, and user experience. Begin by examining existing literature on mobile app development, focusing on the latest trends and best practices in UI/UX design, user engagement, and performance optimization. Explore studies and articles related to the specific challenges and opportunities within the food delivery app domain, such as order processing, real-time tracking, and payment gateways. Additionally, investigate customer preferences and behaviours in the context of food ordering through mobile apps.

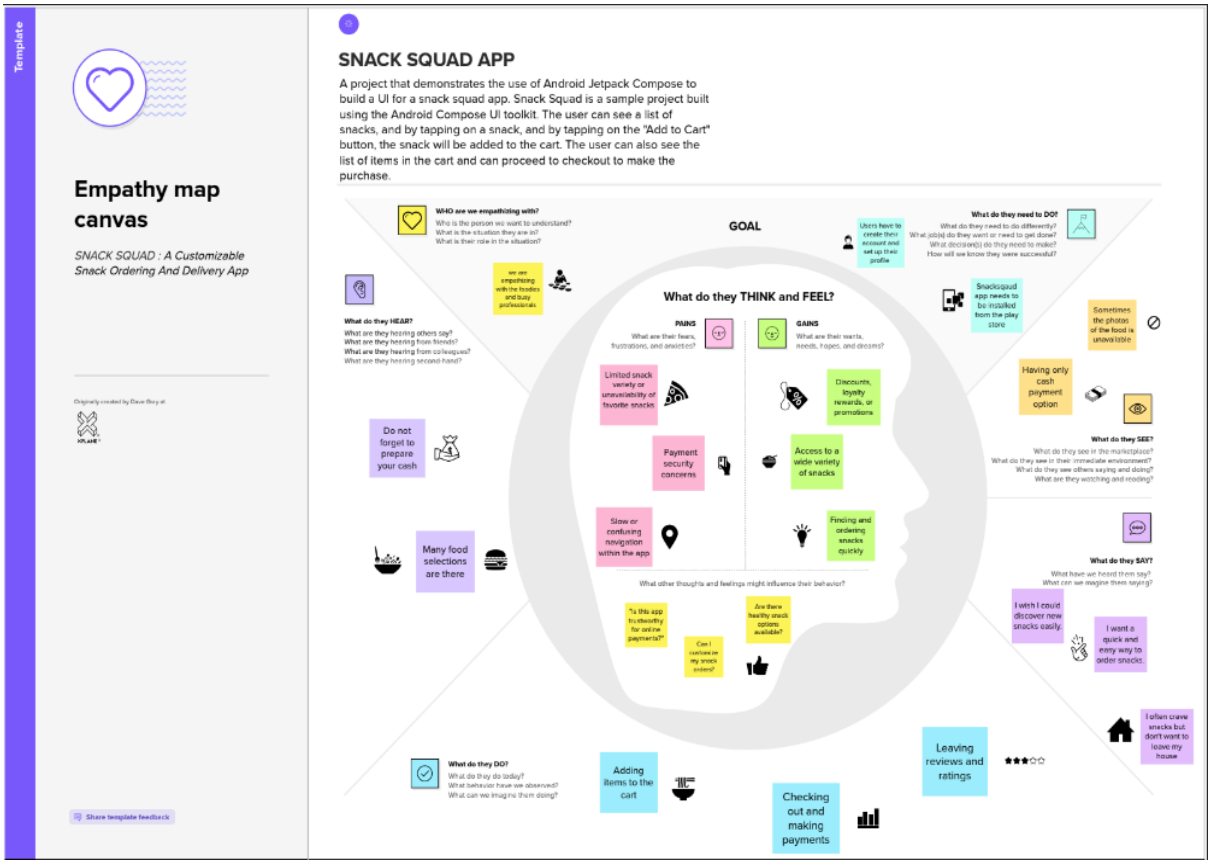
Analysing successful case studies and identifying key features that contribute to user satisfaction in similar applications can inform the design and functionality of your snack delivery app. Keep an eye on emerging technologies or innovations in the mobile app and food delivery industries that could potentially enhance the user experience and set your app apart. The literature survey should not only guide the development process but also provide insights into user expectations and industry benchmarks.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



Link:
https://drive.google.com/file/d/1fvOEVkuWFk9S_QkybMY1F8CJSdQkmXes/view?usp=sharing

3.2 Ideation & Brainstorming

Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference:
https://app.mural.co/t/vitap1976/m/vitap1976/1696925266353/5573bb559af3af7f09486f76198151ee0a201c33?sender=uf38bfaea0a8ccb75f4712375

DRIVE LINK:
https://drive.google.com/file/d/1RZB1rpi78r3IxD9Yj7iJnIg7_84TjtBj/view?usp=sharing

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Template

Brainstorm & idea prioritization

SNACK SQUAD : A Customizable Snack Ordering And Delivery App

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Vanaparthi Bulli Siva
Pasala Maheshwar Reddy
Makireddy Nanaji
Karimilla Yogeshwar Rao

B Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

1. Generating numerous ideas on snack squad app designing
2. Prioritizing and filtering ideas and choose the best ideas
3. Focusing on jetpack compose in Android
4. Making the documents and uploading in Git
5. Design the app

C Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

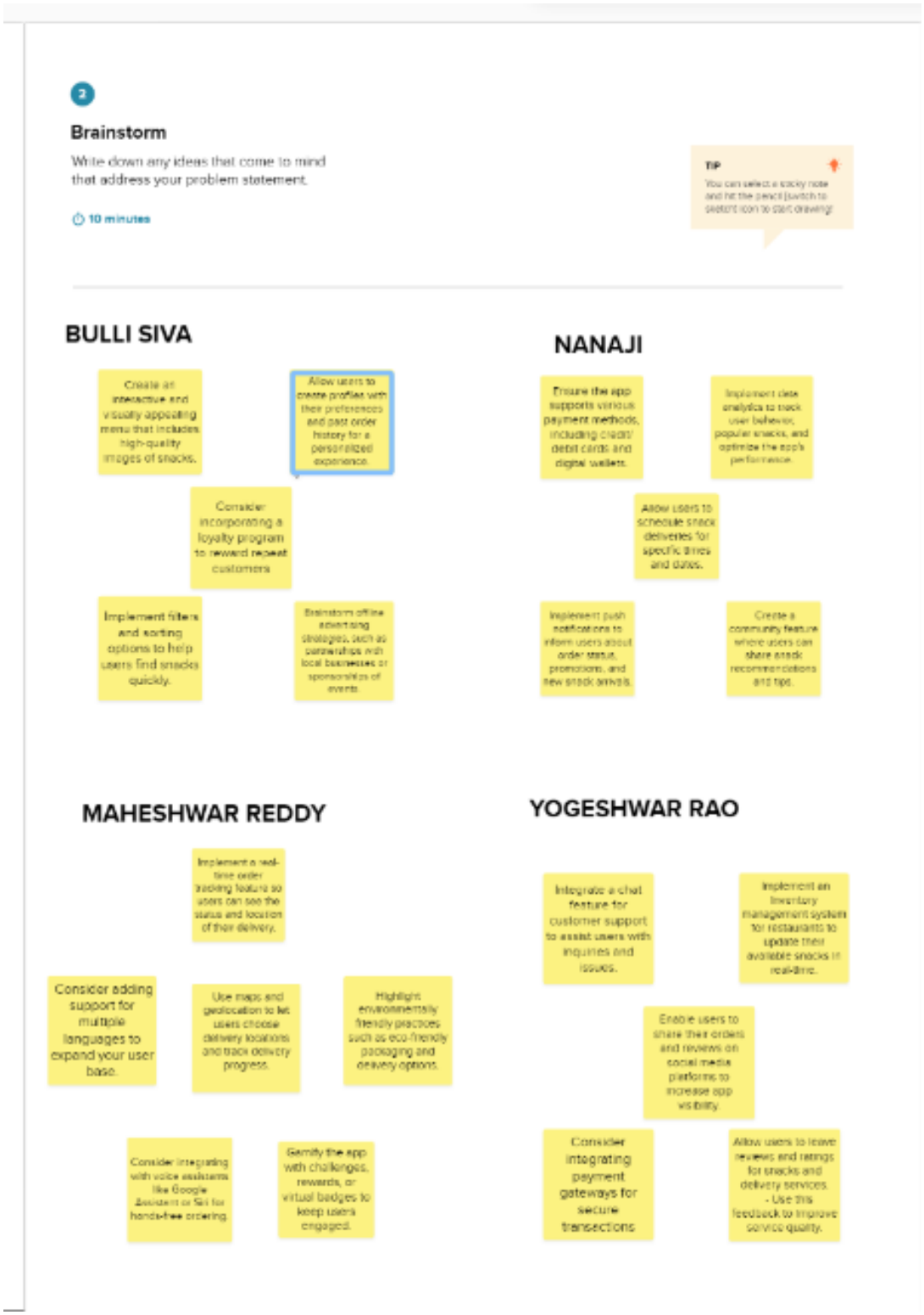
How might we Design a customer-centric snack ordering and delivery app using jetpack compose in kotlin Android?

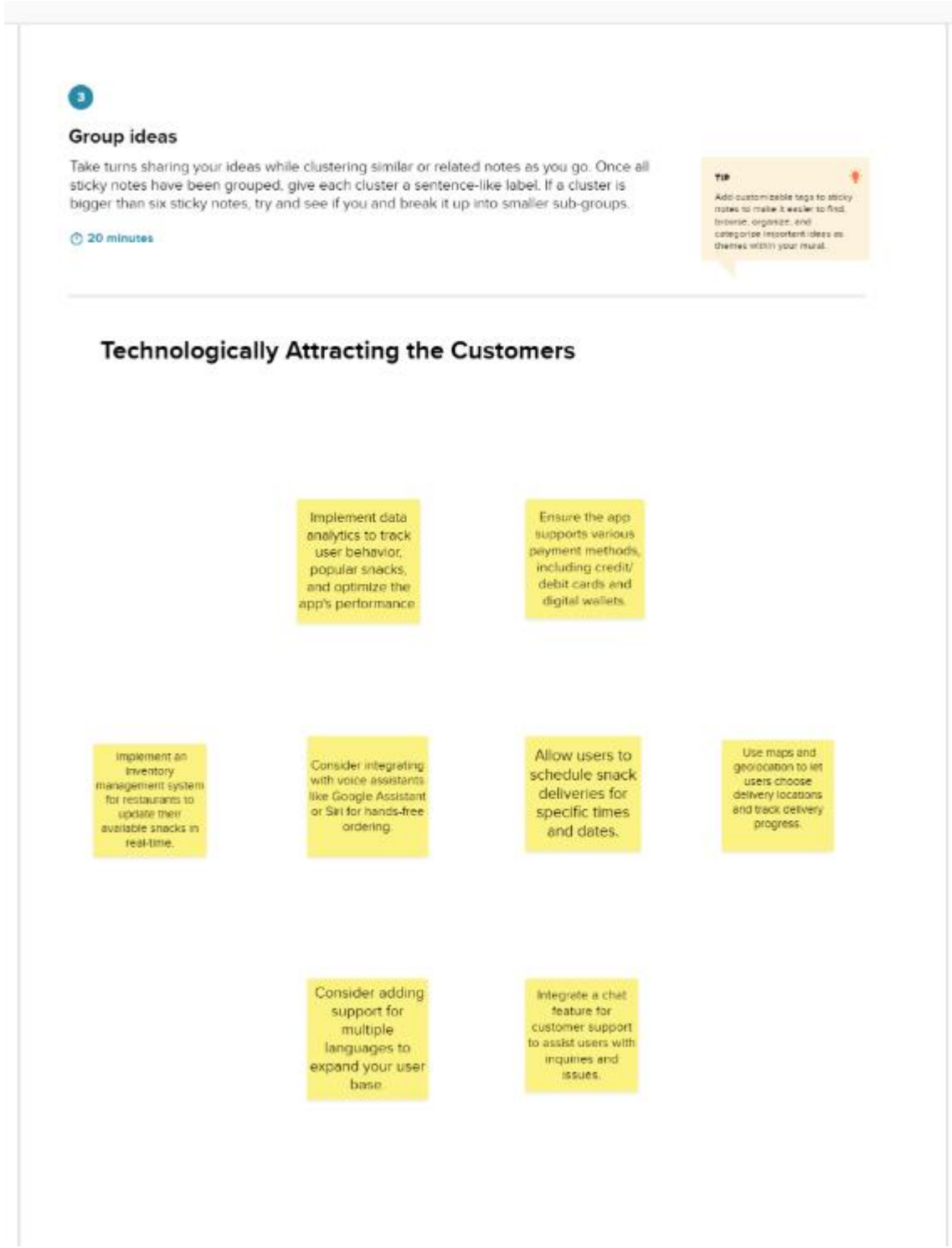
Key rules of brainstorming

To run an smooth and productive session

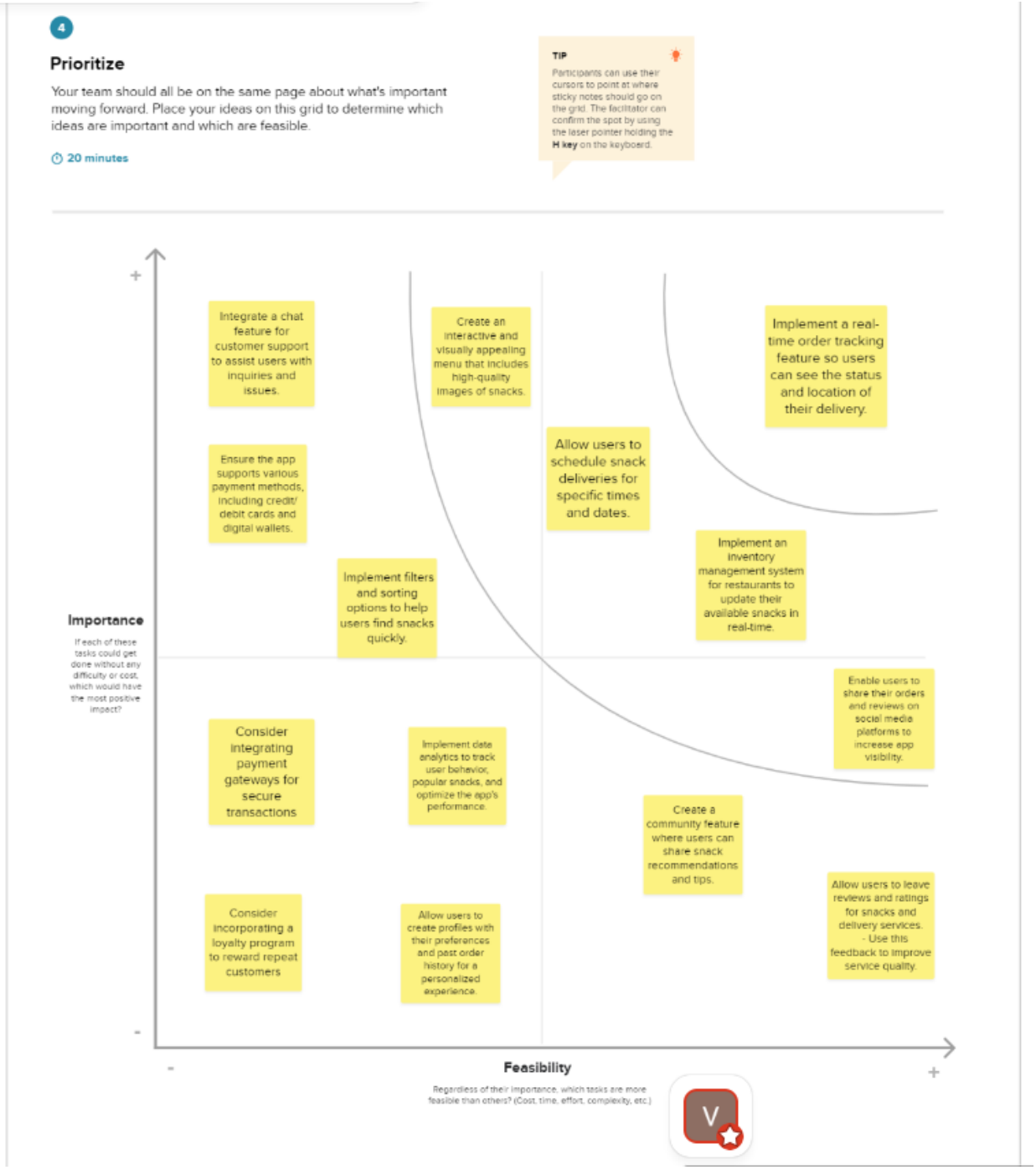
- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

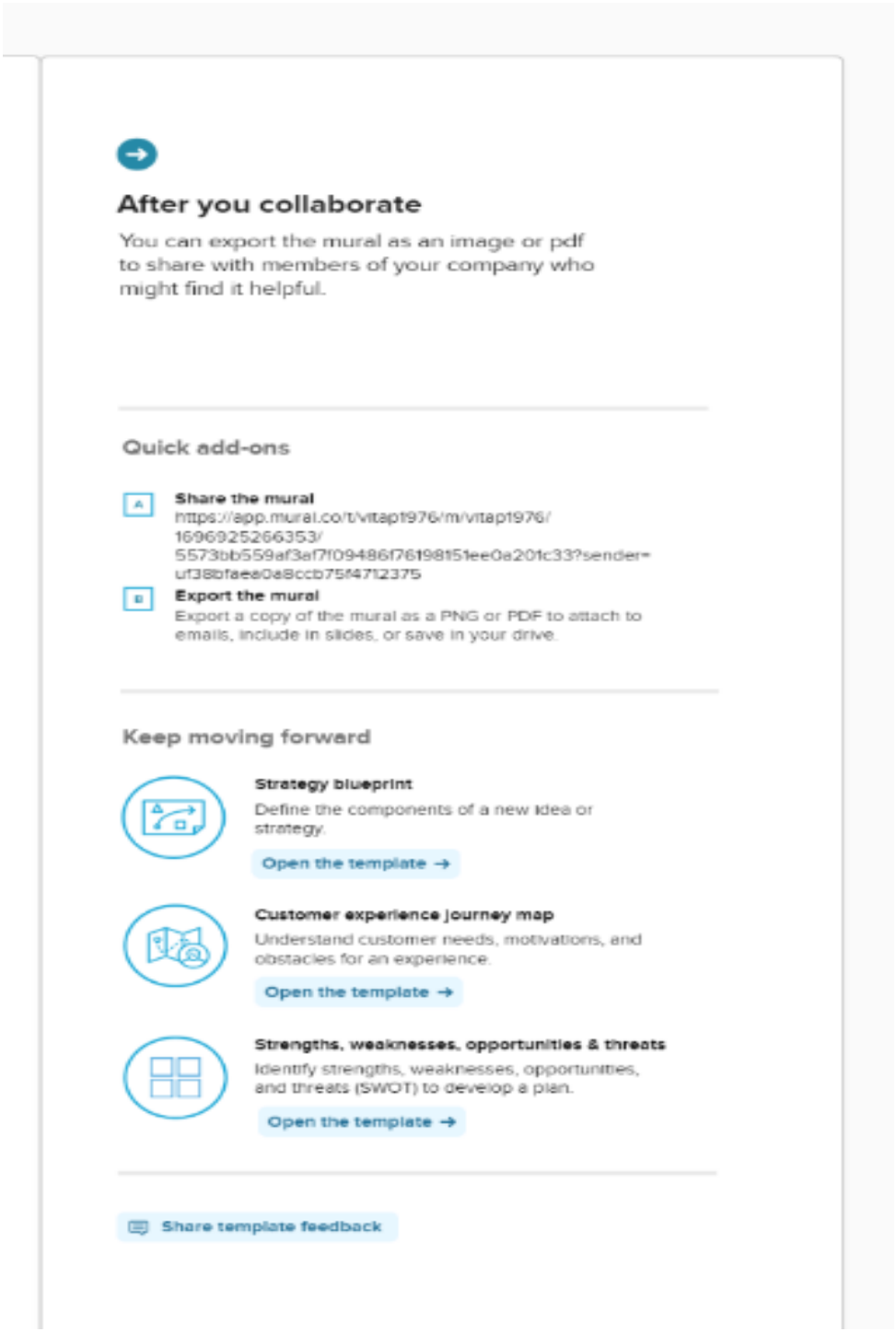
Step-2: Brainstorm, Idea Listing and Grouping




3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
Technologically Attracting the Customers
Implement data analytics to track user behavior, popular snacks, and optimize the app's performance
Ensure the app supports various payment methods, including credit/ debit cards and digital wallets.
Implement an inventory management system for restaurants to update their available snacks in real-time.
Consider integrating with voice assistants like Google Assistant or Siri for hands-free ordering.
Allow users to schedule snack deliveries for specific times and dates.
Use maps and geolocation to let users choose delivery locations and track delivery progress.
Consider adding support for multiple languages to expand your user base
Integrate a chat feature for customer support to assist users with inquiries and issues.

Step-3: Idea Prioritization




After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.
Quick add-ons
A
Share the mural
https://app.mural.co/t/vitap1976/m/vitap1976/1696925266353/5573bb559af3af7f09486f76198151ee0a201c33?sender=uf38bfaee0a8ccb75f4712375
B
Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.
Keep moving forward
Strategy blueprint
Define the components of a new idea or strategy.
Open the template →
Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →
Share template feedback

4. REQUIREMENT ANALYSIS:

4.1 Functional Requirement:

1. User Registration and Authentication: Users should be able to create accounts, log in securely, and maintain profile information.

2. Menu Browsing: The app should allow users to browse a digital menu with detailed information about snacks, including prices and ingredients.

3. Order Customization: Users should be able to customize their snack orders, selecting preferences like toppings, sizes, and quantities.

4. Shopping Cart: The app should include a shopping cart functionality where users can review their selected items, modify quantities, and proceed to checkout.

5. Secure Payment: Implement a secure payment gateway to facilitate online transactions, ensuring the confidentiality of users' financial information.

6. Order Tracking: Provide users with real-time updates on the status of their orders, including confirmation, preparation, and delivery tracking.

7. User Feedback and Ratings: Allow users to provide feedback and ratings for snacks they have ordered, contributing to a dynamic user-driven platform.

8. User Support: Include a support system, such as a chat or helpdesk, to assist users with queries, issues, or assistance during the ordering process.

9. Promotions and Discounts: Implement features to showcase promotions, discounts, and special offers to incentivize users and boost engagement.

10. Cross-Platform Compatibility: Ensure the app works seamlessly on both Android and iOS platforms, reaching a broader user base.

4.2 Non Functional Requirements

1. Performance: The app should have fast response times, even during peak usage, to provide a smooth and responsive user experience.

2. Scalability: The system should be able to handle a growing user base and an increasing number of transactions without a significant decrease in performance.

3. Security: Ensure robust security measures to protect user data, including encryption of sensitive information, secure authentication processes, and secure connections for transactions.

4. Reliability: The app should be reliable, minimizing downtime and ensuring that users can consistently access and use the platform.

5. Usability: Design the user interface with a focus on usability, making it intuitive and easy for users to navigate through the app and complete their orders.

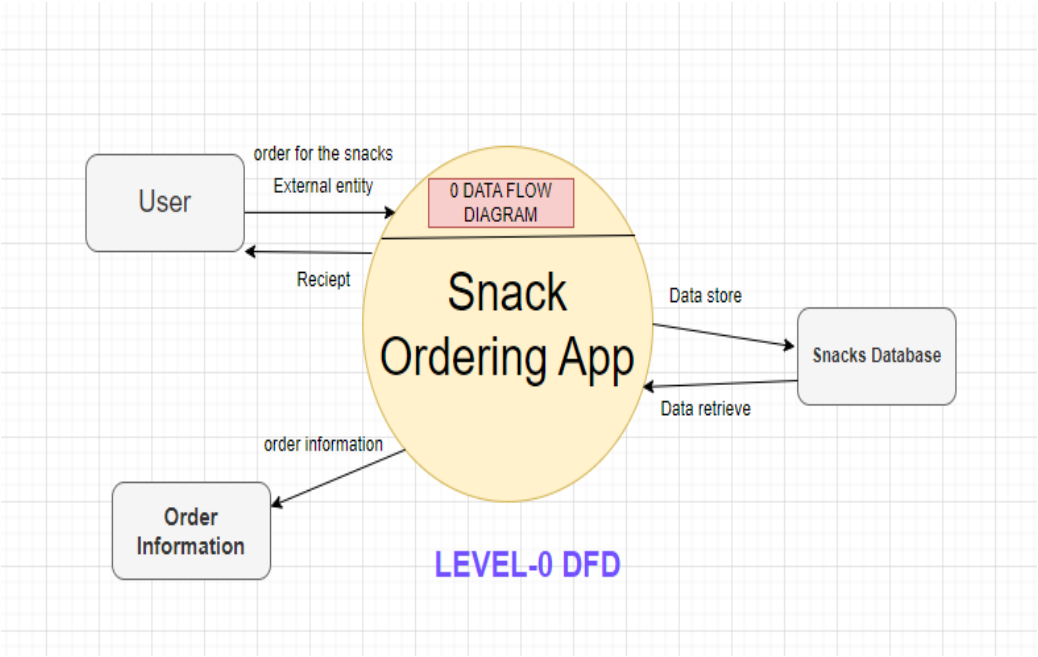
6. Compatibility: Ensure compatibility with a variety of devices and screen sizes to accommodate different user preferences.

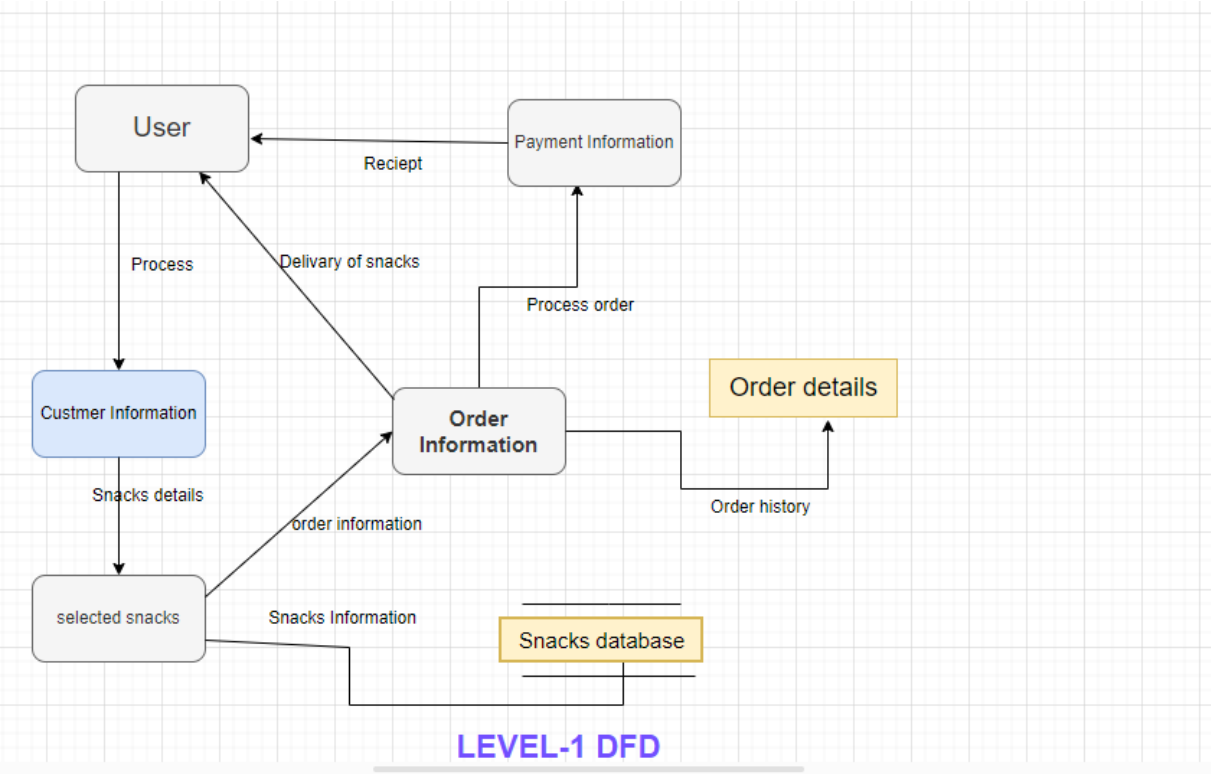
7. Maintainability: Develop the app in a way that facilitates easy maintenance and updates, allowing for the addition of new features and the resolution of any issues that may arise.

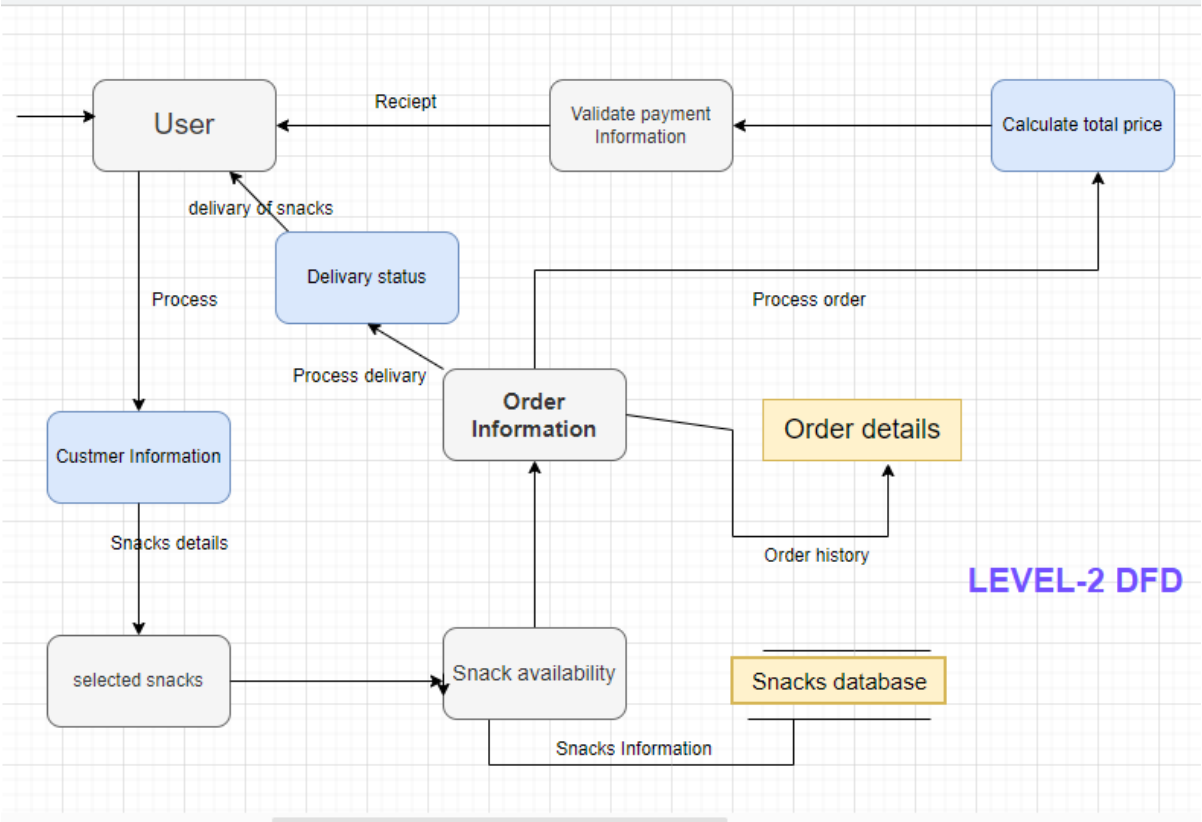
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

Data Flow Diagrams (DFDs) are a visual representation of how data moves and is processed within a system. They are a powerful tool for system analysis, design, and documentation. DFDs use standardized symbols and notation to depict the flow of data and the processes that act on that data. Here's a more detailed description of DFD components and their roles.




User
Payment Information
Reciept
Process
Delivary of snacks
Process order
Custmer Information
Order Information
Order details
Snacks details
Order history
order information
selected snacks
Snacks Information
Snacks database
LEVEL-1 DFD


User
Reciept
Validate payment Information
Calculate total price
delivary of snacks
Delivary status
Process
Process order
Process delivary
Order Information
Order details
Custmer Information
Snacks details
Order history
LEVEL-2 DFD
selected snacks
Snack availability
Snacks database
Snacks Information

User stories

User Type	User Story Number	User Story / Task	Functional Requirement (Epic)	Acceptance Criteria	Priority	Release
Customer	USN-1	As a customer, I want to browse snacks available for ordering.	Snack Selection	I can view a list of available snacks.	High	Sprint-1
Customer	USN-2	As a customer, I want to add snacks to my cart.	Snack Selection	I can add snacks to my shopping cart.	High	Sprint-1
Customer	USN-3	As a customer, I want to specify snack quantity and options.	Snack Selection	I can adjust snack quantities and select options (e.g., size, toppings).	High	Sprint-1
Customer	USN-4	As a customer, I want to review my order before checkout.	Order Placement	I can see a summary of my selected snacks and their total price.	High	Sprint-1
Customer	USN-5	As a customer, I want to place an order.	Order Placement	I can confirm my order and receive an order confirmation.	High	Sprint-1

Customer	USN-6	As a customer, I want to track my order's status.	Order Tracking	I can check the status of my order (e.g., preparing, out for delivery).	Medium	Sprint- 2
Customer	USN-7	As a customer, I want to view my order history.	Order History	I can access a list of my past orders.	Medium	Sprint- 2
Customer	USN-8	As a customer, I want to customize my user profile.	User Profile	I can update my personal information (e.g., name, address).	Medium	Sprint- 2
Customer	USN-9	As a customer, I want to provide feedback on my orders.	Feedback	I can submit feedback and ratings for the snacks and service.	Low	Sprint- 3
Customer	USN-10	As a customer, I want to reset my password if forgotten.	Account Management	I can initiate a password reset and receive an email with instructions.	Low	Sprint- 3

5.2 Solution Architecture:

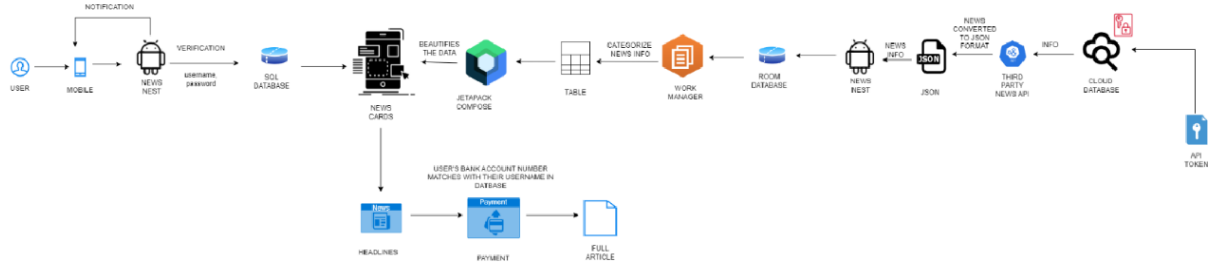
Proposed solution

A proposed solution, in the context of a project or problem-solving, refers to a suggested approach or plan to address a specific issue or achieve a particular objective. It outlines how we intend to tackle a problem or meet a set of requirements. The proposed solution is a well-thought-out strategy that lays out the steps, actions, or methods we will use to resolve the problem or achieve our project's goals.

S.NO.	PARAMETER	DESCRIPTION
1	**Problem statement**	The problem statement outlines the existing issues in the snacks ordering process. Traditional methods are often inefficient, leading to long wait times and limited menu options for customers. There's a clear need for a solution that simplifies the snacks ordering process, making it more convenient and user-friendly.
2	**Idea / Solution description**	This parameter explains the core concept of our snacks ordering Android app. The idea is to create a mobile application that allows users to easily browse through a diverse selection of snacks, place orders with customization options, and make secure payments through the app. Key features include menu exploration, order personalization, real-time order tracking, and support for various payment methods. The aim is to provide a hassle-free and enjoyable snacks ordering experience for users.
3	Novelty / Uniqueness	The uniqueness of our app lies in its innovative features. It includes a personalized recommendation engine that suggests snacks based on user preferences and dietary restrictions. Moreover, the app integrates with loyalty programs to reward frequent customers. These distinctive features set our app apart from existing solutions, offering a more tailored and rewarding snacks ordering experience.

4	Social Impact / Customer Satisfaction	Our app aims to have a positive social impact by reducing wait times, providing personalized recommendations, and offering a user-friendly interface. This, in turn, enhances customer satisfaction. Additionally, the app supports local businesses by optimizing order processing, ensuring that they can efficiently fulfill orders. It also contributes to reducing food wastage through better inventory management.
5	Business Model (Revenue Model)	The business model for our app is based on the freemium approach. Basic features are provided for free, while revenue is generated through premium subscriptions that unlock advanced features. Additionally, our app will earn income by charging commission fees to partner businesses, displaying in-app advertisements, and possibly offering in-app purchases. Data analytics may be used to provide insights and further monetization opportunities. This combination ensures sustainability and growth for the app.
6	Scalability of the Solution	Scalability is an essential aspect of our app's infrastructure. It's designed to handle a growing user base effectively. The app uses cloud-based infrastructure, enabling it to scale horizontally as the user count increases. This ensures optimal performance even during peak usage times. The architecture is set up to add more computing resources as needed, guaranteeing a smooth and responsive user experience regardless of the number of users using the app.

Solution Architecture Diagram



6.PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project Setup	USN-1	Set up the environment needed to do this project by collecting and knowing all the required plugins and library dependencies. Also we should add them in gradle Script files.	15	High	Mahesh
Sprint-2	User Interface Design	USN-2	For every application User Interface plays very important role. So User Interface should be simple and stylish at the same time when lookinginto the screen users should be able to understand. a. Login Interface b. Registration Interface c. List Interface d. Order Interface	48	Moderate to High	Siva Nani Yogeshwar Mahesh
Sprint-3	Database Integration	USN-3	Whenever a new user registers, his logincredentials should be stored in the database. Sothat if a new user enters the same data and try toregister "User Already Exists" message should be notified.	15	High	Siva
Sprint-4	Firebase Authentication	USN-4	Whenever a user enters username or email id or mobile number and password they should be Verified and login action should be done.	10	High	Yogeshwar
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members

Sprint-5	Navigation Features	USN-5	In every application clear and visible navigation buttons should be there. So that User can have Better experience of the App.	20	Moderate	Siva and Mahesh
Sprint-6	Robustness	USN-6	Whenever users use the application concurrently we should be able handle all of them. The app should run perfectly without any lag and difficulty.	15	Low	Nani and Yogeshwar

Project Tracker, Velocity & Burn down Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	2 Days	16 Oct 2023	17 Oct 2023	15	17 Oct 2023
Sprint-2	48	5 Days	17 Oct 2023	21 Oct 2023	48	20 Oct 2023
Sprint-3	15	1 Days	22 Oct 2023	22 Oct 2023	15	21 Oct 2023
Sprint-4	10	1 Days	23 Oct 2023	24 Oct 2023	10	22 Oct 2023
Sprint-5	20	2 Days	23 Oct 2023	24 Oct 2023	20	23 Oct 2023
Sprint-6	15	2 Days	24 Oct 2023	26 Oct 2023	15	23 Oct 2023

Velocity:

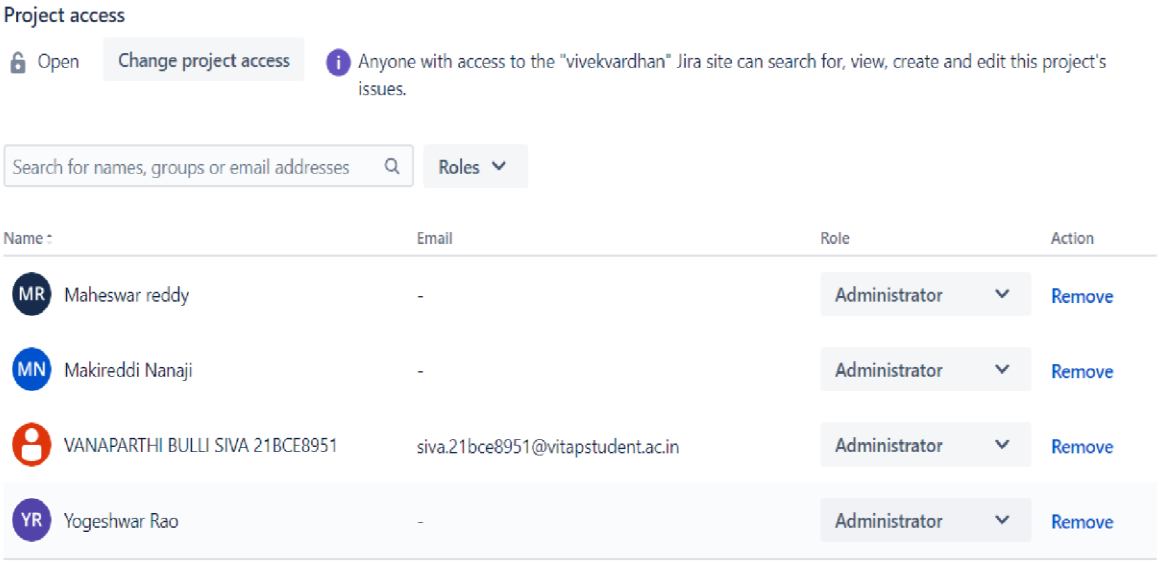
We have a 13-day sprint duration, and the velocity of the team is 16.67 (average points per sprint). If we calculate the team's average velocity (AV) per iteration unit (story points per day)

AV = (sprint duration) / (velocity) = 13/10 = 1.3

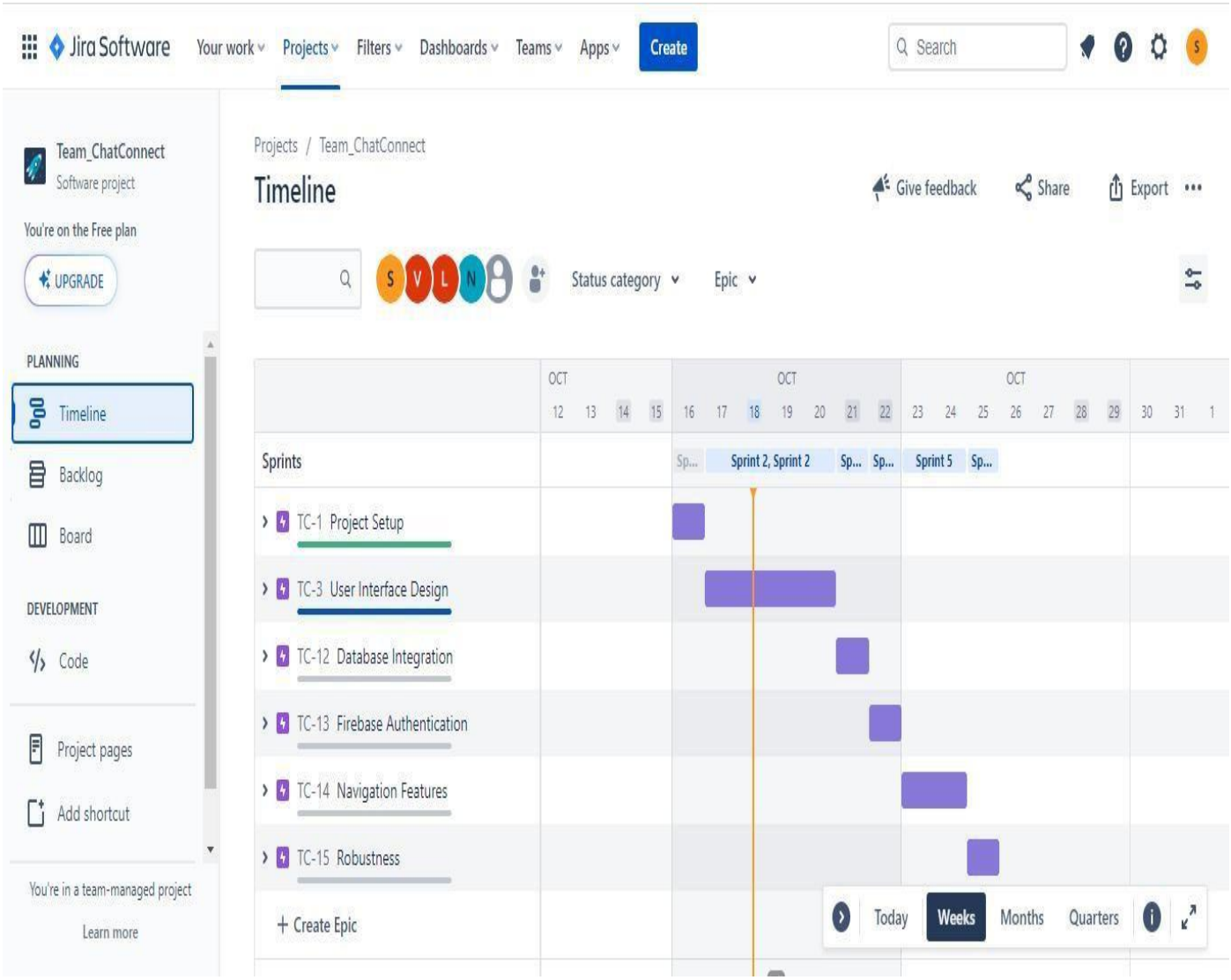
Burn down Chart:

A Burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

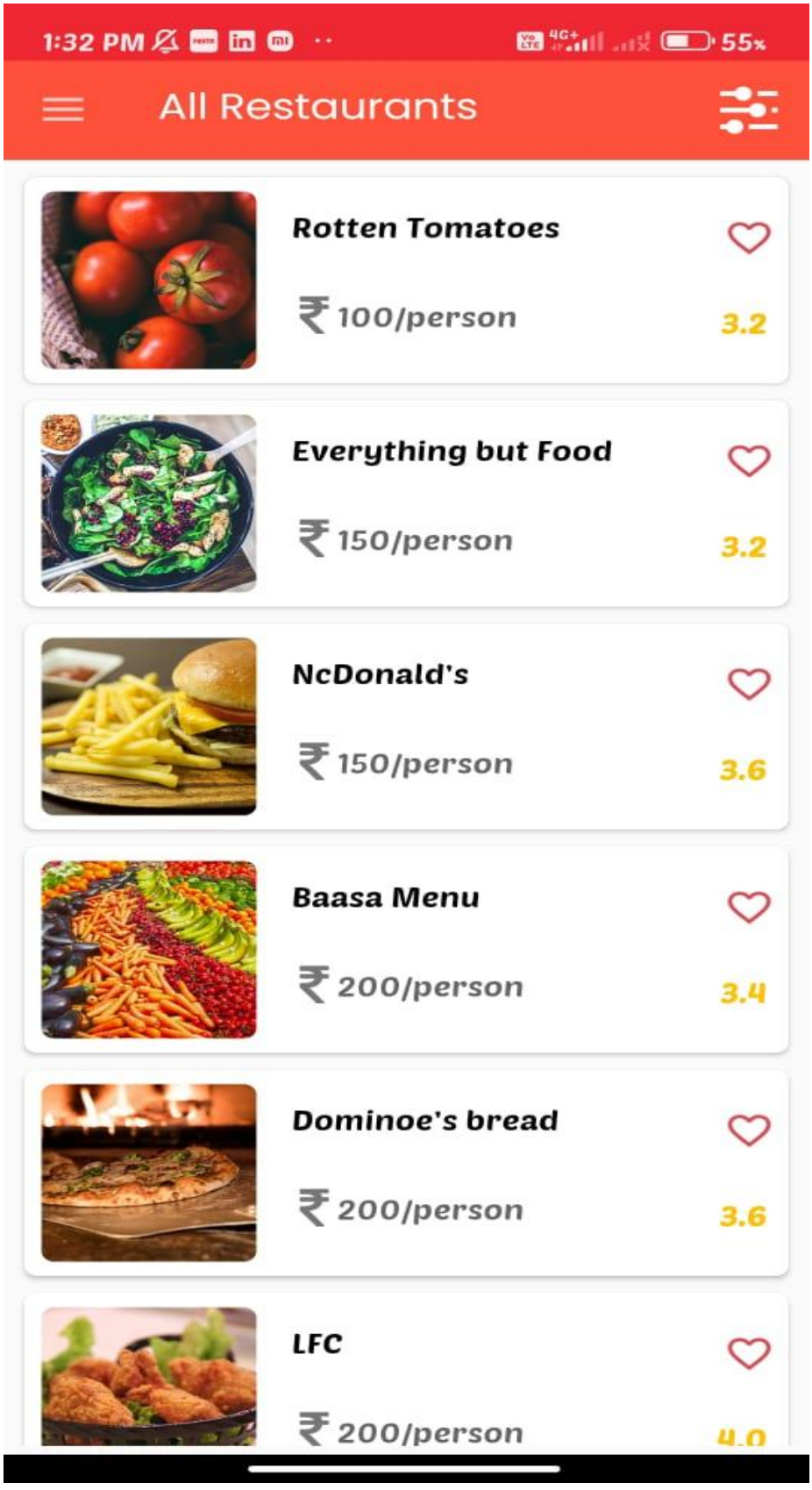
TEAM MEMEBERS:



Timeline:


Jira Software
Your work
Projects
Filters
Dashboards
Teams
Apps
Create
Search
Team_ChatConnect
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
You're in a team-managed project
Learn more
Projects / Team_ChatConnect
Timeline
Give feedback
Share
Export
Status category
Epic
OCT
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1
Sprints
Sprint 2, Sprint 2
Sprint 5
TC-1 Project Setup
TC-3 User Interface Design
TC-12 Database Integration
TC-13 Firebase Authentication
TC-14 Navigation Features
TC-15 Robustness
+ Create Epic
Today
Weeks
Months
Quarters

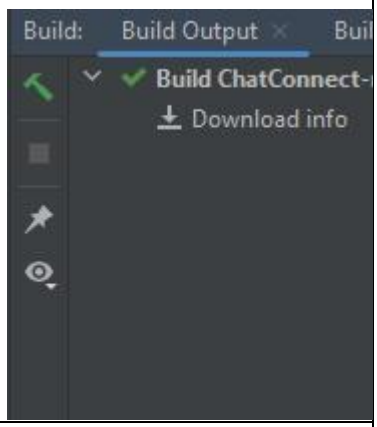
7.CODING & SOLUTIONING

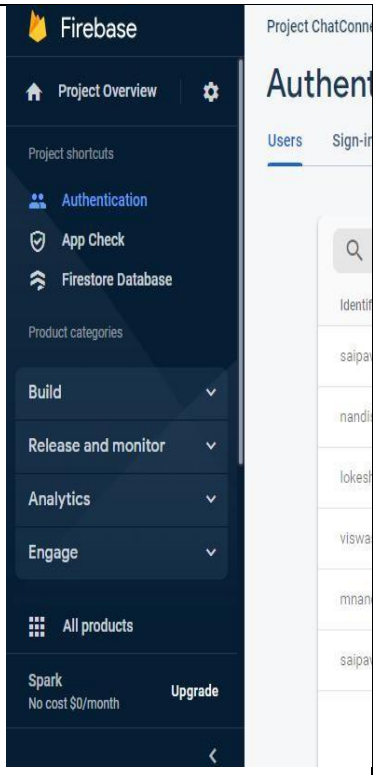


8.PERFORMANCE TESTING

Model Performance Testing:

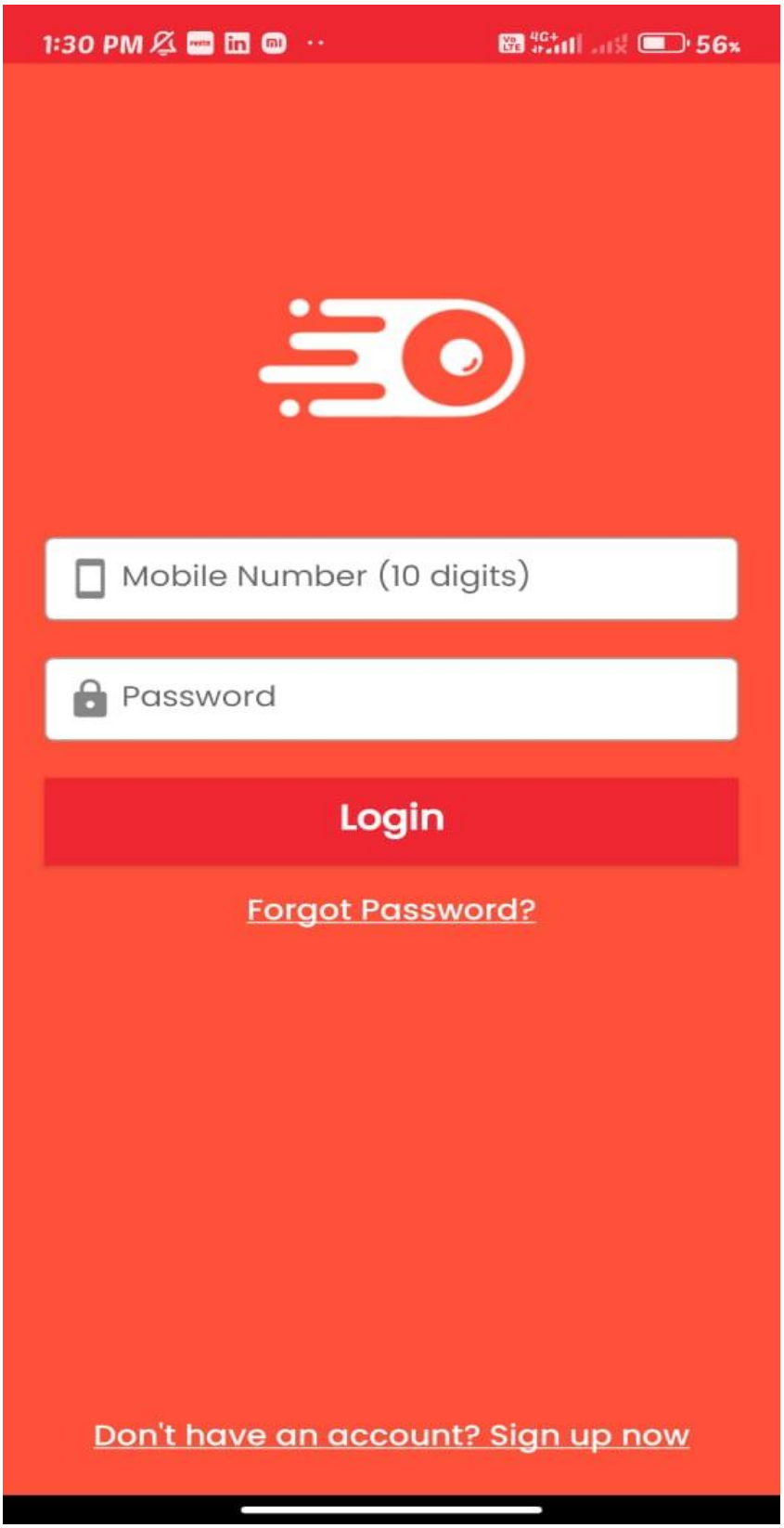
Project team shall fill the following information in model performance testing template.

S.No	Parameter	Values	Screenshot
1.	Metrics	App Launch Time- Screen Render Time- Code Quality-	
2.	Usage	App Size- Customer Experience-	 Customer Experience is Good and Better.

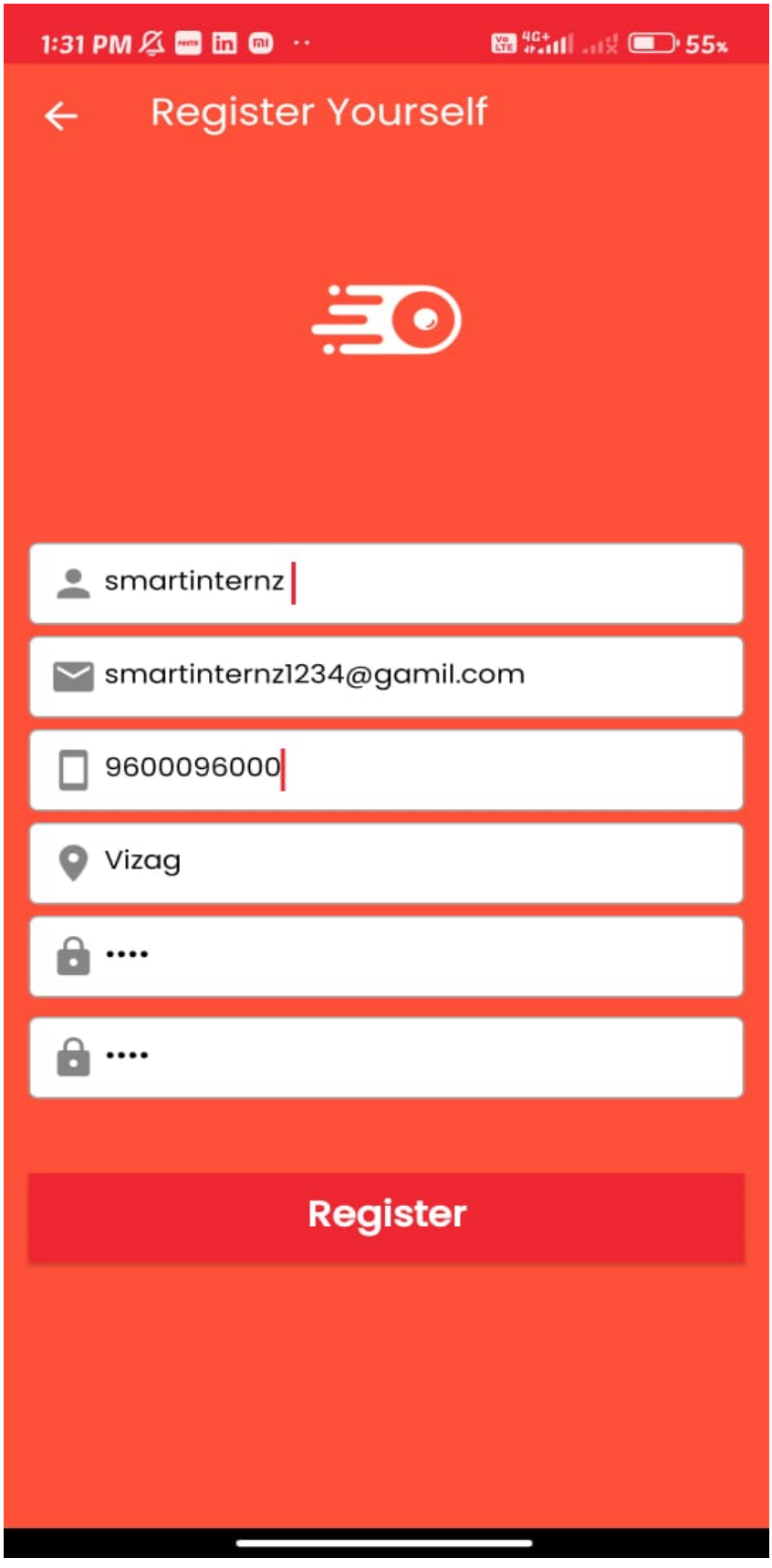
3.	Performance	Error and Crash Rates- Database Query Performance-	 Error crash rates are very less so far.

9.OUTPUT:

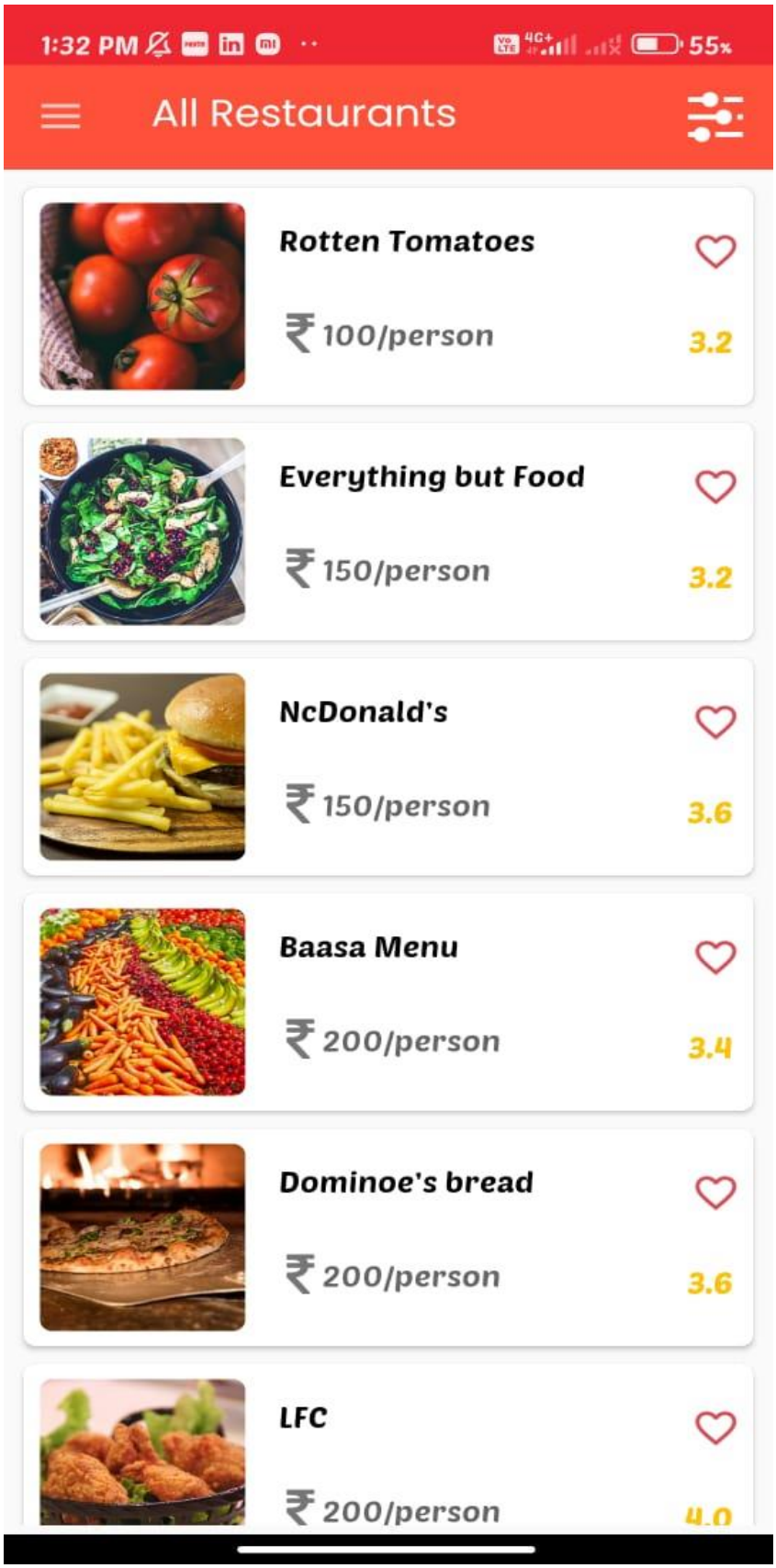
LOGIN PAGE



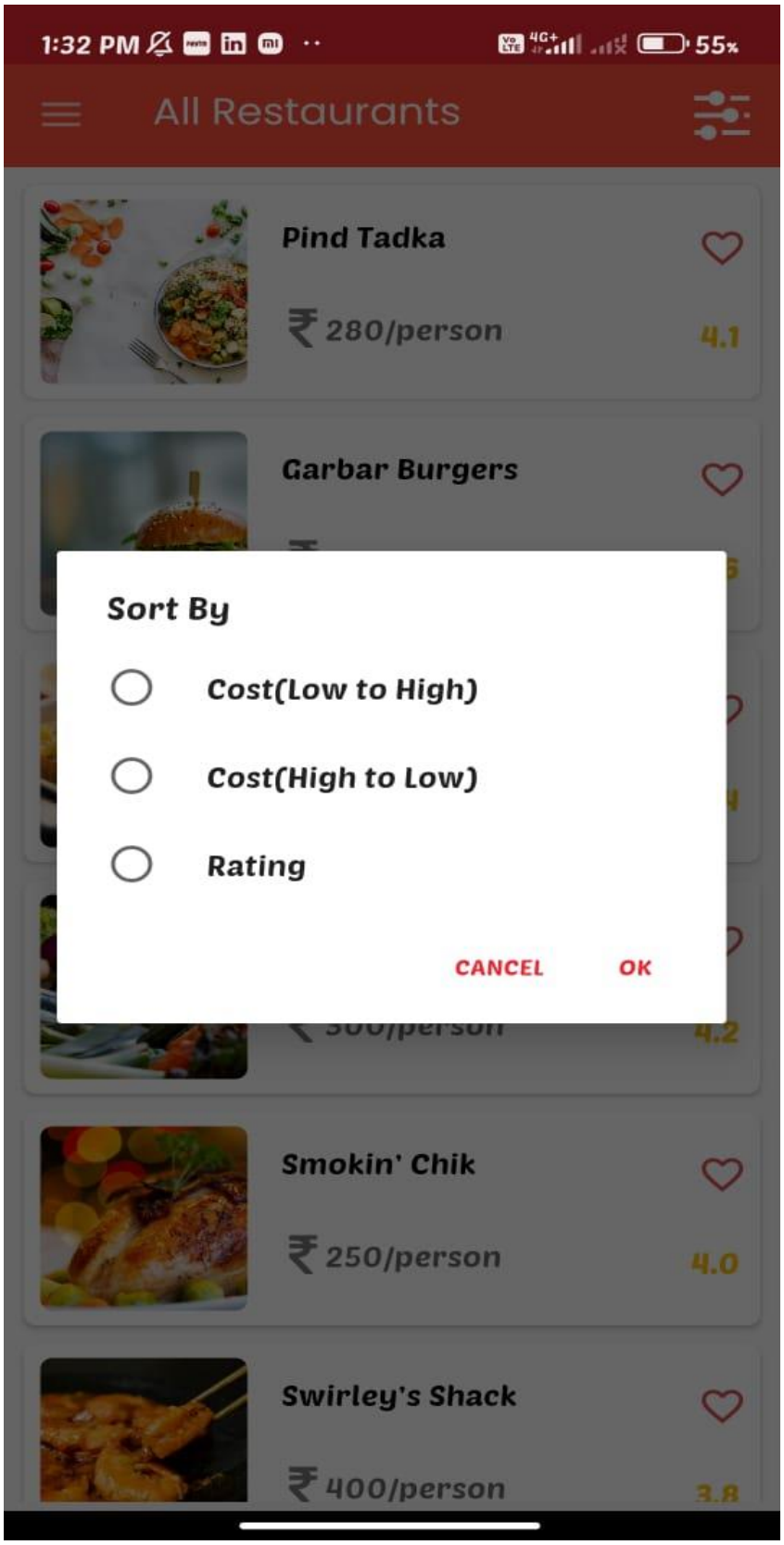
SIGN UP PAGE

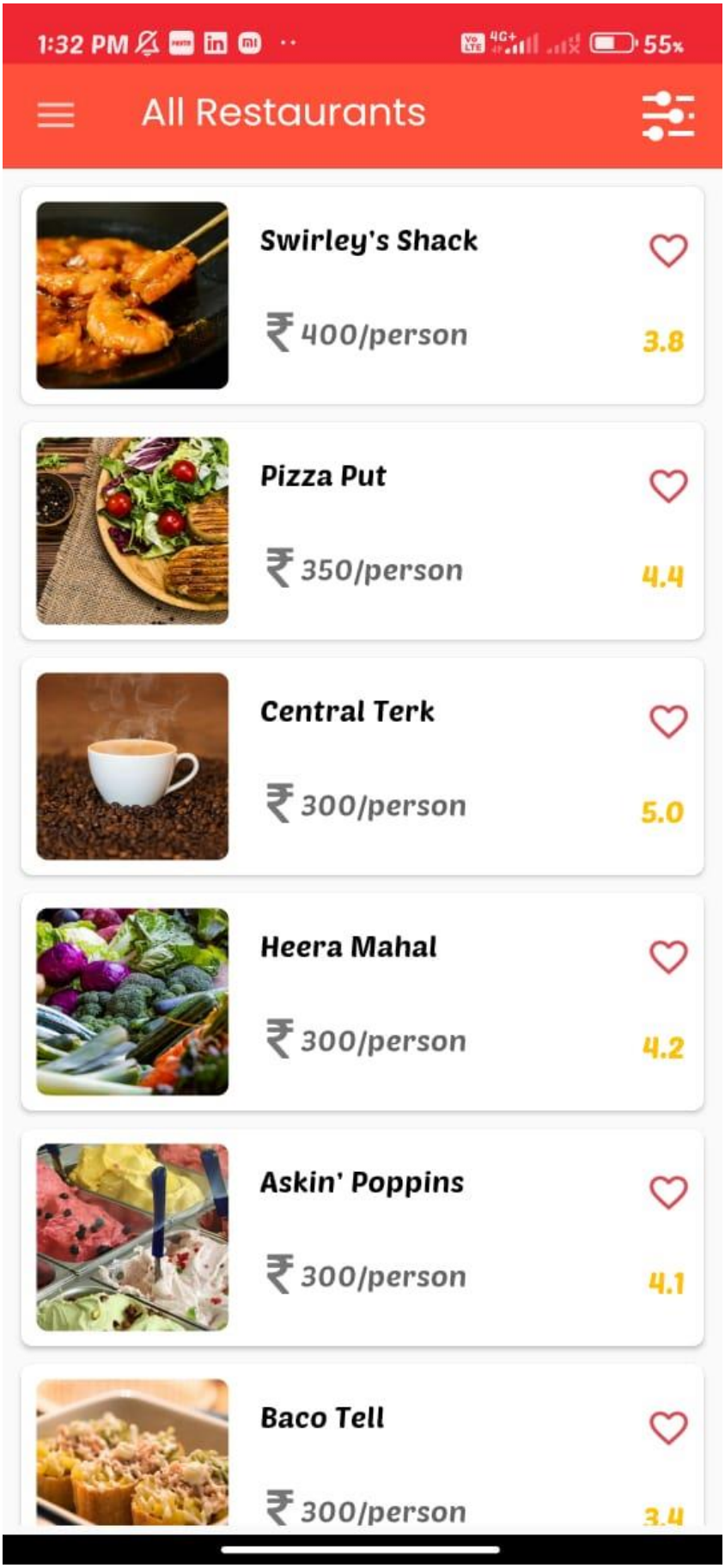

1:31 PM
55%
Register Yourself
smartinternz
smartinternz1234@gamil.com
9600096000
Vizag
Register

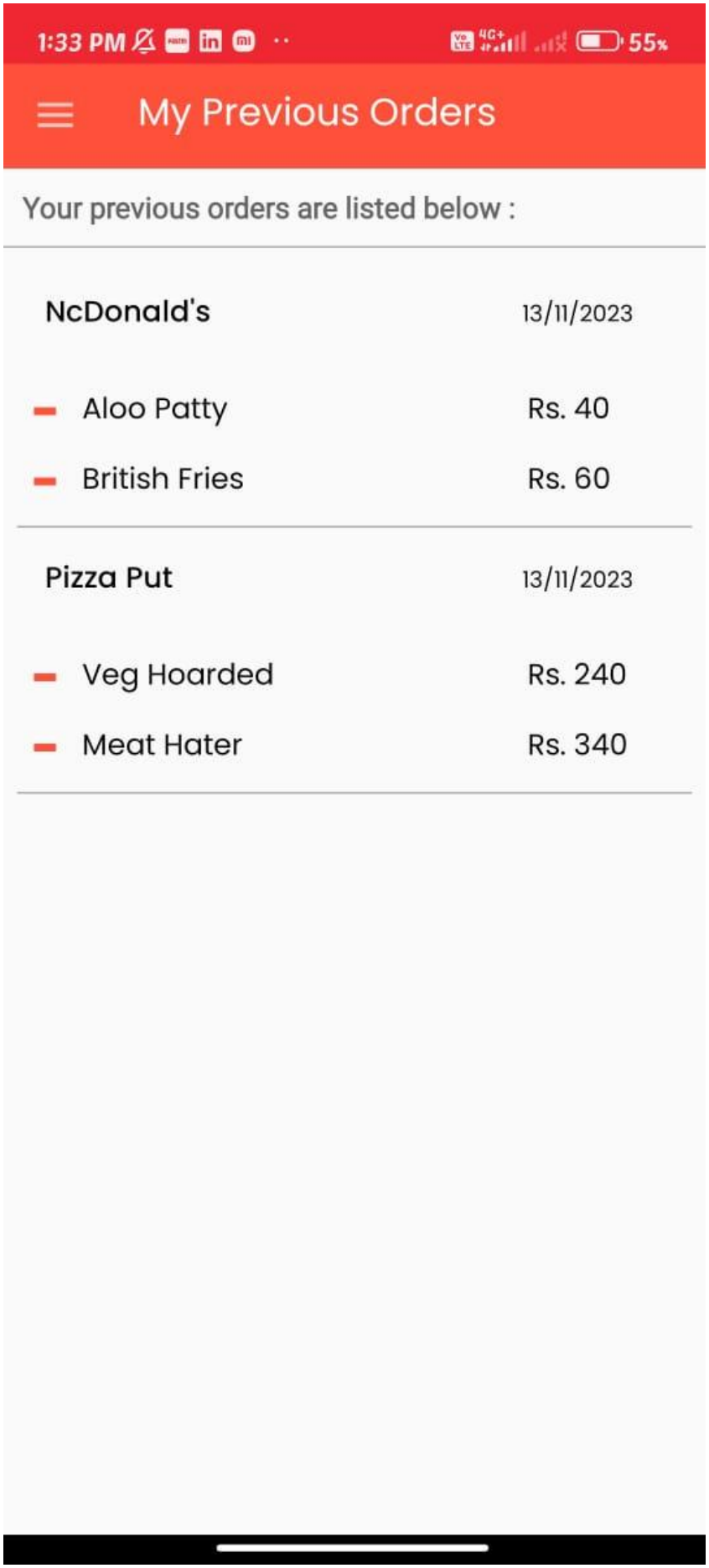
HOME SCREEN

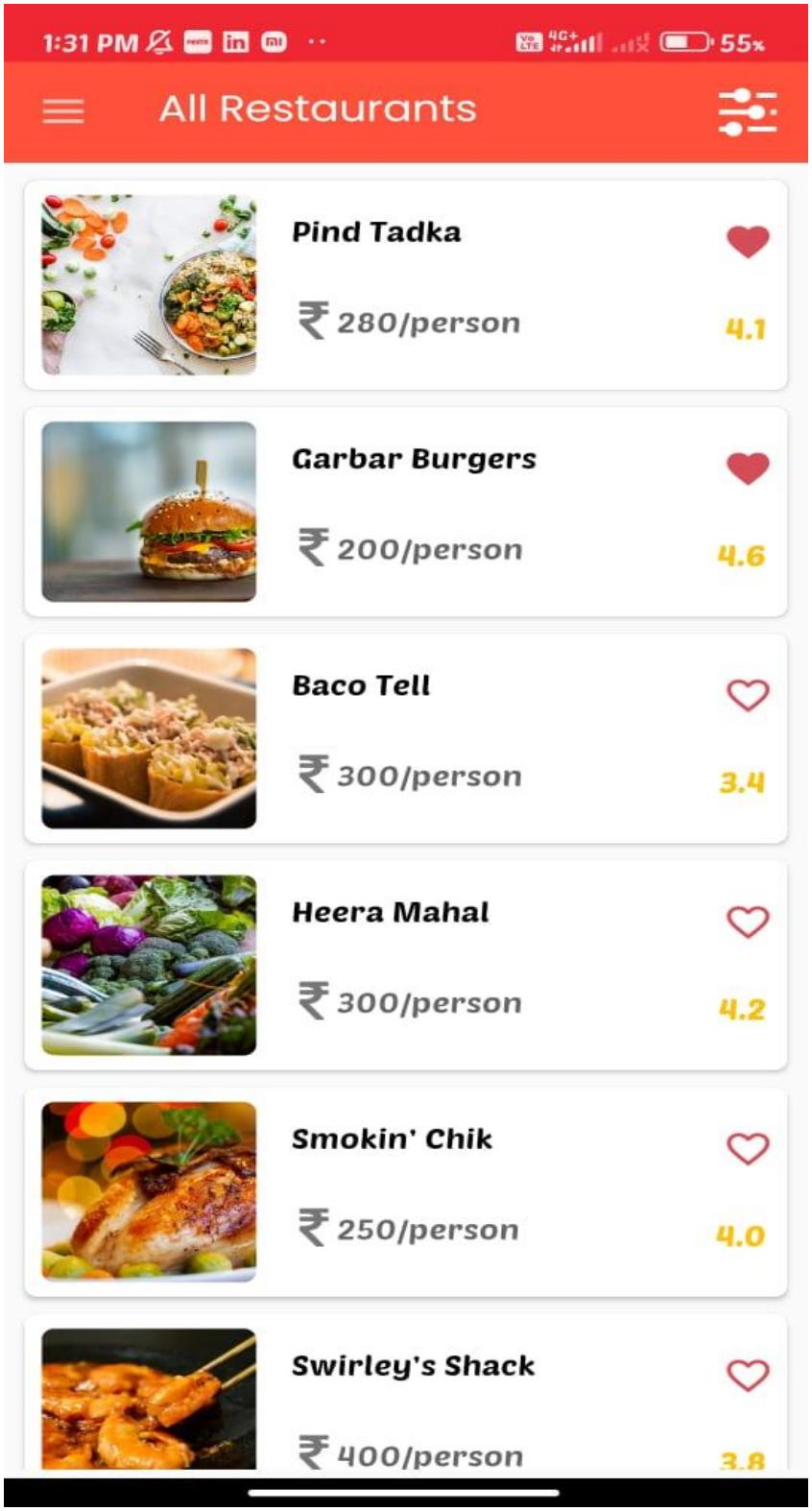

1:32 PM
55%
All Restaurants
Rotten Tomatoes
₹ 100/person
3.2
Everything but Food
₹ 150/person
3.2
NcDonald's
₹ 150/person
3.6
Baasa Menu
₹ 200/person
3.4
Dominoe's bread
₹ 200/person
3.6
LFC
₹ 200/person
4.0

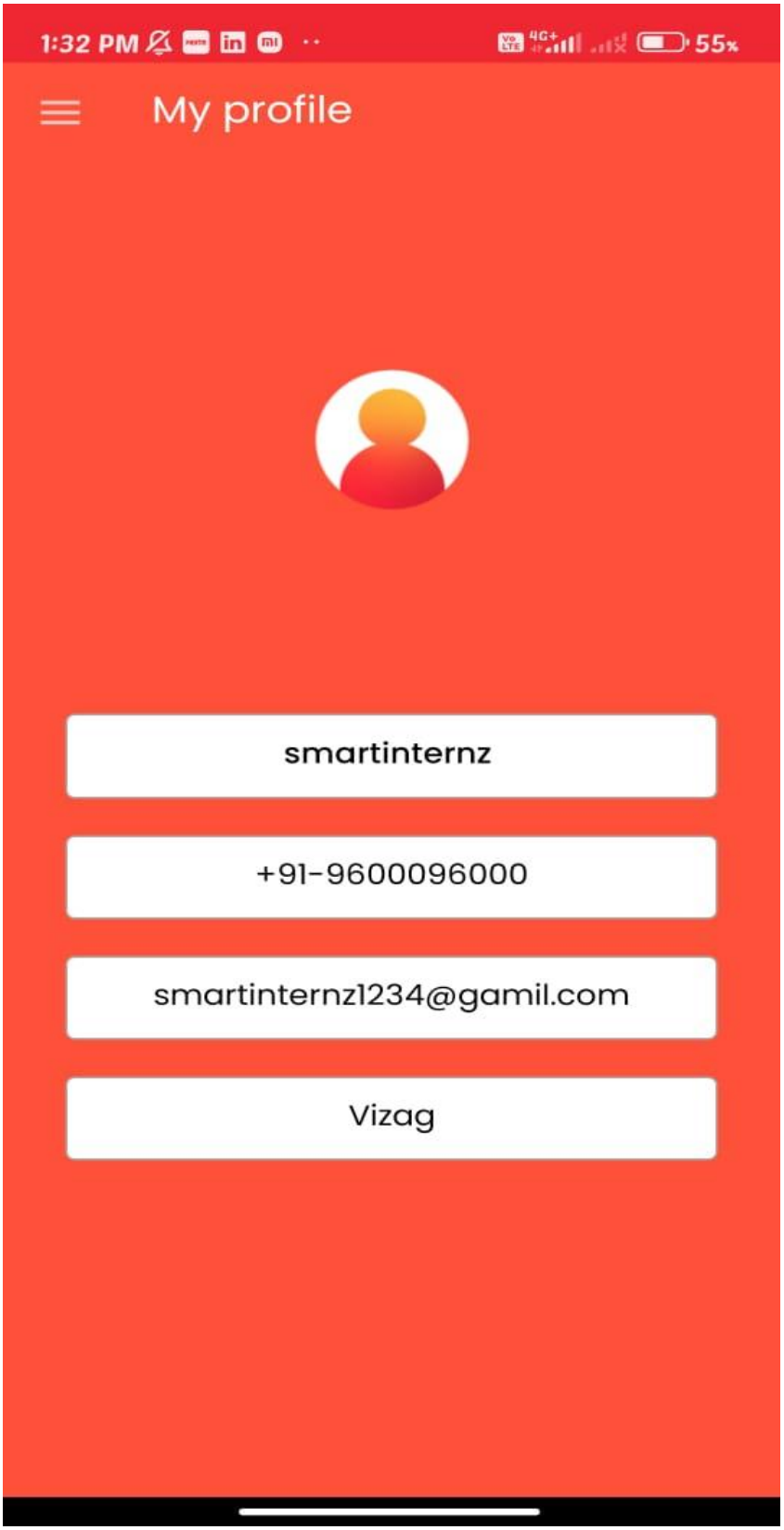
SORTING

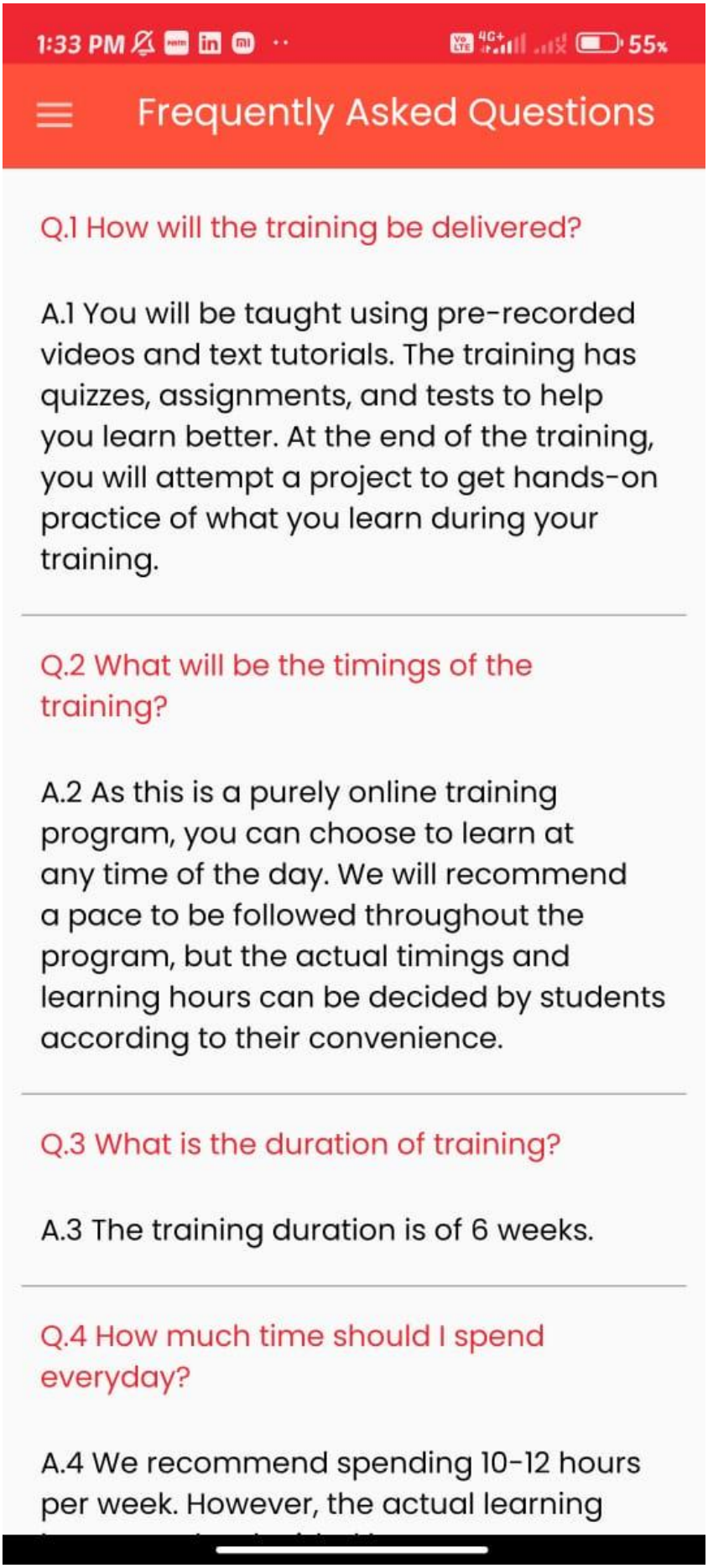

1:32 PM
55%
All Restaurants
Pind Tadka
₹ 280/person
4.1
Garbar Burgers
Sort By
Cost(Low to High)
Cost(High to Low)
Rating
CANCEL
OK
Smokin' Chik
₹ 250/person
4.0
Swirley's Shack
₹ 400/person

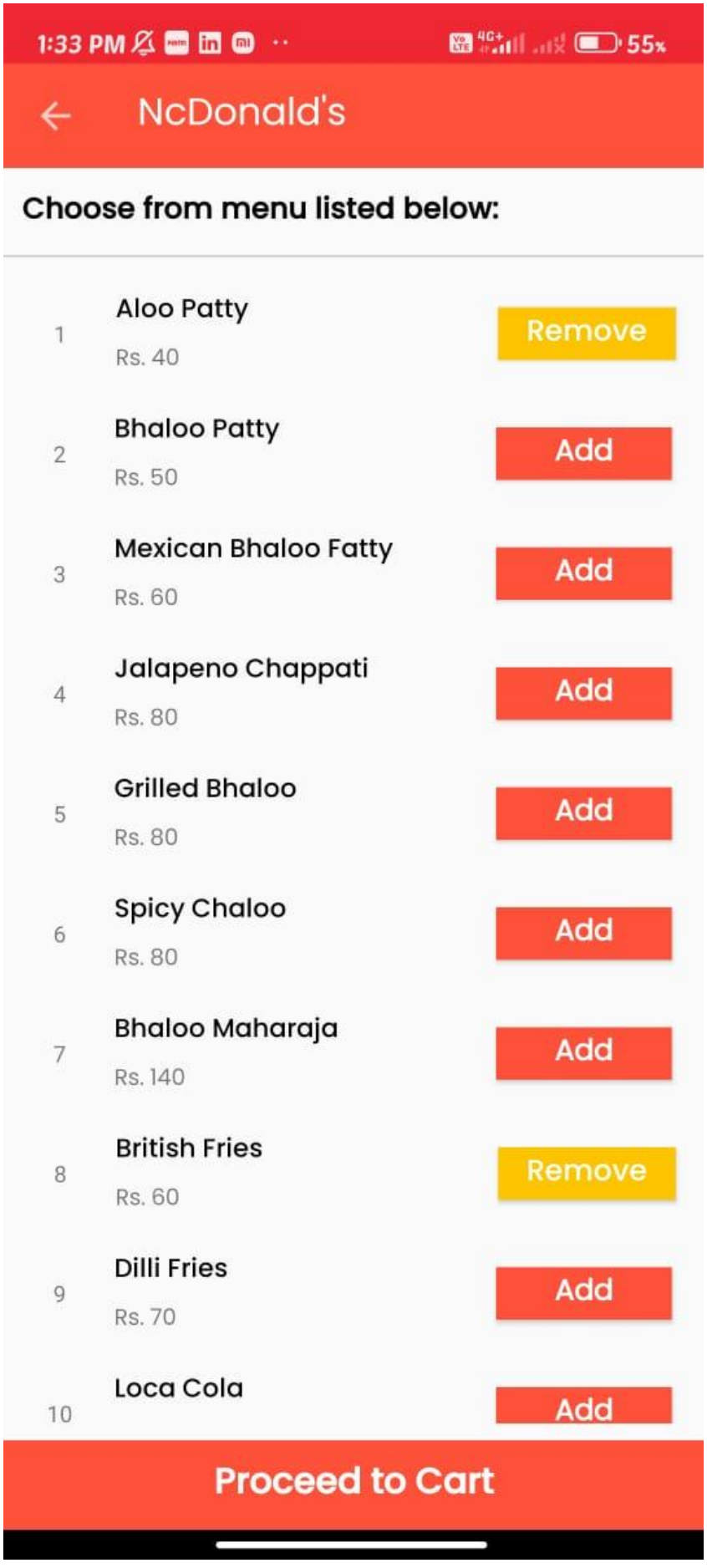

1:32 PM
55%
All Restaurants
Swirley's Shack
₹ 400/person
3.8
Pizza Put
₹ 350/person
4.4
Central Terk
₹ 300/person
5.0
Heera Mahal
₹ 300/person
4.2
Askin' Poppins
₹ 300/person
4.1
Baco Tell
₹ 300/person
3.4


1:33 PM
55%
My Previous Orders
Your previous orders are listed below :
NcDonald's
13/11/2023
Aloo Patty
Rs. 40
British Fries
Rs. 60
Pizza Put
13/11/2023
Veg Hoarded
Rs. 240
Meat Hater
Rs. 340

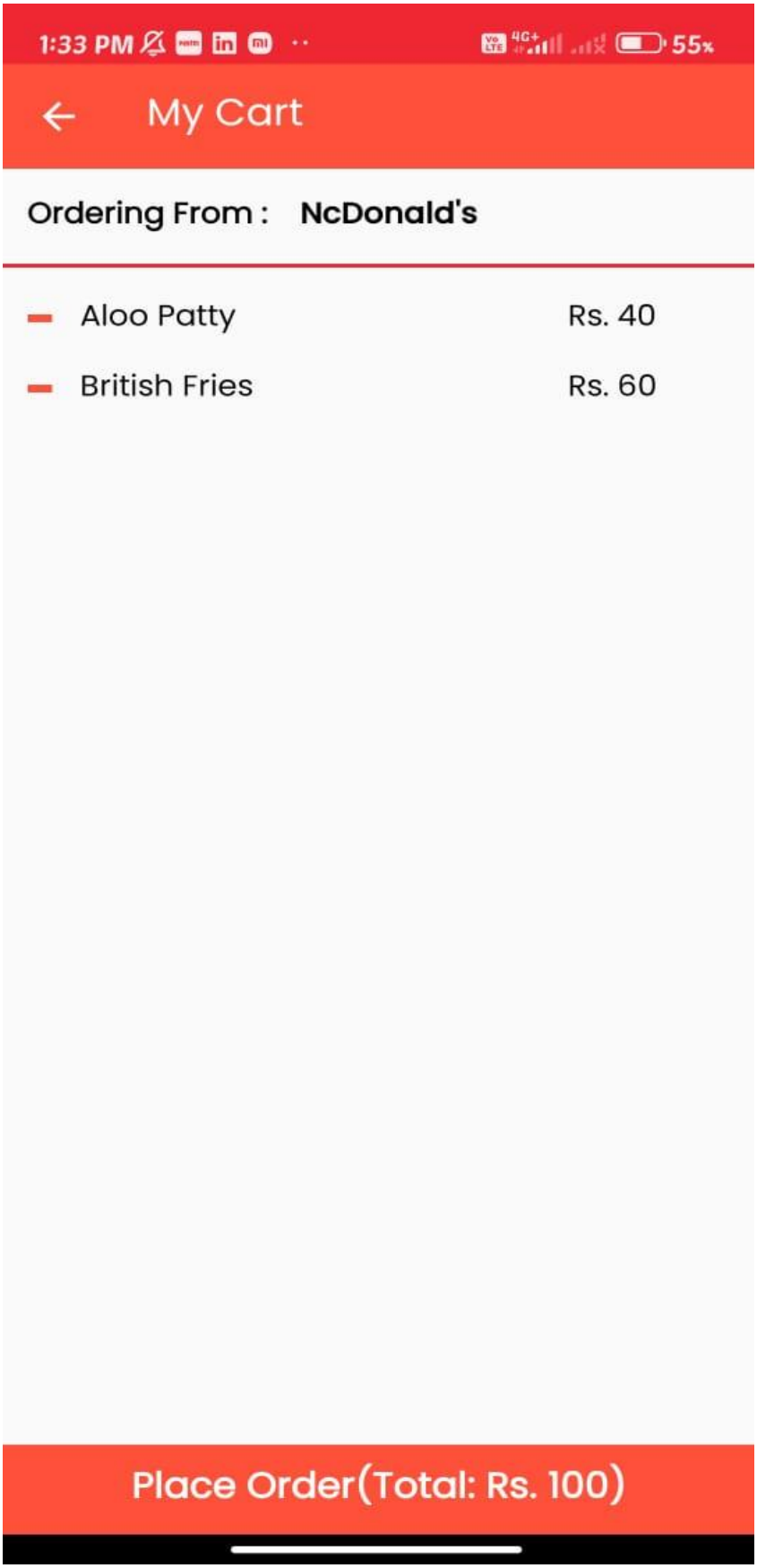

1:31 PM
55%
All Restaurants
Pind Tadka
₹ 280/person
4.1
Garbar Burgers
₹ 200/person
4.6
Baco Tell
₹ 300/person
3.4
Heera Mahal
₹ 300/person
4.2
Smokin' Chik
₹ 250/person
4.0
Swirley's Shack
₹ 400/person
3.8

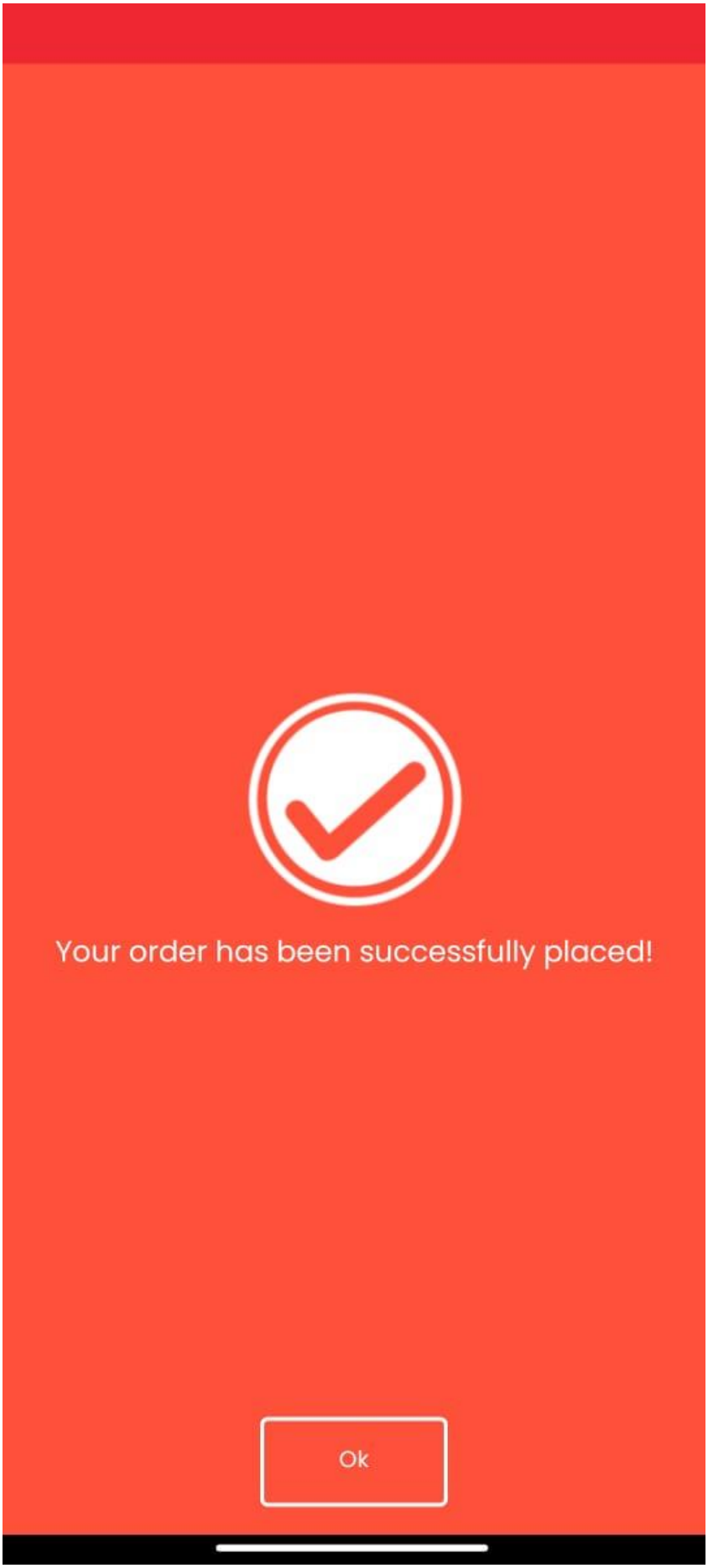

1:32 PM
55%
My profile
smartinternz
+91-9600096000
smartinternz1234@gamil.com
Vizag


1:33 PM
55%
Frequently Asked Questions
Q.1 How will the training be delivered?
A.1 You will be taught using pre-recorded videos and text tutorials. The training has quizzes, assignments, and tests to help you learn better. At the end of the training, you will attempt a project to get hands-on practice of what you learn during your training.
Q.2 What will be the timings of the training?
A.2 As this is a purely online training program, you can choose to learn at any time of the day. We will recommend a pace to be followed throughout the program, but the actual timings and learning hours can be decided by students according to their convenience.
Q.3 What is the duration of training?
A.3 The training duration is of 6 weeks.
Q.4 How much time should I spend everyday?
A.4 We recommend spending 10-12 hours per week. However, the actual learning


1:33 PM
55%
NcDonald's
Choose from menu listed below:
1
Aloo Patty
Rs. 40
Remove
2
Bhaloo Patty
Rs. 50
Add
3
Mexican Bhaloo Fatty
Rs. 60
Add
4
Jalapeno Chappati
Rs. 80
Add
5
Grilled Bhaloo
Rs. 80
Add
6
Spicy Chaloo
Rs. 80
Add
7
Bhaloo Maharaja
Rs. 140
Add
8
British Fries
Rs. 60
Remove
9
Dilli Fries
Rs. 70
Add
10
Loca Cola
Add
Proceed to Cart

FOOD ORDERING




Your order has been successfully placed!
Ok

10. ADVANATGES

1. Convenience:

- Users can enjoy the convenience of ordering snacks anytime, anywhere, without the need to visit physical stores.

2. Variety and Choices:

- The app provides a wide variety of snacks from different vendors, offering users diverse choices to suit their preferences.

3. Efficient Ordering Process:

- Streamlined and user-friendly interfaces make the ordering process quick and efficient, enhancing the overall user experience.

4. Real-time Tracking:

- Users can track their orders in real-time, providing transparency and reducing uncertainty about when their snacks will be delivered.

5. Digital Payments:

- Secure digital payment options within the app eliminate the need for cash transactions, making payments convenient and reliable.

Disadvantages:

1. Technical Issues:

- App crashes, server downtimes, or other technical issues can disrupt the user experience and erode trust in the reliability of the service.

2. Delivery Challenges:

- Timely and reliable delivery is crucial, and challenges such as delays or damaged goods during transit can impact customer satisfaction.

3. Competition and Market Saturation:

- Intense competition in the food delivery app market, coupled with market saturation in some regions, can make it challenging to attract and retain users.

4. Costs for Vendors:

- Vendors may face increased costs due to commission fees, impacting their profit margins and potentially affecting the range of snacks offered.

5. Data Privacy Concerns:

- Handling user data requires strict adherence to privacy regulations. Any breaches or mishandling of data can lead to legal and reputational issues, affecting user trust.

11.CONCLUSION

In conclusion, developing a snack ordering and delivery app using Kotlin for Android brings several advantages, including interoperability with Java, concise and readable code, null safety, co_routines for asynchronous programming, and extension functions for code modularity. However, there are some considerations such as a potential learning curve, longer compilation times, library support variations, community size, and occasional Android Studio integration challenges. Assessing these factors based on your team's expertise and project requirements will help you make an informed decision on whether Kotlin is the right choice for your app development.

12. FUTURE SCOPE

The future scope for your snack ordering and delivery app is promising, with ample opportunities for innovation and growth. Embracing emerging technologies such as artificial intelligence for personalized recommendations, machine learning for demand forecasting, and augmented reality for interactive menu experiences could elevate the app's functionality and user engagement. Integration with Internet of Things (IoT) devices in kitchens or delivery vehicles could optimize operational efficiency. As sustainability becomes a focal point, incorporating eco-friendly packaging and exploring partnerships with environmentally conscious vendors might resonate well with increasingly environmentally aware consumers. Expanding globally, adapting to changing consumer behaviors, and staying agile to incorporate advancements in mobile payment systems are also crucial for sustained success. Continuous adaptation to technological trends, user preferences, and industry developments will position your snack delivery app for long-term relevance and competitiveness in the dynamic landscape of food delivery.

13.CODES

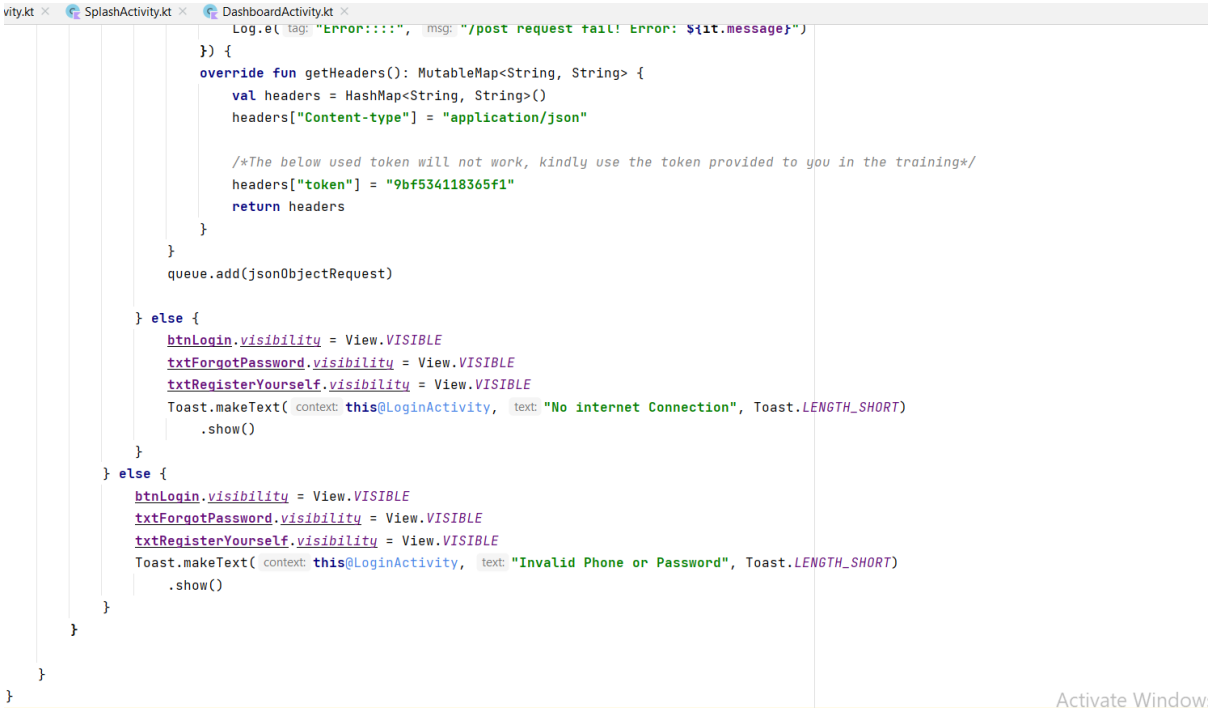
LOGIN ACTIVITY

```
class LoginActivity : AppCompatActivity() {

    /*Declaring all the views present in the activity_login.xml file*/
    lateinit var etMobileNumber: EditText
    lateinit var etPassword: EditText
    lateinit var btnLogin: Button
    lateinit var txtForgotPassword: TextView
    lateinit var txtRegisterYourself: TextView

    /*Variables used in managing the login session*/
    lateinit var sessionManager: SessionManager
    lateinit var sharedPreferences: SharedPreferences

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        /*Initialising the views*/
        etMobileNumber = findViewById(R.id.etMobileNumber)
        etPassword = findViewById(R.id.etPassword)
```

```
        btnLogin = findViewById(R.id.btnLogin)
        txtForgotPassword = findViewById(R.id.txtForgotPassword)
        txtRegisterYourself = findViewById(R.id.txtRegisterYourself)

        /*Initialising the session variables*/
        sessionManager = SessionManager(this)
        sharedPreferences =
            this.getSharedPreferences(sessionManager.PREF_NAME,
sessionManager.PRIVATE_MODE)

        /*Clicking on this text takes you to the forgot password activity*/
        txtForgotPassword.setOnClickListener {
            startActivity(Intent(this@LoginActivity,
ForgotPasswordActivity::class.java))
        }

        /*Clicking on this text takes you to the forgot password activity*/
        txtRegisterYourself.setOnClickListener {
            startActivity(Intent(this, RegisterActivity::class.java))
        }

        /*Start the login process when the user clicks on the login
button*/
        btnLogin.setOnClickListener {

            /*Hide the login button when the process is going on*/
            btnLogin.visibility = View.INVISIBLE

            /*First validate the mobile number and password length*/
            if (Validations.validateMobile(etMobileNumber.text.toString())
&& Validations.validatePasswordLength(etPassword.text.toString())) {
                if
(ConnectionManager().isNetworkAvailable(this@LoginActivity)) {

                    /*Create the queue for the request*/
                    val queue = Volley.newRequestQueue(this@LoginActivity)

                    /*Create the JSON parameters to be sent during the
login process*/
                    val jsonParams = JSONObject()
                    jsonParams.put("mobile_number",
etMobileNumber.text.toString())
                    jsonParams.put("password", etPassword.text.toString())


                    /*Finally send the json object request*/
                    val jsonObjectRequest = object :
JsonObjectRequest(Method.POST, LOGIN, jsonParams,
                        Response.Listener {

                            try {
                                val data = it.getJSONObject("data")
                                val success = data.getBoolean("success")
                                if (success) {
                                    val response =
data.getJSONObject("data")
                                    sharedPreferences.edit()
                                        .putString("user_id",
response.getString("user_id")).apply()
                                    sharedPreferences.edit()
                                        .putString("user_name",
```

```
response.getString("name")).apply()
                                    sharedPreferences.edit()
                                        .putString(
                                            "user_mobile_number",

response.getString("mobile_number")
                                        )
                                        .apply()
                                    sharedPreferences.edit()
                                        .putString("user_address",
response.getString("address"))
                                        .apply()
                                    sharedPreferences.edit()
                                        .putString("user_email",
response.getString("email")).apply()
                                    sessionManager.setLogin(true)
                                    startActivity(
                                        Intent(
                                            this@LoginActivity,
                                            DashboardActivity::class.java
                                        )
                                    )
                                    finish()
                                } else {
                                    btnLogin.visibility = View.VISIBLE
                                    txtForgotPassword.visibility =
View.VISIBLE
                                    btnLogin.visibility = View.VISIBLE
                                    val errorMessage =
data.getString("errorMessage")
                                    Toast.makeText(
                                        this@LoginActivity,
                                        errorMessage,
                                        Toast.LENGTH_SHORT
                                    ).show()
                                }
                            } catch (e: JSONException) {
                                btnLogin.visibility = View.VISIBLE
                                txtForgotPassword.visibility = View.VISIBLE
                                txtRegisterYourself.visibility =
View.VISIBLE
                                e.printStackTrace()
                            }
                        },
                        Response.ErrorListener {
                            btnLogin.visibility = View.VISIBLE
                            txtForgotPassword.visibility = View.VISIBLE
                            txtRegisterYourself.visibility = View.VISIBLE
                            Log.e("Error::::", "/post request fail! Error:
${it.message}")
                        }) {
                        override fun getHeaders(): MutableMap<String,
String> {
                            val headers = HashMap<String, String>()
                            headers["Content-type"] = "application/json"

                            /*The below used token will not work, kindly
use the token provided to you in the training*/
                            headers["token"] = "9bf534118365f1"
                            return headers
                        }
```

```
                    }
                    queue.add(jsonObjectRequest)

                } else {
                    btnLogin.visibility = View.VISIBLE
                    txtForgotPassword.visibility = View.VISIBLE
                    txtRegisterYourself.visibility = View.VISIBLE
                    Toast.makeText(this@LoginActivity, "No internet
Connection", Toast.LENGTH_SHORT)
                        .show()
                }
            } else {
                btnLogin.visibility = View.VISIBLE
                txtForgotPassword.visibility = View.VISIBLE
                txtRegisterYourself.visibility = View.VISIBLE
                Toast.makeText(this@LoginActivity, "Invalid Phone or
Password", Toast.LENGTH_SHORT)
                    .show()
            }
        }

    }
}
```



THE END