CrazyWin

A PROJECT REPORT
Submitted By

Abhay Ranjan
2000290140003

Ashutosh Kumar
2000290140030

Yash Yadav
2000290140141

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Dr. Vipin Kumar
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

CERTIFICATE

Certified that **Abhay Ranjan (2000290140003), Ashutosh Kumar (2000290140030), Yash Yadav (2000290140141)** have carried out the project work having **“CrazyWin”** for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Abhay Ranjan (2000290140003),
Ashutosh Kumar (2000290140030),
Yash Yadav (2000290140141)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Vipin Kumar
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay Shrivastava
Head of Department, Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

In the league of myriad fantasy-sports mobile apps, it is one name that has successfully carved its niche among sports lovers and continues to do so. In simple terms, it is a fantasy sports platform based in India that allows users to play fantasy cricket and games. As the term abbreviates itself, Fantasy Sports are to experience the best feel of sports. It is a virtual game that is played with a more guessing style. Also known as Fantasy Sports league, it usually offers a sign-up bonus, cash prizes, and a lot of enthrallment to its participants.

We try to make a *Fantasy apps like Dream11* with an ability where its participants can win cash prizes legally. **CrazyWin** app architecture allows its users to withdraw their prize money over the transfer from their CrazyWin account linked with a PAN card (mandatory as an ID proof) and get money into the bank account.

It doesn’t come as very surprising to know about the prominence of Fantasy apps like Dream11 in the country and across the world and it has grown really big. As the term abbreviates itself, Fantasy Sports are to experience the best feel of sports.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Vipin Kumar** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava,** Professor and Head of Department, Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Abhay Ranjan
Ashutosh Kumar
Yash Yadav

List of Chapters

Chapter 1 Introduction

1.1 Overall Description

It doesn't come as very surprising to know about the prominence of Fantasy apps like Dream11 in the country and across the world and it has grown really big. As the term abbreviates itself, Fantasy Sports are to experience the best feel of sports. It is a virtual game that is played with a more guessing style. Also known as Fantasy Sports league, it usually offers a sign-up bonus, cash prizes, and a lot of enthrallments to its participants.

By considering all the fact we also try to make a *Fantasy apps like Dream11* with an ability where its participants can win cash prizes legally. **CrazyWin** app architecture allows its users to withdraw their prize money over the transfer from their CrazyWin account linked with a PAN card (mandatory as an ID proof) and get money into the bank account.

In the league of myriad fantasy-sports mobile apps, it is one name that has successfully carved its niche among sports lovers and continues to do so. In simple terms, it is a fantasy sports platform based in India that allows users to play fantasy cricket and games.

1.1.1 Product Objective

The objective of the project is to design and develop fantasy sports software which is quite similar to Dream 11. This application provides a platform for playing fantasy Cricket. It is an online game where users create a Virtual Team of players and earn points based on the performances of these players in real matches. A user who scores the maximum points in his joined contest attains the first rank on the leader board. Offers free and paid contests. A user doesn't has to pay any fee to join a contest and can win real cash.

1.1.2 Product Features

As per shared specifications,

Features can be divided into three categories:

User-friendly Features

- Registration/log-in
- Home Screen
- Contest
- Payment Modes
- My Contest Details

- My Contest
- My Profile
- My Account(Wallets)
- Team Creations & Team Preview

Additional Features

- Live Match Score
- Integration of API
- CRM integration
- Push Notification
- Real-time Analytics

1.3 HARDWARE AND SOFTWARE REQUIREMENT

1.3.1 Hardware Specification

- Central Processing Unit (CPU) - Intel Core i5 6th gen or AMD processor equivalent
- RAM - 8 GB minimum, 16 GB or higher is recommended.
- Operating System (OS) - Ubuntu or Microsoft Windows 10
- Storage - 20 GB

1.3.2 Software Specification

- **Language used:** PHP, Java
- **Editor:** Android Studio
- **Operating System:** Windows 10

1.4 Functional and Non Functional Requirements

1.4.1 Functional Requirements

1. **Signup**
   User need to enter the details for signup over android and can register his account.

2. **Login**
   User need to enter these details for login over android and can login his account.

3. **Home**

   This is the main/default screen that user will land on after logging in to user will select the cricket then all matches of cricket will be shown. In header part, left side will be side bar menu and right side will be wallet or notifications, and when users click on these icons it will redirect to these sections.

4. Contest

At this section user can view the entire contest listings for the respective details like, Filter Contest By: Entry Fee Range, Contest Size(spots);the non Contest Listing includes Contest Name with logo, Entry Fee, Total winning amount of the contest, Winners Count, contest type (confirmed, single and multiple) .

5. Create Team & Join Contest

At this section user firstly select the matches in which they want to join. After match select by users, user will be move to contest screens in which user can check the all contest details according to prize pool, spots, winners and entry fees that already described above in contest screen section. After that user will select the pool that he want to join.

6. Private Contest

When it comes to private contest it enables Users to create a contest ("“Private contest"”) and invite other users, whether existing Users or otherwise, ("“Invited User"”) to create Teams and participate in the Contest(s). Users may create a Private contest to consist of a pre-specified number of Participants that is, consisting of either 2 -100 Participants.

7. Side Bar Menu

At this section, all side bar menu will be listed. Here I am listing all the fields in the side bar menu. Please check below:

- User Image
- User Email
- Home
- My Matches
- My Profile
- My Account
- Refer Friends
- Verify
- Notifications
- Live Score
- More

1. About Us
2. How to Play
3. Fantasy Points
4. FAQ's
5. Support
6. Terms & Conditions
7. Privacy Policy

- Logout (When user click on logout the confirmation popup will be open for logout).

8. My Matches
At this section, user will be able to view matches that he/she joined. User can check all the things in which match which contest he/she joined. In live match user can check the real time rankings and also check the prize break up and leader board.

9. My Profile
At this section, user will be able to view and update their profile details.
Here I am listing all the fields:

- Edit Button
- User Image(User can upload the image)
- Email Address
- Mobile Number
- Team Name(User can change team name only first time)
- DOB
- Gender(User can change the gender)
- State
- City
- Address

10. My Account
At this section, user will be able to view his wallet. In which user can check the total balance, deposited balance, winning balance, and bonus balance. User can check all the transactions by click on all transactions. Users can add balance through different-2 payment gateway and also can withdraw his winning amount.

11. Notifications
At this section, user can check all notifications.

12. Live Score
At this section, user can check all the matches live scores that he joined.

1.4.2 Non - Functional Requirements

1. Software must perform well for 100 live users
2. It should not eat up battery/ network which causes users to uninstall
3. Crashes should be recorded in crash analytics login provided to team

Chapter 2

2. FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

2.1 TECHNICAL STUDY

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs, and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to introduce the technical system. The application is the fact that it has been developed on windows 10 platform and high configuration of 8 GB RAM on Intel Pentium Dual core processor. This is technically feasible. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

2.2 OPERATIONAL STUDY

Operational feasibility is the measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility,

disposability, sustainability, affordability, and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development require appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of
systems engineering that needs to be an integral part of the early design phases.

2.3 BEHAVIORAL STUDY

Establishing the cost-effectiveness of the proposed system i.e., if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need of online social networking facilities. Thus, the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

Chapter 3 Database Design

3.1 Flowchart

Flowchart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing. The process of drawing a flowchart for an algorithm is known as “flowcharting”.

Basic Symbols used in Flowchart Designs

1. **Terminal:** The oval symbol indicates Start, Stop and Halt in a program’s logic flow. A pause/halt is generally used in a program logic under some error conditions. Terminal is the first and last symbols in the flowchart.

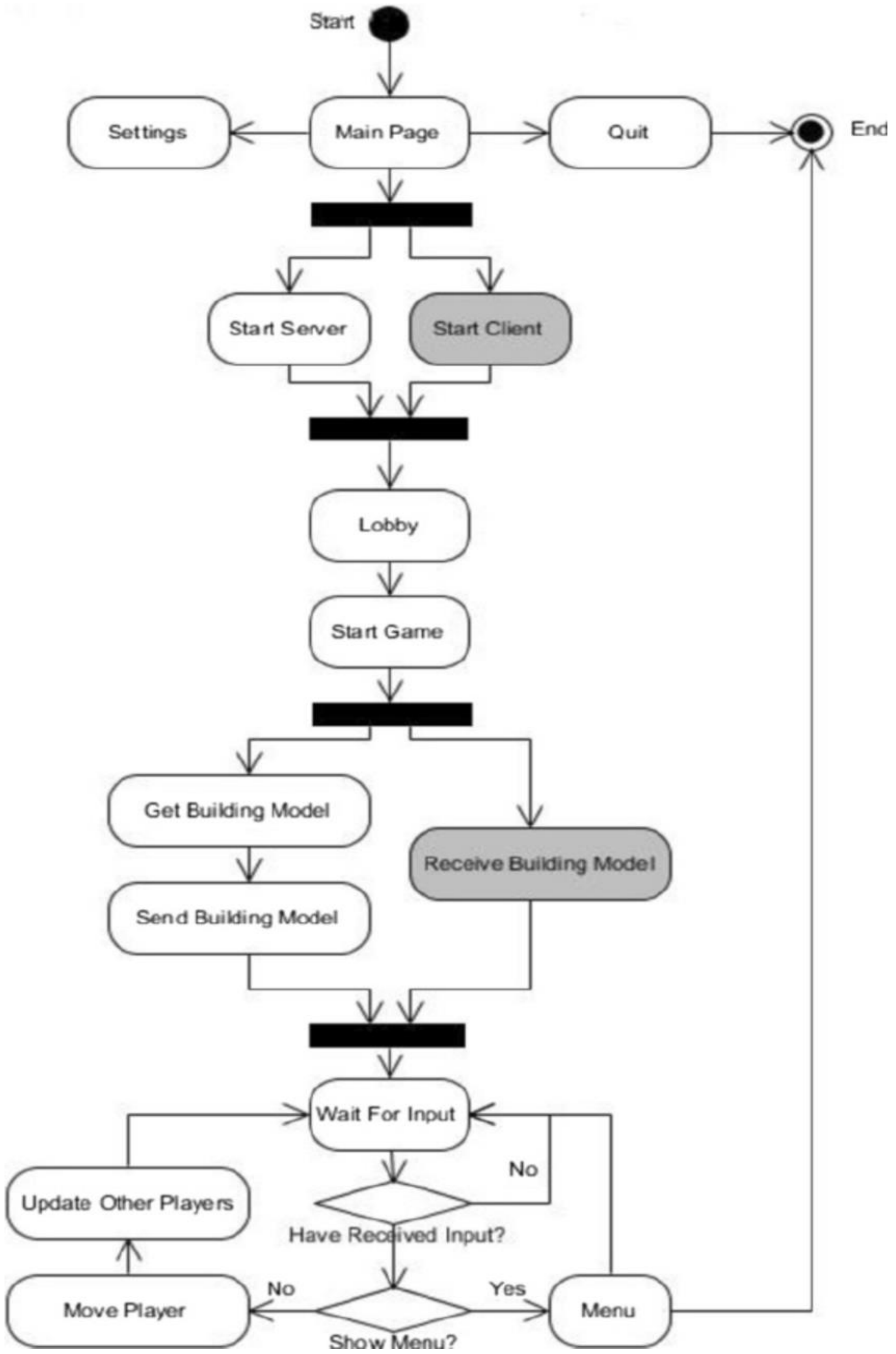
2. **Input/Output**: A parallelogram denotes any function of input/output type. Program instructions that take input from input devices and display output on output devices are indicated with parallelogram in a flowchart.

3. **Processing:** A box represents arithmetic instructions. All arithmetic processes such as adding, subtracting, multiplication and division are indicated by action or process symbol.

4. **Decision:** Diamond symbol represents a decision point. Decision based operations such as yes/no question or true/false are indicated by diamond in flowchart.

5. **Connectors:** Whenever flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.

6. **Flow lines:** Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of flow of control and relationship among different symbols of flowchart.



❖ ACTIVITY DIAGRAM

3.2 USE CASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve

Use case Diagram Components

To answer the question, "What is a use case diagram?" you need to first understand its building blocks. Common components include:

- **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

- **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

- **Goals:** The result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

Use case diagram symbols and notation

The notation for a use case diagram is straightforward and doesn't involve as many types of symbols as other UML diagrams.

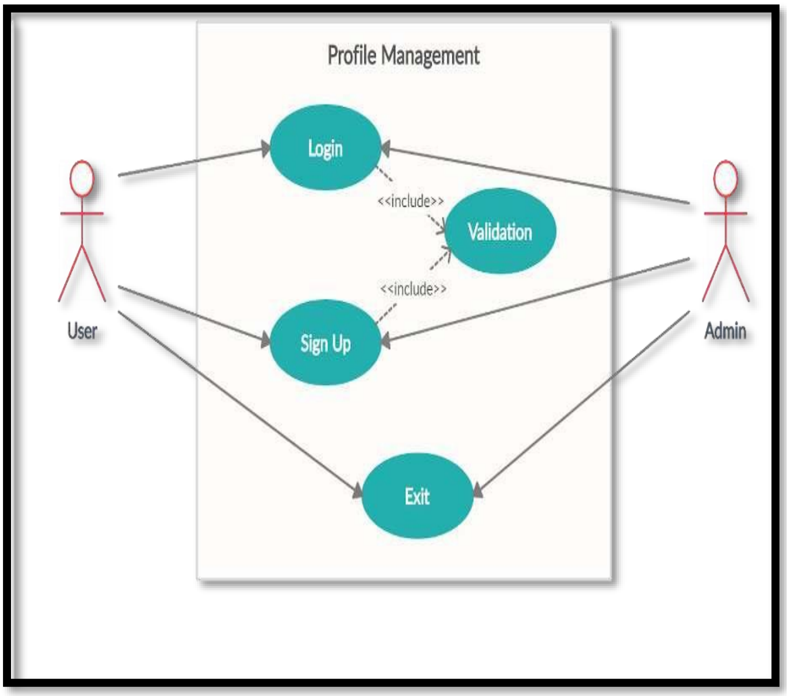
Use cases: Horizontally shaped ovals that represent the different uses that a user might have.

- **Actors:** Stick figures that represent the people employing the use cases.

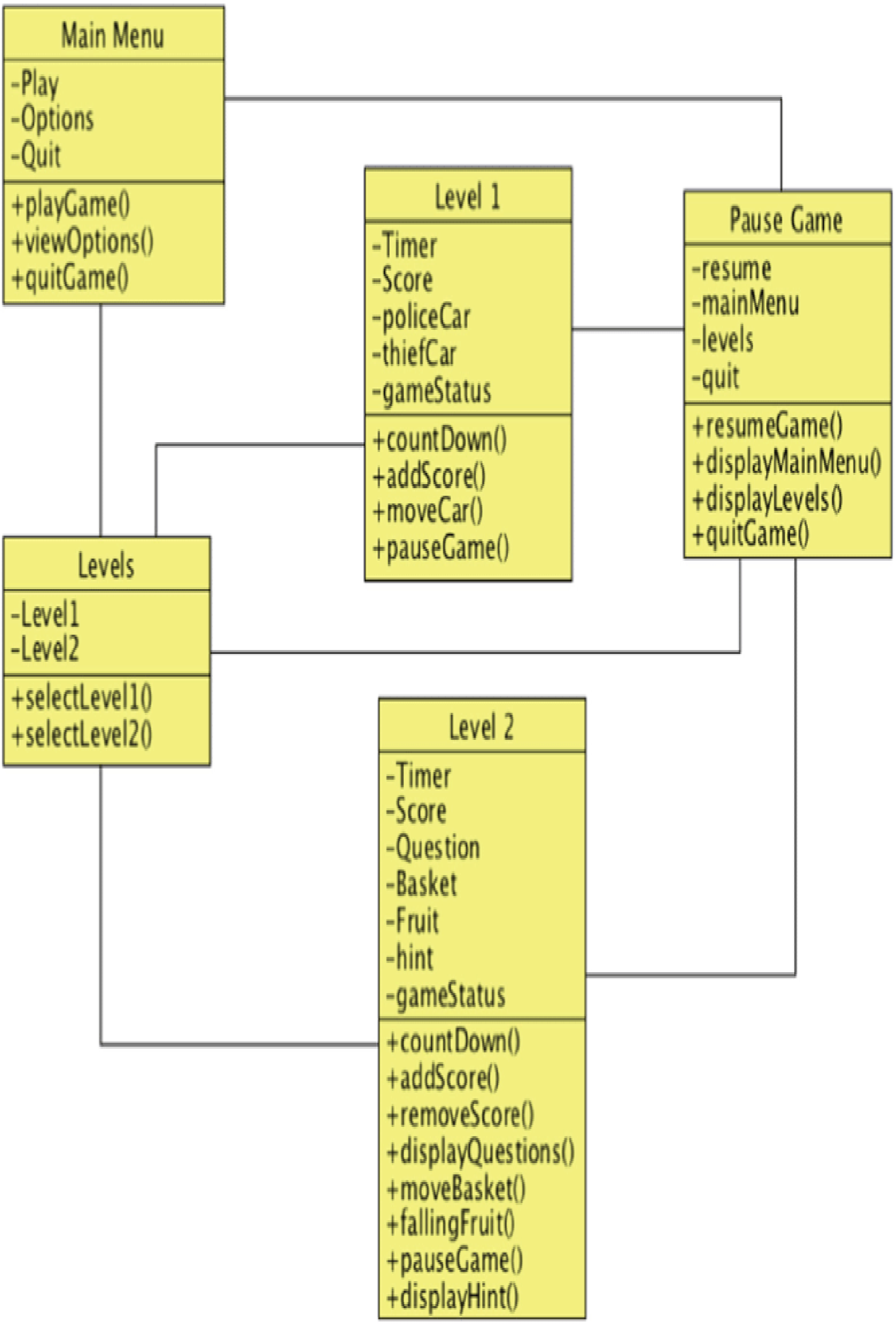
- **Associations:** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

- **System boundary boxes:** A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below.

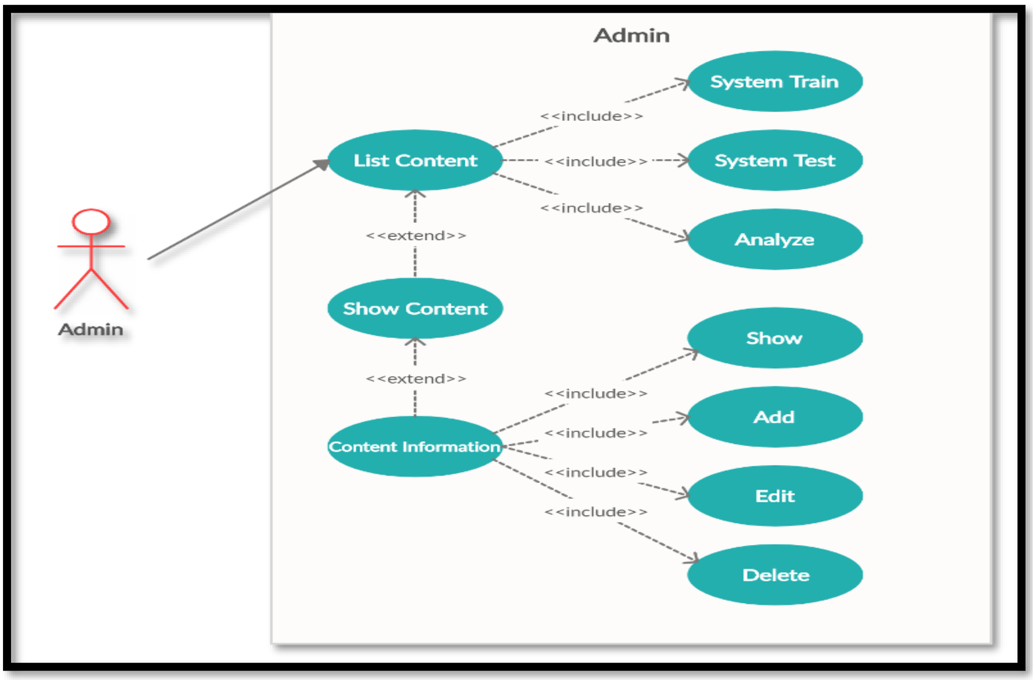
- **Packages:** A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.



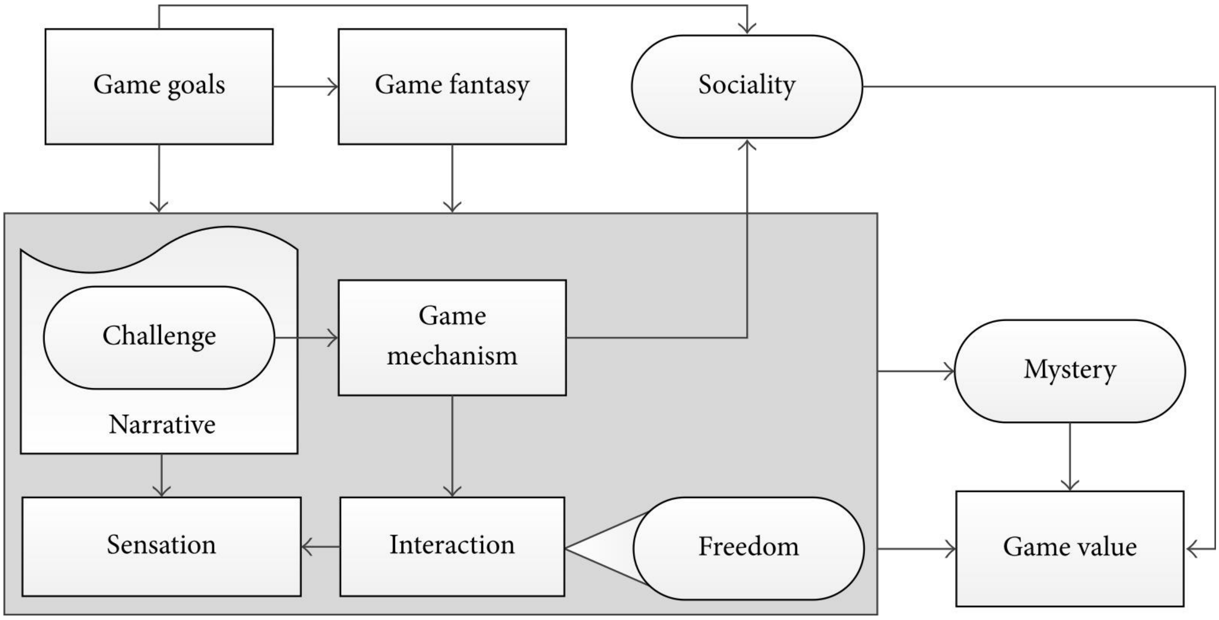
3.1.1 Use Case Diagram



CLASS DIAGRAM



3.1.3 Admin Use Case



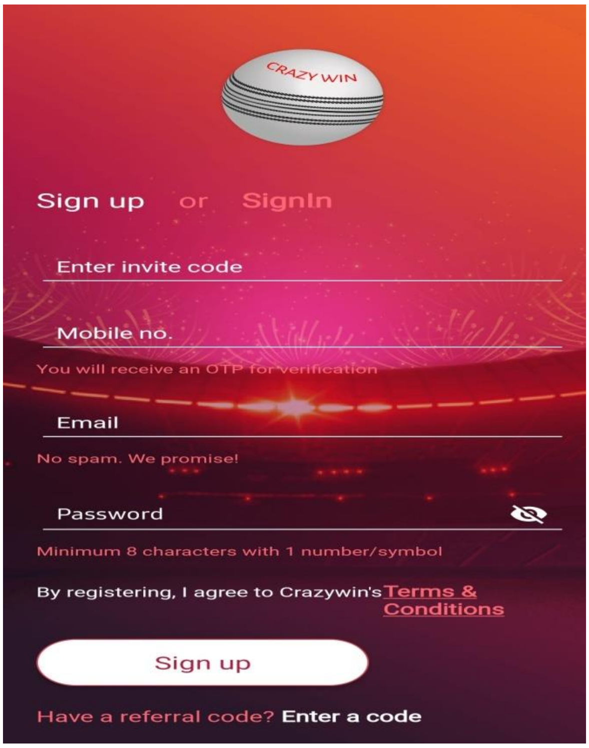
Chapter 4 Layout

Front-end Interface

8. Sign up

User need to enter the details for signup over android and can register his account.

- CrazyWin logo
- Profile photo icon(User can add his profile picture on clicking this icon, he can add from phone gallery and from his phone camera and profile photo will be optional)
- Email ID(User need to add a verify email ID)
- Phone Number(Phonenumberwillbeonly10digits)
- Password(Inpasswordfield,OneCapital,onenumericdigitandonesymbolshouldbemandatory)
- Referral Code (It will be optional, user can enter his referral code that sent by his friend. If he enter his referral code then he get some referral amount in his wallet as a bonus and also get some bonus to referred person that sent a code. Referral amount will be set from admin panel).
- One line for agree Terms & Conditions and Privacy Policy (When users clicks on these two Terms & Conditions and Privacy Policy then It will be redirected to these pages.
- Register Button
- LoginLink(Inwhichifuseralreadyhaveanaccountthenheneedtoclickonloginforenterhisaccount)



CRAZY WIN

Sign up or SignIn

Enter invite code

Mobile no.

You will receive an OTP for verification

Email

No spam. We promise!

Password

Minimum 8 characters with 1 number/symbol

By registering, I agree to Crazywin's Terms & Conditions

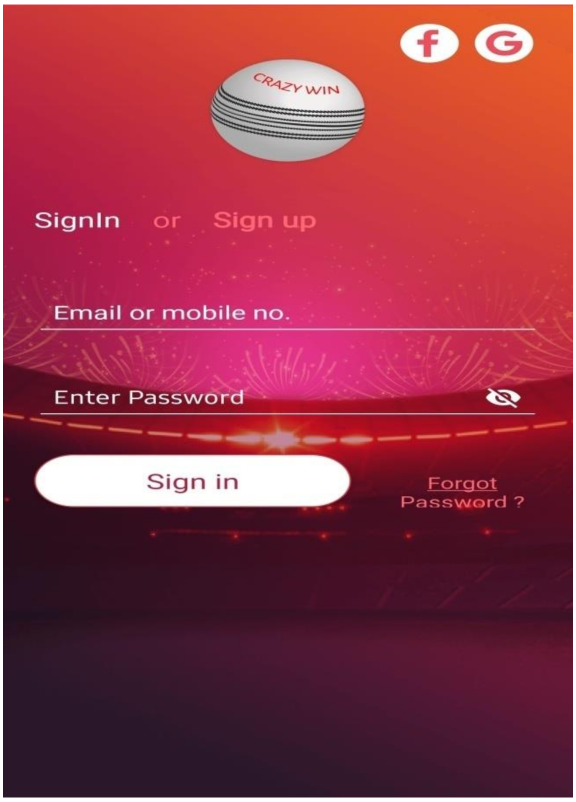
Sign up

Have a referral code? Enter a code

9. **Login**

User need to enter these details for login over android and can login his account.

- Crazy Win logo
- Login with Email and Mobile Number (User can login with email and phone number, If user enter his email id and click on login button then password screen will open then he can enter his password and can login. If he enter his mobile number then OTP screen will be open then he will enter OTP and can login his account)
- Login Button
- Forgot Password link (When user click on forgot password, the Forgot password screen will be open in which user need to enter his email and mobile number for OTP and can change the password. If user enter his mail id then OTP will be sent to his mail id and if he enter his mobile number then OTP will be sent to his mobile number and enter the opt and can change his password)
- Social Media Logins (User can login through Facebook and Google through his account. Users cannot login his Facebook account through mobile number)
- Create account link(When user will click on this link then create his new account)



10. Home

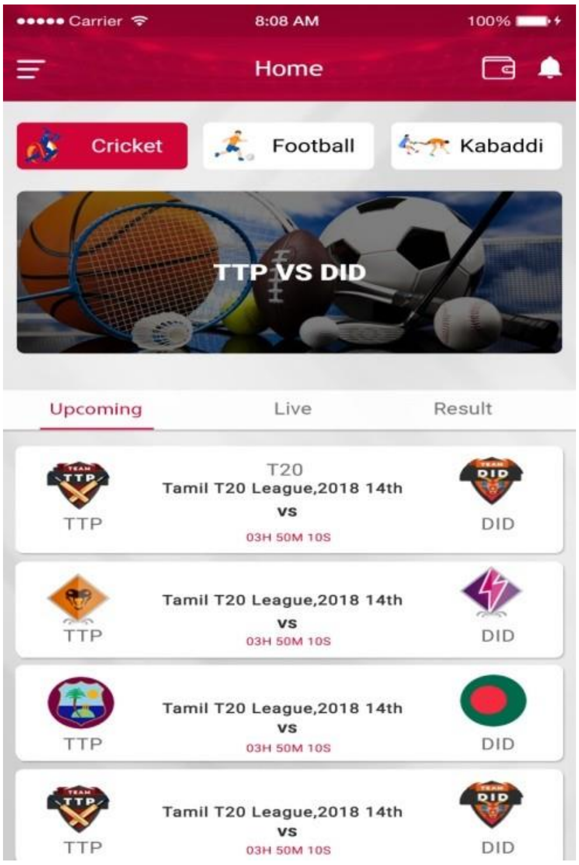
This is the main/default screen that user will land on after logging in to user will select the cricket then all matches of cricket will be shown. In header part, left side will be side bar menu and right side will be wallet or notifications, and when users click on these icons it will redirect to these sections.

There will be three types of filter that I am describing:

- **Upcoming** – In this section all upcoming matches will be shown. These matches will be created from admin panel. In upcoming matches all upcoming matches list will be shown, in which user can select the match for join contest. In these matches short team name with team logo will be show, in middle series name will be show and also mention the type of match like ODI, Test, T20 and will also show the remaining time of match to start and the format will be Day, hours, minutes and seconds. User can only join contest only from upcoming matches (fixtures). Once timing will be finished then match will be moved upcoming to live .Once match is in live user cannot join any contest.
- **Live** – In this section all live matches will be show. These live matches shifted from upcomingmatches.Inlivematchesusercancheckallthedetailslikejoinedcontest, live score, real time points and real time ranks. User can

check only these details only in the joined contest. Suppose if there not any matches in live then it will shown o live match currently.

- **Results** – Once the match completed and rank and prize distributed then it will move live section to results section and match status will be changed as completed. Suppose if any matches abundant and cancel then admin will cancel from admin panel and cancel status will be show at result section of a cancel match. Once match completed then user can check the match results and his ranking and prizes as well. User can check these only in joined contest matches.



11.Contest

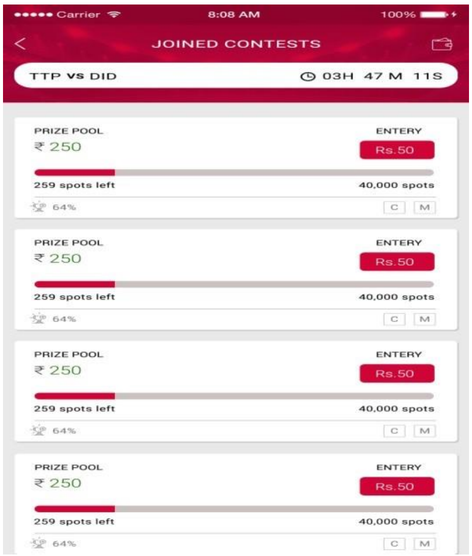
At this section user can view the entire contest listings for the respective details like, Filter Contest By: Entry Fee Range, Contest Size(spots);the non Contest Listing includes Contest Name with logo, Entry Fee, Total winning amount of the contest, Winners Count, contest type (confirmed, single and multiple) .

Here I am describing the filters in contest details, in header the both team name in short between the match and also will be show timer. If user click on entry fee and contest size then user can filter these things:

a. Prize pool(User can filter ascending or descending orders)
b. Spots(User can filter ascending or descending orders)
c. Winners(User can filter ascending or descending orders)
d. Entry(User can filter ascending or descending orders)

If user not create any team then users can create team by clicking over create team. If user already created a team then my team button will be show and user can check his own created team and also can check the team preview and edit team and also create a clone of createdteam.Whenusersclickonjoinedcontestthenhe/shecancheckallthe joinedcontestdetails.

Users can also create a private contest through this screen and can also join the contest by enter contest code.



●●●●● Carrier 8:08 AM 100%

JOINED CONTESTS

TTP vs DID 03H 47 M 11S

PRIZE POOL ENTERY
₹ 250 Rs.50
259 spots left 40,000 spots
64% C M

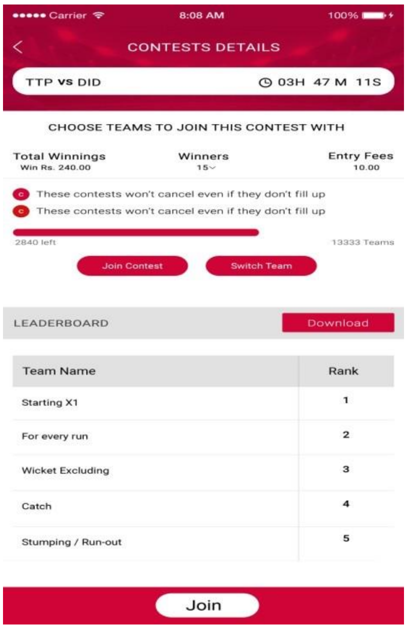
PRIZE POOL ENTERY
₹ 250 Rs.50
259 spots left 40,000 spots
64% C M

PRIZE POOL ENTERY
₹ 250 Rs.50
259 spots left 40,000 spots
64% C M

PRIZE POOL ENTERY
₹ 250 Rs.50
259 spots left 40,000 spots
64% C M

User can also check the contest details of a particular contest in which users can check the total winnings, total winners and entry fees. If user clicks on winners then total number of winnerswillbeshowninwhichusercancheckalltheranksandinwhichhowmuchprizehecanwin.

Users can download the leader board in a pdf format through which he/she can check the how many number of users join this contest. In leader board pdf all users team mentioned. User can also join the team and switch team from here.



●●●●● Carrier 8:08 AM 100%

CONTESTS DETAILS

TTP vs DID 03H 47 M 11S

CHOOSE TEAMS TO JOIN THIS CONTEST WITH

Total Winnings	Winners	Entry Fees
Win Rs. 240.00	15	10.00

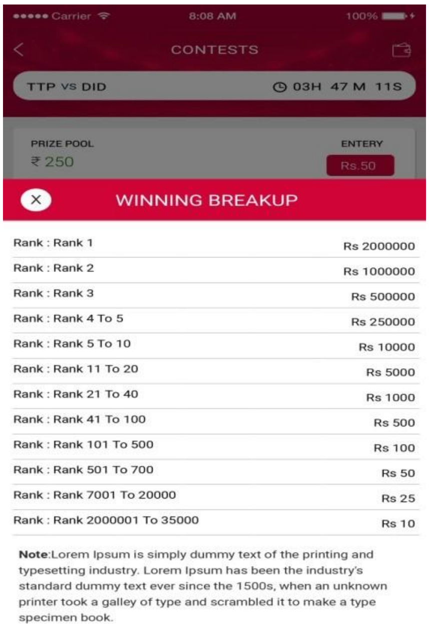
These contests won't cancel even if they don't fill up

These contests won't cancel even if they don't fill up

2840 left 13333 Teams

LEADERBOARD

Team Name	Rank
Starting X1	1
For every run	2
Wicket Excluding	3
Catch	4
Stumping / Run-out	5



●●●●● Carrier 8:08 AM 100%

CONTESTS

TTP VS DID 03H 47 M 11S

PRIZE POOL ₹ 250

ENTERY Rs.50

WINNING BREAKUP

Rank : Rank 1	Rs 2000000
Rank : Rank 2	Rs 1000000
Rank : Rank 3	Rs 500000
Rank : Rank 4 To 5	Rs 250000
Rank : Rank 5 To 10	Rs 10000
Rank : Rank 11 To 20	Rs 5000
Rank : Rank 21 To 40	Rs 1000
Rank : Rank 41 To 100	Rs 500
Rank : Rank 101 To 500	Rs 100
Rank : Rank 501 To 700	Rs 50
Rank : Rank 7001 To 20000	Rs 25
Rank : Rank 2000001 To 35000	Rs 10

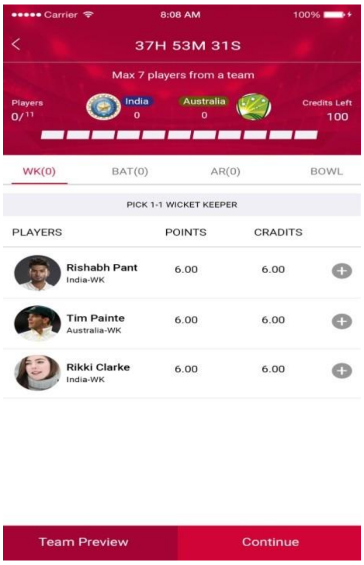
Note:Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

11. Create Team & Join Contest

At this section user firstly select the matches in which they want to join. After match select by users, user will be move to contest screens in which user can check the all contest details according to prize pool, spots, winners and entry fees that already described above in contest screen section. After that user will select the pool that he want to join.

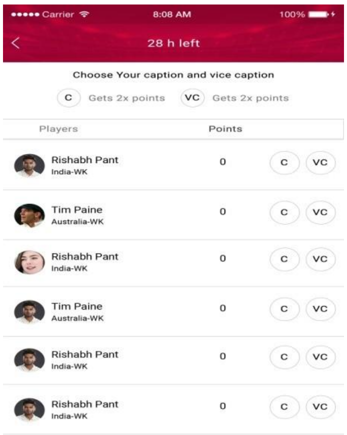
Users can create an own team by clicking over create team button. Users can select 11 players team under the 100 credits. Users can choose (1-3) Wicket-Keeper, (3-5) Batsman, (1-3) All-Rounder, (3-5) Bowler. User can select maximum 7 players from one team. After the team created user need to select Captain and Vice-Captain. Once the team created then user can edithisteam,previewhisteam(inthisoneteamplayerwillshowinredcolourandotherteamplayerwillbeshowinbluecolourandalsointhisscreeninplayerspointswillalsobedisplayandtheformatsofplayerwillbeshowasWicket-keeperinabovethenbatsmanthenall-roundersandin below will be show the bowlers) and can make clone of created team or he/she can create a new team. One user can create maximum 6 teams(In future we can increase a team)

Once team created, user will join a contest by clicking over join contest. User will click over join contest the confirmation pop will be open in which show the user total balance, entry fees of contest, how much cash bonus amount users to join contest and how much amount to pay by user to join contest. Once user click on Join contest the confirmation will be show as “Joined Contest Successfully”.
Users can check the joined contest detail on click over a contest. User can also check from My Matches.



PICK 1-1 WICKET KEEPER

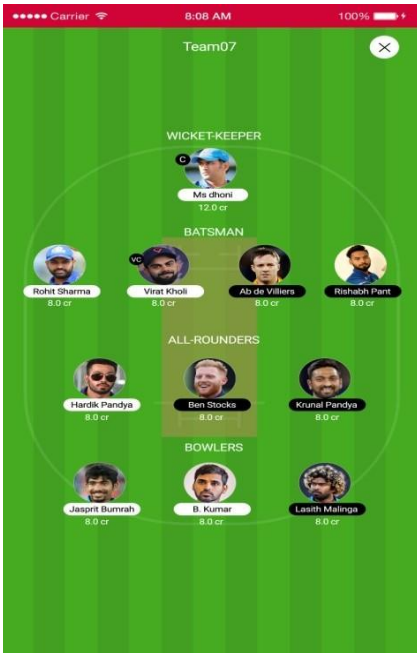
PLAYERS	POINTS	CRADITS
Rishabh Pant India-WK	6.00	6.00
Tim Painte Australia-WK	6.00	6.00
Rikki Clarke India-WK	6.00	6.00

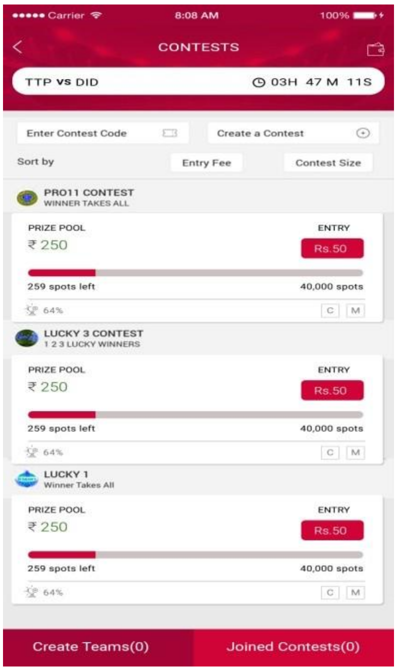


Choose Your caption and vice caption

C Gets 2x points VC Gets 2x points

Players	Points		
Rishabh Pant India-WK	0	C	VC
Tim Paine Australia-WK	0	C	VC
Rishabh Pant India-WK	0	C	VC
Tim Paine Australia-WK	0	C	VC
Rishabh Pant India-WK	0	C	VC
Rishabh Pant India-WK	0	C	VC





PRO11 CONTEST
WINNER TAKES ALL

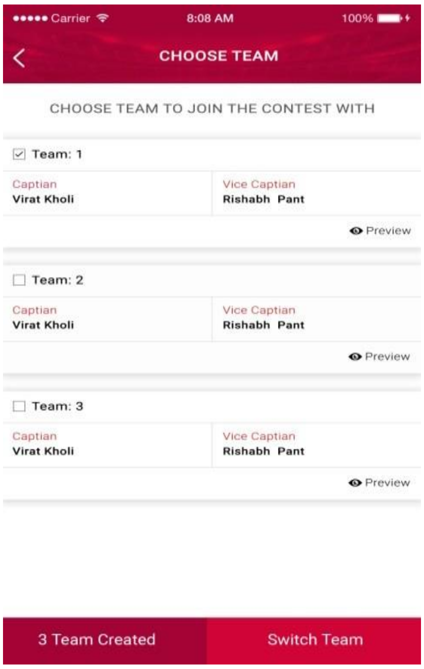
PRIZE POOL ₹ 250 ENTRY Rs.50
259 spots left 40,000 spots
64% C M

LUCKY 3 CONTEST
1 2 3 LUCKY WINNERS

PRIZE POOL ₹ 250 ENTRY Rs.50
259 spots left 40,000 spots
64% C M

LUCKY 1
Winner Takes All

PRIZE POOL ₹ 250 ENTRY Rs.50
259 spots left 40,000 spots
64% C M



●●●●● Carrier 8:08 AM 100%

CHOOSE TEAM

CHOOSE TEAM TO JOIN THE CONTEST WITH

☑ Team: 1

Captian
Virat Kholi

Vice Captian
Rishabh Pant

Preview

□ Team: 2

Captian
Virat Kholi

Vice Captian
Rishabh Pant

Preview

□ Team: 3

Captian
Virat Kholi

Vice Captian
Rishabh Pant

Preview

3 Team Created | Switch Team

12. Private Contest

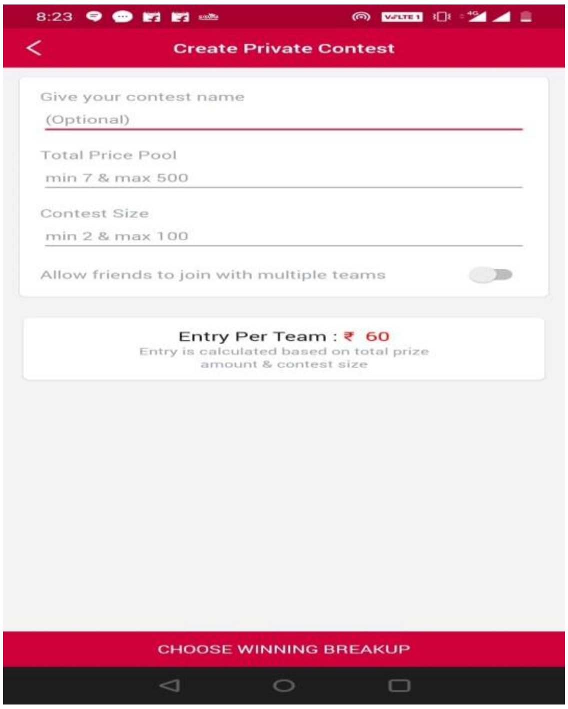
When it comes to private contest it enables Users to create a contest (""Private contest"") and invite other users, whether existing Users or otherwise, (""Invited User"") to create Teams and participate in the Contest(s). Users may create a Private contest to consist of a pre-specified number of Participants that is, consisting of either 2 -100 Participants. The User creating the Private contest shall pay the pre-designated amount for such Private contest and thereby join that Private contest and shall supply a name for the Private contest and be provided with a unique identification code ("contest Code") (which will be issued to the account of such User).The User agrees and understands that once the Private contest is created no change shall be permitted in the terms or constitution of the Private contest, except for a change in the name of the contest. The User can share the private contest code via whatsApp or other social media app.

The private contest fields will be showing like:

- Contest Name

- Total Price Pool
- Contest Size (min 2 and max 100, we already have set in admin panel that I described in admin panel)
- Entry Per Team(It depends on user winner)

Here I am describing the private contest formula through which everyone calculate the entry fee per team.



Create Private Contest

Give your contest name
(Optional)

Total Price Pool
min 7 & max 500

Contest Size
min 2 & max 100

Allow friends to join with multiple teams

Entry Per Team : ₹ 60
Entry is calculated based on total prize amount & contest size

13. Side Bar Menu

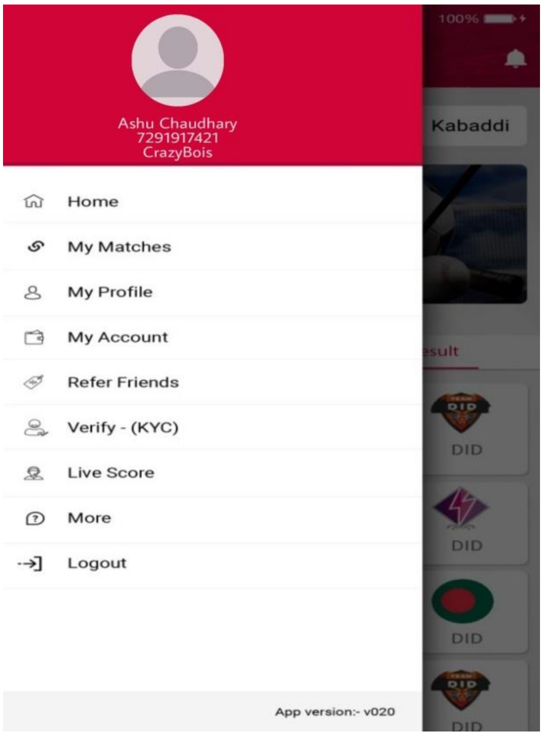
At this section, all side bar menu will be listed. Here I am listing all the fields in the side bar menu. Please check below:

- User Image
- User Email
- Home
- My Matches
- My Profile
- My Account
- Refer Friends
- Verify
- Notifications
- Live Score
- More

  8. About Us

9. How to Play
10. Fantasy Points
11. FAQ’s
12. Support
13. Terms & Conditions
14. Privacy Policy

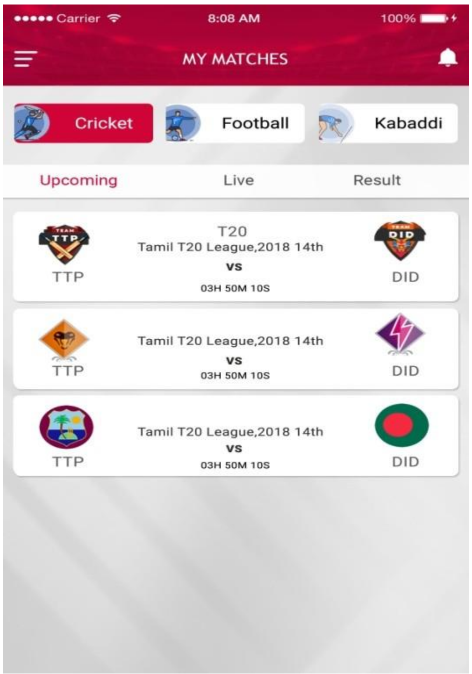
- Logout(When user click on logout the confirmation popup will be open for logout)
- AppVersion(Itwillbeshowonlyinappandthecurrentversionofappwillbeshow)

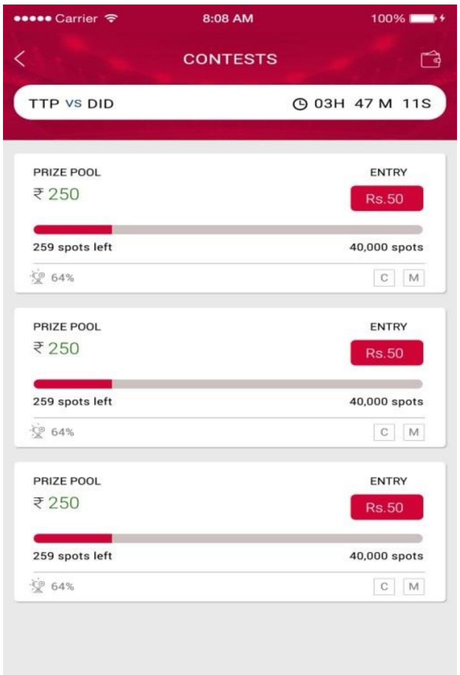


8. My Matches

At this section, user will be able to view matches that he/she joined. User can check all the things in which match which contest he/she joined. In live match user can check the real time rankings and also check the prize break up and leader board

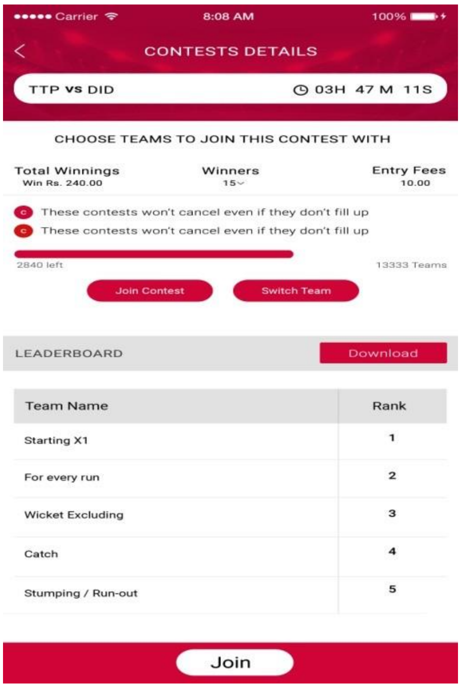
Note: In this section firstly user need to select the sports type (cricket, football an kabaddi)after that he/she can check the all upcoming, live and results section that I already have described in home screen .In this screen users can check only the joined contests or pools .All things will be show similar to home screens but only difference In this screen here will be show only joined contest(joined matches)that was joined by users.





When user click on prize breakup, the all ranking and prize will be shown that was set from admin panel for this match.

When user click on leader board, the all players ranking will be displayed from first rank to last rank and in live matches it will be showing in real time. Suppose I joined any contest and my current rank is 12 then in first entry my rank will be show after that all rank will be show from1stto last rank and user can download the leader board.



CHOOSE TEAMS TO JOIN THIS CONTEST WITH

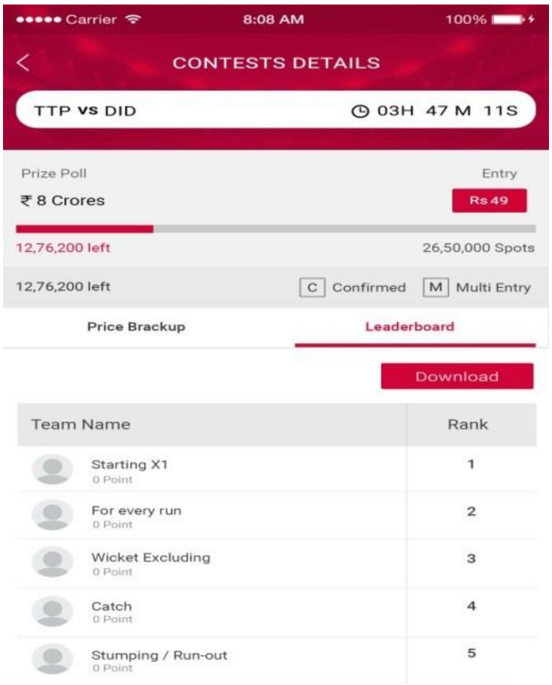
Total Winnings	Winners	Entry Fees
Win Rs. 240.00	15	10.00

These contests won't cancel even if they don't fill up

These contests won't cancel even if they don't fill up

2840 left 13333 Teams

Team Name	Rank
Starting X1	1
For every run	2
Wicket Excluding	3
Catch	4
Stumping / Run-out	5



Prize Poll: ₹ 8 Crores

Entry: Rs 49

12,76,200 left 26,50,000 Spots

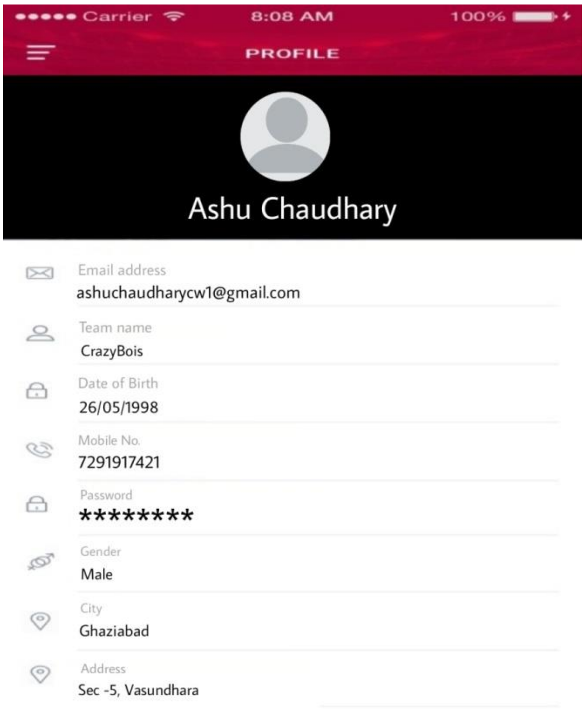
12,76,200 left C Confirmed M Multi Entry

Team Name	Rank
Starting X1 0 Point	1
For every run 0 Point	2
Wicket Excluding 0 Point	3
Catch 0 Point	4
Stumping / Run-out 0 Point	5

9. My Profile

At this section, user will be able to view and update their profile details. Here I am listing all the fields:

- Edit Button
- User Image(User can upload the image)
- Email Address
- Mobile Number
- Team Name(User can change team name only first time)
- DOB
- Gender(User can change the gender)
- State
- City
- Address



Ashu Chaudhary

Email address
ashuchaudharycw1@gmail.com

Team name
CrazyBois

Date of Birth
26/05/1998

Mobile No.
7291917421

Password
****

Gender
Male

City
Ghaziabad

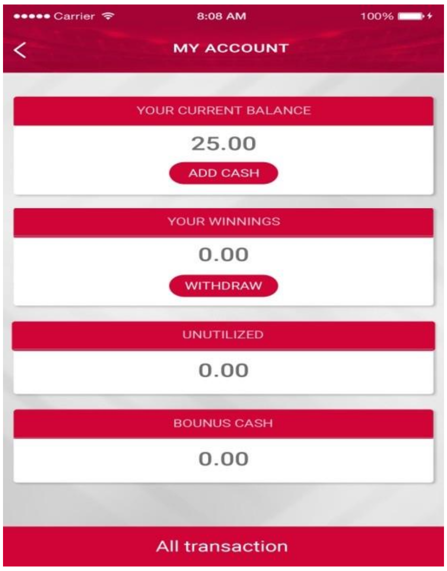
Address
Sec -5, Vasundhara

User can edit his profile on clicking edit button. In which user can upload profile picture. User can change Team name only first time. User can change the gender.

User can change his password on clicking the change password button, in which user need to enter his old password then new password then re-enter the new password and click on update button the password will be changed successfully.

13. My Account

At this section, user will be able to view his wallet. In which user can check the total balance, deposited balance, winning balance, and bonus balance. User can check all the transactions by click on all transactions. Users can add balance through different-2 payment gateway and also can withdraw his winning amount.



YOUR CURRENT BALANCE
25.00

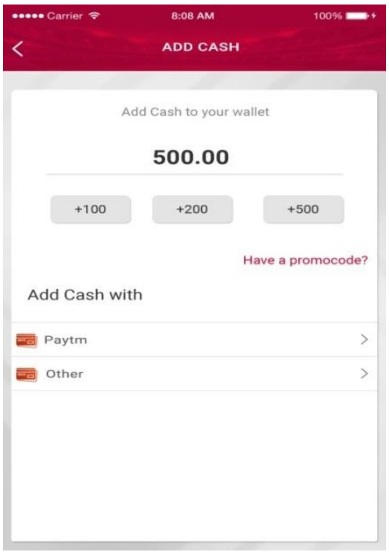
YOUR WINNINGS
0.00

UNUTILIZED
0.00

BOUNUS CASH
0.00

Here I am describing all the sections one by one Please check below:

- **Your Current Balance (Add Cash)**
  In this section user can check his total balance and can also add balance by clicking over add cash button. User can add cash by using different-2 payment gateways. After enter the amount user need to select payment gateway option. User can also the apply promo code by clicking over promo code. User need to enter promo code and click on apply button then promo code will be applied or if reject then will show the error message.
  After that user will click on continue button it will be redirect to third party page where user can provide his card details, wallet details, net banking, etc and can add the cash in his wallet.



ADD CASH

Add Cash to your wallet

500.00

Add Cash with

- **Your Winnings (Withdraw)**
  In this section user can check his total winning that earn from win contest and can also withdraw his winning amount. User need to click on withdraw button and enter the withdraw amount and click on submit button. The withdraw request will be sent to admin panel. Once the withdraw request approved from admin panel after that admin will send his with draw amount on users bank account and amount will be send by admin manually. We are not using any payout API now. We have set the minimum withdraw amount 200 from coding end. Once user withdraw his amount then that amount will be deducted in his winning balance.

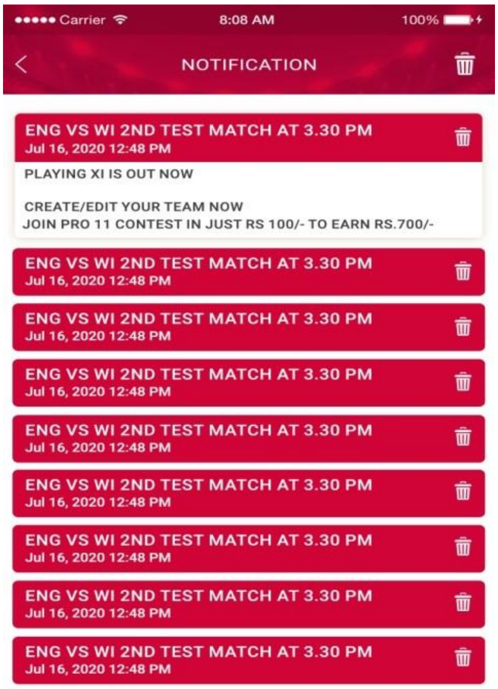
- **Unutilized Balance**
  When It comes to unutilized balance it is a balance that adds by user.

- **Bonus Cash**
  When It comes to bonus cash it is a balance that user earn by referral.

- **All Transactions**
  In this section user can check his all transactions in which all debit and credit history records will be show. In this section if user joins any contest then that amount also be shown. Every debit and credit amount will be show here.

14. Notifications

At this section, user can check all notifications. If user click on a particular notification then that notification details will be open in which user can check content and images. User can delete a particular notification by clicking over delete icon. If user wants to delete all notifications then he/she need to click on delete icon that will be show in header, once click on that icon the pop up will be open if user click on yes then all notifications will be deleted.



ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM
PLAYING XI IS OUT NOW

CREATE/EDIT YOUR TEAM NOW
JOIN PRO 11 CONTEST IN JUST RS 100/- TO EARN RS.700/-

ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM

ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM

ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM

ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM

ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM

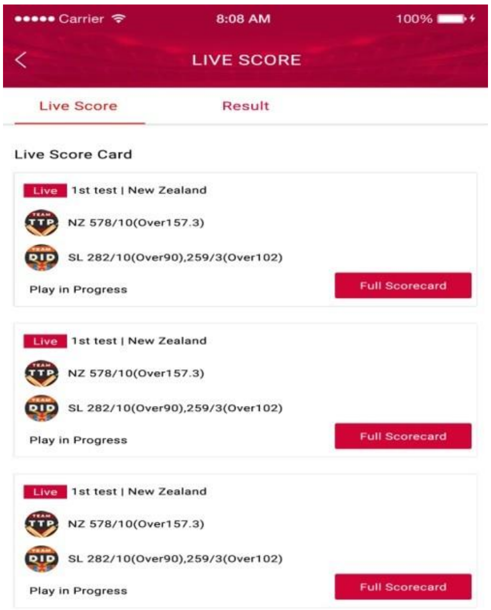
ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM

ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM

ENG VS WI 2ND TEST MATCH AT 3.30 PM
Jul 16, 2020 12:48 PM

15. Live Score

At this section, user can check all the matches live scores that he joined. Users can check all matches live scores with full scoreboard. User can also check the results scoreboard of joined matches.

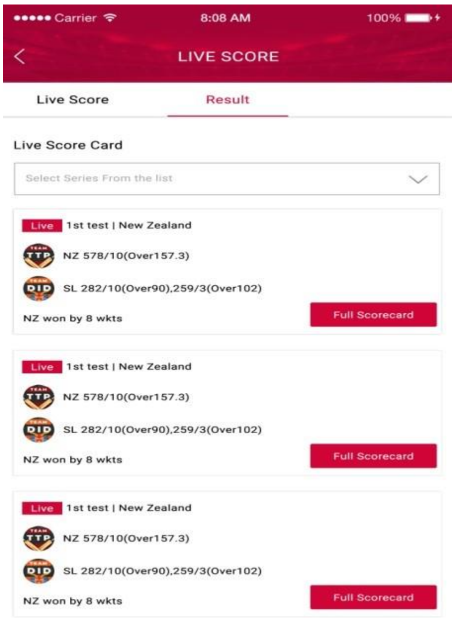


Live Score Card

Live 1st test | New Zealand
NZ 578/10(Over157.3)
SL 282/10(Over90),259/3(Over102)
Play in Progress

Live 1st test | New Zealand
NZ 578/10(Over157.3)
SL 282/10(Over90),259/3(Over102)
Play in Progress

Live 1st test | New Zealand
NZ 578/10(Over157.3)
SL 282/10(Over90),259/3(Over102)
Play in Progress



Live Score Card

Live 1st test | New Zealand
NZ 578/10(Over157.3)
SL 282/10(Over90),259/3(Over102)
NZ won by 8 wkts

Live 1st test | New Zealand
NZ 578/10(Over157.3)
SL 282/10(Over90),259/3(Over102)
NZ won by 8 wkts

Live 1st test | New Zealand
NZ 578/10(Over157.3)
SL 282/10(Over90),259/3(Over102)
NZ won by 8 wkts

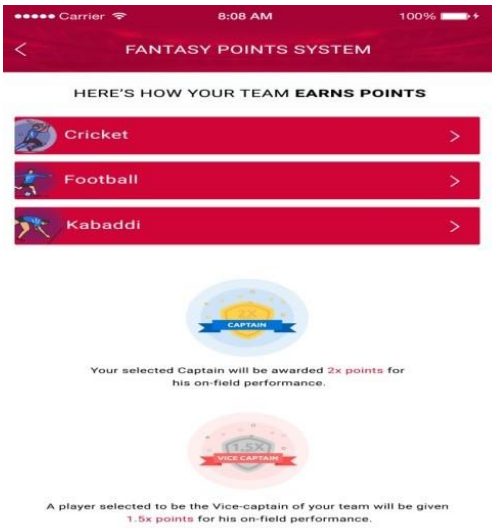
16. CMS Pages

In this section, user can check all the CMS pages. Here I am listing all the pages.

- **About Us**
- In this page user can check about company details that uploaded by admin.

- **How to Play**
- In this page user can check the process for game play in which how to play content will be shown.

- **Fantasy Points**
- In this page user can check the fantasy point system for different-2 fantasy sports. All content will be uploaded from admin panel.



FANTASY POINTS SYSTEM

HERE'S HOW YOUR TEAM **EARNS POINTS**

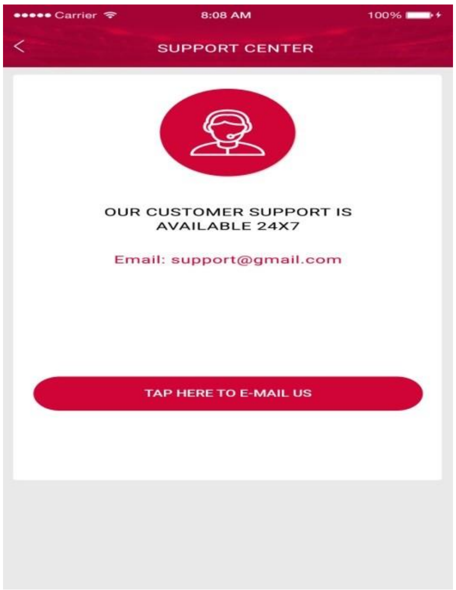
Your selected Captain will be awarded 2x points for his on-field performance.

A player selected to be the Vice-captain of your team will be given 1.5x points for his on-field performance.

FAQ's

In this page user can check the frequently asked content.

- **Support**
  In this page user can contact to customer support. When user click on Email us then it will be redirect to email drafting page and when user click on call us on then it will be redirect to phone dial page.



SUPPORT CENTER

OUR CUSTOMER SUPPORT IS AVAILABLE 24X7

Email: support@gmail.com

- **Terms & Conditions**
  In this page user can check the terms & conditions content.

- **Privacy Policy**
  In this page user can check the privacy policy content.

SPLASH SCREEN:

```
package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.content.pm.Signature;
import android.os.Bundle;
import android.os.Handler;

import androidx.appcompat.app.AppCompatActivity;

import android.util.Base64;
import android.util.Log;

import com.elevendreamer.R;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import static
com.elevendreamer.APICallingPackage.Constants.SPLASH_TIME_OUT;


public class SplashScreen extends AppCompatActivity {

    SplashScreen activity;
    Context context;

    private SharedPreferences loginPreferences;
    private Boolean saveLogin;
    private SharedPreferences.Editor loginPrefsEditor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        context = activity = this;
        printHashKey(context);

        loginPreferences = getSharedPreferences("loginPrefs",
MODE_PRIVATE);
        loginPrefsEditor = loginPreferences.edit();
        saveLogin = loginPreferences.getBoolean("saveLogin", false);
        new Handler().postDelayed(new Runnable() {


            @Override
            public void run() {

                if (saveLogin == true) {
                    HomeScreen();
                } else {
                    LoginScreen();
                }
            }
        }, SPLASH_TIME_OUT);

    }

    public void LoginScreen() {
        Intent i = new Intent(SplashScreen.this, MainActivity.class);
        startActivity(i);
    }

    public void HomeScreen() {
        Intent i = new Intent(SplashScreen.this, HomeActivity.class);
        startActivity(i);
    }


    public static void printHashKey(Context pContext) {
        try {
            PackageInfo info =
pContext.getPackageManager().getPackageInfo(pContext.getPackageName(),
                    PackageManager.GET_SIGNATURES);
            for (Signature signature : info.signatures) {
                MessageDigest md = MessageDigest.getInstance("SHA");
                md.update(signature.toByteArray());
                String hashKey = new String(Base64.encode(md.digest(), 0));
                Log.i("TAG", "printHashKey() Hash Key: " + hashKey);
            }
        } catch (NoSuchAlgorithmException e) {
            Log.e("TAG", "printHashKey()", e);
        } catch (Exception e) {
            Log.e("TAG", "printHashKey()", e);
        }
    }
}
```

MAIN ACTIVITY JAVA CODE :

```
package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;

import android.view.View;

import com.elevendreamer.R;
import com.elevendreamer.databinding.ActivityMainBinding;

public class MainActivity extends AppCompatActivity {


    Context context;
    MainActivity activity;

    ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding=DataBindingUtil.setContentView(this,
R.layout.activity_main);
        context = activity = this;



        binding.LLStartLogin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(activity,LoginActivity.class);
                startActivity(i);
            }
        });

        binding.LLStartSignUp.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new 
Intent(activity,RegistrationActivity.class);
                i.putExtra("Reffered","Yes");
                startActivity(i);
            }
        });

        binding.tvLetsPlay.setOnClickListener(new 
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new 
Intent(activity,RegistrationActivity.class);
                i.putExtra("Reffered","No");
                startActivity(i);
            }
        });



    }



    @Override
    public void onBackPressed() {


        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);


    }
}
```

HOME PAGE :

```
package com.elevendreamer.activity;

import android.Manifest;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.os.StrictMode;
import androidx.annotation.NonNull;

import com.elevendreamer.BuildConfig;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityHomeBinding;
import com.google.android.material.tabs.TabLayout;
import androidx.core.app.ActivityCompat;
import androidx.databinding.DataBindingUtil;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.core.content.ContextCompat;
import androidx.core.content.FileProvider;
import androidx.appcompat.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.TextView;
import android.widget.Toast;

import com.facebook.login.LoginManager;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.common.api.Status;
import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.fragment.HomeFragment;
import com.elevendreamer.fragment.MoreFragment;
import com.elevendreamer.fragment.MyContestFragment;
import com.elevendreamer.fragment.ProfileFragment;

import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.URL;
import java.net.URLConnection;
import java.util.ArrayList;
import java.util.List;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.APKNAME;
import static com.elevendreamer.APICallingPackage.Config.APKURL;
import static com.elevendreamer.APICallingPackage.Config.UPDATEAPP;
import static com.elevendreamer.APICallingPackage.Constants.UPDATEAPPTYPE;

public class HomeActivity extends AppCompatActivity implements
GoogleApiClient.OnConnectionFailedListener,ResponseManager {
    private int[] tabIcons = {
            R.drawable.home_icon,
            R.drawable.contest_icon,
            R.drawable.profile_icon,
            R.drawable.more_icon
    };
    Context context;
    HomeActivity activity;
    ResponseManager responseManager;
    APIRequestManager apiRequestManager;
    //Auto Login
    private SharedPreferences loginPreferences;
    private Boolean saveLogin;
    private SharedPreferences.Editor loginPrefsEditor;
    public static GoogleApiClient mGoogleApiClient;
    Typeface LatoBold,LatoRegular,Ravenscroft;
    public static SessionManager sessionManager;

    int progressStatus = 0;

    public static final int DIALOG_DOWNLOAD_PROGRESS = 0;
    private ProgressDialog pDialog;

    String[] permissions = new String[]{
            Manifest.permission.WRITE_EXTERNAL_STORAGE,
            Manifest.permission.READ_EXTERNAL_STORAGE
    };


    String APP_BACKUP_FOLDER = Environment.getExternalStorageDirectory()
            + File.separator + "11dreamer" + File.separator + "App";
     Dialog dialog;

     ActivityHomeBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = DataBindingUtil.setContentView(this,R.layout.activity_home);

        context = activity = this;
        sessionManager = new SessionManager();
```

```
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);


        StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build());



        Animation shake = AnimationUtils.loadAnimation(activity, R.anim.shake);
        binding.imNotification.startAnimation(shake);
```

REGISTRATION .JAVA :

```
package com.elevendreamer.activity;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import androidx.annotation.NonNull;
import com.elevendreamer.APICallingPackage.Config;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityRegitrationBinding;

import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;

import android.util.Log;
import android.view.View;
import android.view.WindowManager;

import com.facebook.AccessToken;
import com.facebook.CallbackManager;
import com.facebook.FacebookCallback;
import com.facebook.FacebookException;
import com.facebook.FacebookSdk;
import com.facebook.GraphRequest;
import com.facebook.GraphResponse;
import com.facebook.login.LoginManager;
import com.facebook.login.LoginResult;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.common.api.Status;
import com.google.firebase.iid.FirebaseInstanceId;
import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Class.Validations;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.UserDetails;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.LOGIN;
import static com.elevendreamer.APICallingPackage.Config.SIGNUP;
import static com.elevendreamer.APICallingPackage.Constants.LOGINTYPE;
import static com.elevendreamer.APICallingPackage.Constants.SIGNUPTYPE;

public class RegistrationActivity extends AppCompatActivity implements
ResponseManager, GoogleApiClient.OnConnectionFailedListener  {

    ResponseManager responseManager;
    APIRequestManager apiRequestManager;

    Context context;
    RegistrationActivity activity;
    String MobileNumber,EmailId,Password,ReferralCode;
    String Reffered;



    CallbackManager callbackManager;
    AccessToken accessTokan;



    private final String TAG = "RegistrationActivity";
    private static final int RC_SIGN_IN = 007;
    private GoogleApiClient mGoogleApiClient;

    String LoginType = "Normal";



    //Auto Login
    private SharedPreferences loginPreferences;
    private Boolean saveLogin;
    private SharedPreferences.Editor loginPrefsEditor;

    //SMS Permission
    String[] permissions = new String[]{
            android.Manifest.permission.READ_SMS,
            android.Manifest.permission.RECEIVE_SMS
    };

    SessionManager sessionManager;

    ActivityRegitrationBinding binding;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        callbackManager = CallbackManager.Factory.create();
        binding=DataBindingUtil.setContentView(this, R.layout.activity_regitration);

this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_ST
ATE_ALWAYS_HIDDEN);
        context = activity = this;
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        initViews();
        sessionManager = new SessionManager();

        loginPreferences = getSharedPreferences("loginPrefs", MODE_PRIVATE);
        loginPrefsEditor = loginPreferences.edit();
        saveLogin = loginPreferences.getBoolean("saveLogin", false);

        Reffered = getIntent().getStringExtra("Reffered");



        if (Reffered.equals("Yes")){
            binding.inputRegRefCode.setVisibility(View.VISIBLE);
            binding.LLFaceGoogle.setVisibility(View.GONE);
        }
        else {
            binding.inputRegRefCode.setVisibility(View.GONE);
            binding.LLFaceGoogle.setVisibility(View.VISIBLE);
        }

        binding.tvRegNext.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
    public void onClick(View view) {

        MobileNumber = binding.etMobileNo.getText().toString();
        EmailId = binding.etEmail.getText().toString();
        Password = binding.etPassword.getText().toString();
        if (Reffered.equals("Yes")){
            ReferralCode = binding.etReferralCode.getText().toString();
        }
        else {
            ReferralCode = "";
        }

        if (MobileNumber.equals("")){
            ShowToast(context,"Enter Mobile Number");
            binding.etMobileNo.requestFocus();
        }
        else if (!MobileNumber.matches(Validations.MobilePattern)){
            binding.etMobileNo.requestFocus();
            ShowToast(context,"Enter Valid Mobile Number");
        }
        else if (EmailId.equals("")){
            binding.etEmail.requestFocus();
            ShowToast(context,"Enter Email Id");

        } else if(!Validations.isValidEmail(EmailId)){
            binding.etEmail.requestFocus();
            ShowToast(context,"Enter Valid Email Id");
        }
        else if (Password.equals("")){
```

```
                    binding.etPassword.requestFocus();
                    ShowToast(context,"Enter Password");
                }
                else if (Password.length()<8&& !Validations.isValidPassword(Password)){

                    ShowToast(context,"Password Pattern Not Macthed");
                }
                else {
                    LoginType = "Normal";

                    if (Build.VERSION.SDK_INT >= 23) {
                        checkPermissions();
                    } else {
                        callSignupApi(true);
                    }

                }

            }
        });



        binding.RLFBLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                LoginType = "Email";
                //fb_login_button.performClick();
                initFbObject(context);
            }
        });
        binding.RLGmailLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                LoginType = "Email";
                Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
                startActivityForResult(signInIntent, RC_SIGN_IN);
            }
        });
        binding.LLEnterCode.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(activity,RegistrationActivity.class);
                i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                i.putExtra("Reffered","Yes");
                startActivity(i);
            }
        });

        binding.tvSignInText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(activity,LoginActivity.class);
                i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(i);
            }
        });
        binding.fbLoginButton.setReadPermissions("email");
        binding.fbLoginButton.setReadPermissions("public_profile");
        binding.fbLoginButton.setReadPermissions("user_birthday");
        binding.fbLoginButton.setReadPermissions("user_friends");

        //G+

        GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestEmail()
                .build();

        mGoogleApiClient = new GoogleApiClient.Builder(this)
                .enableAutoManage(this, this)
                .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                .build();

        // Customizing G+ button
        binding.btnSignIn.setSize(SignInButton.SIZE_STANDARD);
        binding.btnSignIn.setScopes(gso.getScopeArray());



        binding.btnSignIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
                startActivityForResult(signInIntent, RC_SIGN_IN);
            }
        });
        binding.tvTearmsandConditions.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent i = new Intent(activity, WebviewAcitivity.class);
                i.putExtra("Heading","TERMS & CONDITIONS");
                i.putExtra("URL", Config.TERMSANDCONDITIONSURL);
                startActivity(i);
            }
        });


    }

    public void initViews(){


        binding.Head.tvHeaderName.setText("REGISTER & PLAY");
        binding.Head.imBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onBackPressed();
            }
        });

    }
    private void callSignupApi(boolean isShowLoader) {
        try {

            apiRequestManager.callAPI(SIGNUP,
                    createRequestJson(), context, activity, SIGNUPTYPE,
                    isShowLoader,responseManager);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJson() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("mobile", MobileNumber);
            jsonObject.put("email", EmailId);
            jsonObject.put("password", Password);
            jsonObject.put("code", ReferralCode);
            jsonObject.put("type", LoginType);


        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }

    private void callLoginApi(boolean isShowLoader) {
```

```
    try {

          apiRequestManager.callAPI(LOGIN,
                createRequestJsonLogin(), context, activity, LOGINTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
          e.printStackTrace();
    }
  }

  JSONObject createRequestJsonLogin() {
    JSONObject jsonObject = new JSONObject();
    try {
          jsonObject.put("mobile", EmailId);
          jsonObject.put("password", Password);
          jsonObject.put("type", LoginType);
          jsonObject.put("token", FirebaseInstanceId.getInstance().getToken());


    } catch (JSONException e) {
          e.printStackTrace();
    }
    return jsonObject;
  }




  @Override
```

```
    public void getResult(Context mContext, String type,String message, JSONObject result) {
        Validations.showProgress(context);
        if (type.equals(SIGNUPTYPE)) {
            Log.e("RegistrationActivity>>", "getResult: >>>" + result + "\n" + type);

            try {
                String UserId = result.getString("user_id");
                String mobile = result.getString("mobile");
                String email = result.getString("email");
                String LoginType = result.getString("type");

                if (LoginType.equals("Normal")) {
                    Validations.hideProgress();
                    Intent i = new Intent(activity, VerifyOTPActivity.class);
                    i.putExtra("Number", mobile);
                    i.putExtra("Activity", "Registration");
                    i.putExtra("UserId", UserId);
                    i.putExtra("Password", Password);
                    startActivity(i);
                } else {
                    Validations.hideProgress();
                    callLoginApi(true);
                }

            } catch (JSONException e) {
                e.printStackTrace();
            }

        } else if (type.equals(LOGINTYPE)) {
            Validations.hideProgress();
            loginPrefsEditor.putBoolean("saveLogin", true);
            loginPrefsEditor.commit();

            try {
                UserDetails userDetails = new UserDetails();
                userDetails.setUser_id(result.getString("user_id"));
                userDetails.setName(result.getString("name"));
                userDetails.setMobile(result.getString("mobile"));
                userDetails.setEmail(result.getString("email"));
                userDetails.setType(result.getString("type"));
                userDetails.setVerify(result.getString("verify"));
                userDetails.setReferral_code(result.getString("referral_code"));
                sessionManager.setUser(context, userDetails);
                Intent i = new Intent(activity, HomeActivity.class);
                startActivity(i);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            Validations.hideProgress();
        }
    }

    @Override
    public void onError(Context mContext, String type, String message) {

        if (type.equals(SIGNUPTYPE)) {
            revokeAccess();
            ShowToast(context,""+message);
        }
        else if (type.equals(LOGINTYPE)){
            ShowToast(context,"Some Error Occured While Login. Please Try Again");
        }

    }

    public void RequestData() {
        GraphRequest request = 
GraphRequest.newMeRequest(AccessToken.getCurrentAccessToken(),
                new GraphRequest.GraphJSONObjectCallback() {
                    @Override
                    public void onCompleted(JSONObject object, GraphResponse response) {

                        JSONObject json = response.getJSONObject();
                        try {
                            if (json != null) {

                                String FBName = json.getString("name");
                                String ProfileUrl = 
object.getJSONObject("picture").getJSONObject("data").getString("url");
                                String Id = json.getString("id");
                                //  String gender = json.getString("gender");
                                String Fname = json.getString("first_name");
                                String Lname = json.getString("last_name");
                                String FBEmail = json.getString("email");

                                MobileNumber = "";
                                EmailId = FBEmail;
                                Password = Id;

                                callSignupApi(true);
                                Log.e("RegistrationActivity>",json.toString());
                            }
                        } catch (JSONException e) {
                            e.printStackTrace();
                            LoginManager.getInstance().logOut();
                            final AlertDialog.Builder ab = new AlertDialog.Builder(activity);
                            ab.setMessage("Due to your facebook privacy settings," +
                                    "Facebook is denied to provide enough data for " +
                                    "login process.You can use our other Signup process");
                            ab.setPositiveButton("SignUp", new DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface dialog, int id) {
                                    Intent i = new Intent(activity,MainActivity.class);
                                    startActivity(i);
                                }
                            });
                            ab.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface dialog, int id) {
                                    dialog.dismiss();
                                }
                            });
                            ab.setCancelable(false);
                            AlertDialog alert = ab.create();
                            alert.show();


                        }
                    }
                });
```

```
        Bundle parameters = new Bundle();
        parameters.putString("fields", "id,name,link,email,picture,gender,first_name,last_name,birthday");
        request.setParameters(parameters);
        request.executeAsync();
    }

    private void revokeAccess() {

        try {
            Auth.GoogleSignInApi.revokeAccess(mGoogleApiClient).setResultCallback(
                    new ResultCallback<Status>() {
                        @Override
                        public void onResult(Status status) {
                            // updateUI(false);
                        }
                    });
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }



    private void handleSignInResult(GoogleSignInResult result) {
        Log.d(TAG, "handleSignInResult:" + result.isSuccess());
        if (result.isSuccess()) {
            // Signed in successfully, show authenticated UI.
            GoogleSignInAccount acct = result.getSignInAccount();

            Log.e(TAG, "display name: " + acct.getDisplayName());
            String personName = acct.getDisplayName();
            // String personPhotoUrl = acct.getPhotoUrl().toString();
            String email = acct.getEmail();
            String Id = acct.getId();

            MobileNumber = "";
            EmailId = email;
            Password = Id;
            callSignupApi(true);

            Log.e(TAG, "Name: " + personName + ", email: " + email
                    + ", Image: " /personPhotoUrl/);
        } else {
        }
    }



    @Override
    public void onStart() {
        super.onStart();
    }

    @Override
    public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
        // An unresolvable error has occurred and Google APIs (including Sign-In) will not
        // be available.
        Log.d(TAG, "onConnectionFailed:" + connectionResult);
    }
    @Override
    protected void onActivityResult ( int requestCode, int resultCode, Intent data){
        super.onActivityResult(requestCode, resultCode, data);

        // Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);
        if (requestCode == RC_SIGN_IN) {
            GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
            handleSignInResult(result);
        }
        else {
            callbackManager.onActivityResult(requestCode, resultCode, data);
            Validations.showProgress(context);
            final Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    Validations.hideProgress();
                }
            }, 5000);
        }

    }



    private boolean checkPermissions() {
        int result;
        List<String> listPermissionsNeeded = new ArrayList<>();
        for (String p : permissions) {
            result = ContextCompat.checkSelfPermission(this, p);
            if (result != PackageManager.PERMISSION_GRANTED) {
                listPermissionsNeeded.add(p);
            }
        }
        if (!listPermissionsNeeded.isEmpty()) {
            ActivityCompat.requestPermissions(this, listPermissionsNeeded.toArray(new String[listPermissionsNeeded.size()]), 100);
            return false;
        }
        else {
            callSignupApi(true);
        }
        return true;
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
        if (requestCode == 100) {
            if (grantResults.length > 0
                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                callSignupApi(true);
            }
            else {
                callSignupApi(true);
            }
            return;
        }
    }

    private void initFbObject(final Context mContext) {
```

```
    //initilaize facebbok sdk
    FacebookSdk.sdkInitialize(mContext);

FacebookSdk.setApplicationId(mContext.getResources().getString(R.string.facebook_ap
p_id));


    callbackManager = CallbackManager.Factory.create();


    LoginManager.getInstance().logInWithReadPermissions((Activity)
        mContext, Arrays.asList("public_profile", "email"));
    //Facebook
    LoginManager.getInstance().registerCallback(callbackManager, new
FacebookCallback<LoginResult>() {
      @Override
      public void onSuccess(LoginResult loginResult) {
        //showProgressDialog();
        String userId = loginResult.getAccessToken().getUserId();
        accessTokan = loginResult.getAccessToken();


        try {
          GraphRequest request =
GraphRequest.newMeRequest(loginResult.getAccessToken(), new
GraphRequest.GraphJSONObjectCallback() {
            @Override
            public void onCompleted(final JSONObject object, GraphResponse
response) {
              Log.i("LoginActivity", response.toString());
              // Get facebook data from login


              getFacebookData(object);


              LoginManager.getInstance().logOut();
```

```
                    }
                });
                Bundle parameters = new Bundle();
                parameters.putString("fields", "id,name,email,link,picture");
                request.setParameters(parameters);
                request.executeAsync();
            } catch (Exception e) {
                Log.e(getClass().getName(), e.toString());
            }

        }

        @Override
        public void onCancel() {

        }

        @Override
        public void onError(FacebookException error) {
            Log.e("FacebookError", "" + error);
        }
    });
}




/get facebook data/
private String getFacebookData(JSONObject object) {
        try {
        if (object != null) {
            Log.e("fb_response", object.toString());

            String  user_fullname = object.optString("name");
            String  user_fb_id = object.optString("id");
            String user_email = object.optString("email");
            JSONObject jsonObject = object.optJSONObject("picture");
            JSONObject jobj = jsonObject.optJSONObject("data");
            String user_profile_pic = jobj.optString("url");



            MobileNumber = "";
            EmailId = user_email;
            Password = user_fb_id;

            callSignupApi(true);

            Log.e("RegistrationActivity>",object.toString());
            LoginManager.getInstance().logOut();
        }

    } catch (Exception e) {
        e.printStackTrace();

    }
    return "";
}

}
```

LOGIN ACTIVITY :

```
package com.elevendreamer.activity;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.view.WindowManager;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;

import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Class.Validations;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.UserDetails;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityLoginBinding;
import com.facebook.AccessToken;
import com.facebook.CallbackManager;
import com.facebook.FacebookCallback;
import com.facebook.FacebookException;
import com.facebook.FacebookSdk;
import com.facebook.GraphRequest;
import com.facebook.GraphResponse;
import com.facebook.login.LoginManager;
import com.facebook.login.LoginResult;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.common.api.Status;
import com.google.firebase.iid.FirebaseInstanceId;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.Arrays;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.LOGIN;
import static com.elevendreamer.APICallingPackage.Config.SIGNUP;
import static com.elevendreamer.APICallingPackage.Constants.LOGINTYPE;
import static com.elevendreamer.APICallingPackage.Constants.SIGNUPTYPE;


public class LoginActivity extends AppCompatActivity implements ResponseManager,
GoogleApiClient.OnConnectionFailedListener {
```

```
ResponseManager responseManager;
APIRequestManager apiRequestManager;

Context context;
LoginActivity activity;


String EmailorMobile,Password;


CallbackManager callbackManager;
AccessToken accessTokan;

private final String TAG = "RegistrationActivity";
private static final int RC_SIGN_IN = 007;
private GoogleApiClient mGoogleApiClient;


String SEmailId,SPassword,SLoginType;

//Auto Login
private SharedPreferences loginPreferences;
private Boolean saveLogin;
private SharedPreferences.Editor loginPrefsEditor;


String Back = "0";
SessionManager sessionManager;
ActivityLoginBinding binding;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        callbackManager = CallbackManager.Factory.create();
        binding=DataBindingUtil.setContentView(this, R.layout.activity_login);

this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_ST
ATE_ALWAYS_HIDDEN);
        context = activity = this;
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        sessionManager = new SessionManager();

        loginPreferences = getSharedPreferences("loginPrefs", MODE_PRIVATE);
        loginPrefsEditor = loginPreferences.edit();
        saveLogin = loginPreferences.getBoolean("saveLogin", false);
        binding.tvForgotPassword.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                EmailorMobile = binding.etEmailMobile.getText().toString();

                if (EmailorMobile.equals("")){
                    ShowToast(context,"Enter Email or Mobile");
                }else if (EmailorMobile.contains("@")) {
                    if (!Validations.isValidEmail(EmailorMobile)) {
                        binding.etEmailMobile.requestFocus();
                        ShowToast(context, "Enter Valid Email Id");
                    } else {
                        Back = "1";
            }
        } else if (TextUtils.isDigitsOnly(EmailorMobile)) {
            if (!EmailorMobile.matches(Validations.MobilePattern)) {
                binding.etEmailMobile.requestFocus();
                ShowToast(context, "Enter Valid Mobile Number");
            }
            else {
                Back = "1";

            }
        }
        else {
            binding.etEmailMobile.requestFocus();
            ShowToast(context, "Enter Valid Mobile Number or Email");
        }
    }
});



binding.tvLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        EmailorMobile = binding.etEmailMobile.getText().toString();
        Password = binding.etPassword.getText().toString();
        if (EmailorMobile.equals("")){
            ShowToast(context,"Enter Email or Mobile");
        }
        else
        if (Password.equals("")){
            ShowToast(context,"Enter Password");
        }
        else if (Password.length()<8&& !Validations.isValidPassword(Password)){

            ShowToast(context,"Password Pattern Not Macthed");
        }
        else {
            SLoginType = "Normal";
            callLoginApi(true);

        }
    }
});

binding.tvForgotPassword.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        EmailorMobile = binding.etEmailMobile.getText().toString();

        if (EmailorMobile.equals("")){
            ShowToast(context,"Enter Email or Mobile");
        }else if (EmailorMobile.contains("@")) {
            if (!Validations.isValidEmail(EmailorMobile)) {
                binding.etEmailMobile.requestFocus();
                ShowToast(context, "Enter Valid Email Id");
            } else {
                Intent i = new Intent(activity, ForgotVerifyOTPActivity.class);
                i.putExtra("type","Email");
                i.putExtra("EmailorMobile",EmailorMobile);
            startActivity(i);

        }
    } else if (TextUtils.isDigitsOnly(EmailorMobile)) {
        if (!EmailorMobile.matches(Validations.MobilePattern)) {
            binding.etEmailMobile.requestFocus();
            ShowToast(context, "Enter Valid Mobile Number");
        }
        else {
            Intent i = new Intent(activity, ForgotVerifyOTPActivity.class);
            i.putExtra("type","Number");
            i.putExtra("EmailorMobile",EmailorMobile);
            startActivity(i);

        }
    }
    else {
        binding.etEmailMobile.requestFocus();
        ShowToast(context, "Enter Valid Mobile Number or Email");
    }


}
});

binding.tvSignUpText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(activity,RegistrationActivity.class);
        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
```

```
            i.putExtra("Reffered","No");
            startActivity(i);
        }
    });
    binding.RLFBLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            SLoginType = "Email";
            initFbObject(context);
        }
    });

    binding.RLGmailLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            SLoginType = "Email";
            Intent signInIntent =
Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
            startActivityForResult(signInIntent, RC_SIGN_IN);
        }
    });

    binding.fbLoginButton.setReadPermissions("email");
    binding.fbLoginButton.setReadPermissions("public_profile");
    binding.fbLoginButton.setReadPermissions("user_birthday");
    binding.fbLoginButton.setReadPermissions("user_friends");
    //G+ Login
    GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestEmail()
            .build();
    mGoogleApiClient = new GoogleApiClient.Builder(this)
            .enableAutoManage(this, this)
            .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
            .build();

    // Customizing G+ button
    binding.btnSignIn.setSize(SignInButton.SIZE_STANDARD);
    binding.btnSignIn.setScopes(gso.getScopeArray());



    binding.btnSignIn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent signInIntent =
Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
            startActivityForResult(signInIntent, RC_SIGN_IN);
        }
    });

}

private void callLoginApi(boolean isShowLoader) {
    try {

        apiRequestManager.callAPI(LOGIN,
                createRequestJson(), context, activity, LOGINTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
```

```
        e.printStackTrace();
    }
}

JSONObject createRequestJson() {
    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("mobile", EmailorMobile);
        jsonObject.put("password", Password);
        jsonObject.put("type",SLoginType );
        jsonObject.put("token", FirebaseInstanceId.getInstance().getToken());

    } catch (JSONException e) {
        e.printStackTrace();
    }
    return jsonObject;
}
private void callSignupApi(boolean isShowLoader) {
    try {
        apiRequestManager.callAPI(SIGNUP,
                createRequestJsonSignUp(), context, activity, SIGNUPTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

JSONObject createRequestJsonSignUp() {
    JSONObject jsonObject = new JSONObject();
```

```
        try {
            jsonObject.put("mobile", "");
            jsonObject.put("email", SEmailId);
            jsonObject.put("password", SPassword);
            jsonObject.put("code", "");
            jsonObject.put("type", SLoginType);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }
    @Override
    public void getResult(Context mContext, String type,String message, JSONObject
result) {
        Validations.showProgress(context);
        if (type.equals(LOGINTYPE)){
            String Verify = "0";
            String MobNumber = "";
            String UserId = "";

            Log.e("MainActivity>>", "getResult: >>>"+ result+"\n"+type);
            ShowToast(context,message);

            try {
                Verify = result.getString("verify");
                MobNumber = result.getString("mobile");
                UserId = result.getString("user_id");

                if (Verify.equals("0")){
                    Validations.hideProgress();
                    binding.etEmailMobile.setText("");
            binding.etPassword.setText("");
            Intent i = new Intent(activity, VerifyOTPActivity.class);
            i.putExtra("Number", MobNumber);
            i.putExtra("Activity", "Login");
            i.putExtra("UserId", UserId);
            i.putExtra("Password", Password);
            startActivity(i);

        }
        else {
            Validations.hideProgress();
            loginPrefsEditor.putBoolean("saveLogin", true);
            loginPrefsEditor.commit();

            UserDetails userDetails = new UserDetails();
            userDetails.setUser_id(UserId);
            userDetails.setName(result.getString("name"));
            userDetails.setMobile(result.getString("mobile"));
            userDetails.setEmail(result.getString("email"));
            userDetails.setType(result.getString("type"));
            userDetails.setReferral_code(result.getString("referral_code"));

            userDetails.setVerify(Verify);
            sessionManager.setUser(context, userDetails);
            binding.etEmailMobile.setText("");
            binding.etPassword.setText("");
            Intent i = new Intent(activity, HomeActivity.class);
            startActivity(i);
        }
    }
```

```
            catch (Exception e){
                e.printStackTrace();
            }

        }
    }



    @Override
    public void onError(Context mContext, String type, String message) {
        if (type.equals(LOGINTYPE)){
            ShowToast(context,message);
            try {
                LoginManager.getInstance().logOut();
            }
            catch (Exception e){
                e.printStackTrace();
            }
            try {
                revokeAccess();
            }
            catch (Exception e){
                e.printStackTrace();
            }
        }

    }
    public void RequestData() {
        GraphRequest request =
GraphRequest.newMeRequest(AccessToken.getCurrentAccessToken(),
                new GraphRequest.GraphJSONObjectCallback() {
```

```
          @Override
          public void onCompleted(JSONObject object, GraphResponse response) {
             JSONObject json = response.getJSONObject();
             try {
                if (json != null) {

                   String FBName = json.getString("name");
                   String ProfileUrl = object.getJSONObject("picture").
                         getJSONObject("data").getString("url");
                   String Id = json.getString("id");
                   //  String gender = json.getString("gender");
                   String Fname = json.getString("first_name");
                   String Lname = json.getString("last_name");
                   String FBEmail = json.getString("email");

                   EmailorMobile = FBEmail;
                   Password = Id;

                   callLoginApi(true);
                   Log.e("RegistrationActivity>",json.toString());
                }
             } catch (JSONException e) {
                e.printStackTrace();
                LoginManager.getInstance().logOut();
                final AlertDialog.Builder ab = new AlertDialog.Builder(activity);
                ab.setMessage("Due to your facebook privacy settings," +
                      "Facebook is denied to provide enough data for " +
                      "login process.You can use our other Signup process");
                ab.setPositiveButton("SignUp", new DialogInterface.OnClickListener()
{
                   public void onClick(DialogInterface dialog, int id) {
```

```
                                Intent i = new Intent(activity,MainActivity.class);
                                startActivity(i);
                            }
                        });
                        ab.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
                            public void onClick(DialogInterface dialog, int id) {
                                dialog.dismiss();
                            }
                        });
                        ab.setCancelable(false);
                        AlertDialog alert = ab.create();
                        alert.show();



                    }
                }
            });
        Bundle parameters = new Bundle();
        parameters.putString("fields",
"id,name,link,email,picture,gender,first_name,last_name,birthday");
        request.setParameters(parameters);
        request.executeAsync();
    }

    private void revokeAccess() {
        Auth.GoogleSignInApi.revokeAccess(mGoogleApiClient).setResultCallback(
                new ResultCallback<Status>() {
                    @Override
                    public void onResult(Status status) {
                        // updateUI(false);
                }
            });
    }

    private void handleSignInResult(GoogleSignInResult result) {
        Log.d(TAG, "handleSignInResult:" + result.isSuccess());
        if (result.isSuccess()) {
            // Signed in successfully, show authenticated UI.
            GoogleSignInAccount acct = result.getSignInAccount();

            Log.e(TAG, "display name: " + acct.getDisplayName());

            String personName = acct.getDisplayName();
            String email = acct.getEmail();
            String Id = acct.getId();

            EmailorMobile = email;
            Password = Id;
            callLoginApi(true);
            Log.e(TAG, "Name: " + personName + ", email: " + email
                    + ", Image: " /personPhotoUrl/);

        } else {
            // Signed out, show unauthenticated UI.

        }
    }
    @Override
    public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
        // An unresolvable error has occurred and Google APIs (including Sign-In) will not
        // be available.
        Log.d(TAG, "onConnectionFailed:" + connectionResult);
    }

    @Override
    protected void onActivityResult ( int requestCode, int resultCode, Intent data){
        super.onActivityResult(requestCode, resultCode, data);

        // Result returned from launching the Intent from
GoogleSignInApi.getSignInIntent(...);
        if (requestCode == RC_SIGN_IN) {
            GoogleSignInResult result =
Auth.GoogleSignInApi.getSignInResultFromIntent(data);
            handleSignInResult(result);
        }



        callbackManager.onActivityResult(requestCode, resultCode, data);
        Validations.showProgress(context);
        final Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                Validations.hideProgress();
            }
        }, 6000);

    }

    @Override
    public void onBackPressed() {
        if (Back.equals("1")){
            Back = "0";


        }
        else {
            super.onBackPressed();
        }
    }


    private void initFbObject(final Context mContext) {
        //initilaize facebbok sdk
        FacebookSdk.sdkInitialize(mContext);

FacebookSdk.setApplicationId(mContext.getResources().getString(R.string.facebook_ap
p_id));


        callbackManager = CallbackManager.Factory.create();


        LoginManager.getInstance().logInWithReadPermissions((Activity)
                mContext, Arrays.asList("public_profile", "email"));
        //Facebook
        LoginManager.getInstance().registerCallback(callbackManager, new
FacebookCallback<LoginResult>() {
            @Override
            public void onSuccess(LoginResult loginResult) {
                String userId = loginResult.getAccessToken().getUserId();
                accessTokan = loginResult.getAccessToken();


                try {
                    GraphRequest request = GraphRequest.newMeRequest(loginResult
                            .getAccessToken(), new GraphRequest.GraphJSONObjectCallback() {
```

```
                        @Override
                        public void onCompleted(final JSONObject object, GraphResponse
response) {
                            Log.i("LoginActivity", response.toString());
                            // Get facebook data from login

                            getFacebookData(object);

                            LoginManager.getInstance().logOut();

                        }
                    });
                    Bundle parameters = new Bundle();
                    parameters.putString("fields", "id,name,email,link,picture");
                    request.setParameters(parameters);
                    request.executeAsync();
                } catch (Exception e) {
                    Log.e(getClass().getName(), e.toString());
                }

            }

            @Override
            public void onCancel() {

            }

            @Override
            public void onError(FacebookException error) {
                Log.e("FacebookError", "" + error);
            }
```

```
    });
}

/get facebook data/
private String getFacebookData(JSONObject object) {
    try {
        if (object != null) {
            Log.e("fb_response", object.toString());

            String  user_fullname = object.optString("name");
            String  user_fb_id = object.optString("id");
            String user_email = object.optString("email");
            JSONObject jsonObject = object.optJSONObject("picture");
            JSONObject jobj = jsonObject.optJSONObject("data");
            String user_profile_pic = jobj.optString("url");



            EmailorMobile = user_email;
            Password = user_fb_id;

            callLoginApi(true);
            Log.e("RegistrationActivity>",object.toString());
            LoginManager.getInstance().logOut();
        }

    } catch (Exception e) {
        e.printStackTrace();

    }
    return "";
    }


}
```

VERIFY OTP CODE

```
[9:57 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.content.pm.Signature;
import android.os.Bundle;
import android.os.Handler;

import androidx.appcompat.app.AppCompatActivity;

import android.util.Base64;
import android.util.Log;

import com.elevendreamer.R;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import static com.elevendreamer.APICallingPackage.Constants.SPLASH_TIME_OUT;


public class SplashScreen extends AppCompatActivity {

    SplashScreen activity;
    Context context;

    private SharedPreferences loginPreferences;
    private Boolean saveLogin;
    private SharedPreferences.Editor loginPrefsEditor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        context = activity = this;
        printHashKey(context);

        loginPreferences = getSharedPreferences("loginPrefs", MODE_PRIVATE);
        loginPrefsEditor = loginPreferences.edit();
        saveLogin = loginPreferences.getBoolean("saveLogin", false);
        new Handler().postDelayed(new Runnable() {


            @Override
            public void run() {

                if (saveLogin == true) {
                    HomeScreen();
                } else {
                    LoginScreen();
                }
            }
        }, SPLASH_TIME_OUT);

    }

    public void LoginScreen() {
        Intent i = new Intent(SplashScreen.this, MainActivity.class);
        startActivity(i);
    }

    public void HomeScreen() {
        Intent i = new Intent(SplashScreen.this, HomeActivity.class);
        startActivity(i);
    }



    public static void printHashKey(Context pContext) {
        try {
            PackageInfo info =
pContext.getPackageManager().getPackageInfo(pContext.getPackageName(),
                    PackageManager.GET_SIGNATURES);
            for (Signature signature : info.signatures) {
                MessageDigest md = MessageDigest.getInstance("SHA");
                md.update(signature.toByteArray());
                String hashKey = new String(Base64.encode(md.digest(), 0));
                Log.i("TAG", "printHashKey() Hash Key: " + hashKey);
            }
        } catch (NoSuchAlgorithmException e) {
            Log.e("TAG", "printHashKey()", e);
        } catch (Exception e) {
            Log.e("TAG", "printHashKey()", e);
        }
    }
}
[10:10 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.CountDownTimer;

import androidx.databinding.DataBindingUtil;
import androidx.localbroadcastmanager.content.LocalBroadcastManager;
import androidx.appcompat.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.WindowManager;

import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.UserDetails;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityVerifyOtpBinding;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.concurrent.TimeUnit;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.LOGIN;
import static com.elevendreamer.APICallingPackage.Config.RESENDOTP;
import static com.elevendreamer.APICallingPackage.Config.VERIFYOTP;
import static com.elevendreamer.APICallingPackage.Constants.LOGINTYPE;
import static com.elevendreamer.APICallingPackage.Constants.RESENDOTPTYPE;
import static com.elevendreamer.APICallingPackage.Constants.VERIFYOTPTYPE;

public class VerifyOTPActivity extends AppCompatActivity implements
ResponseManager {
    ResponseManager responseManager;
    APIRequestManager apiRequestManager;
    Context context;
    VerifyOTPActivity activity;
    String OTP;
    String IntentNumber,IntentUserId,IntentPassword,IntentActivity;
    private static CountDownTimer countDownTimer;
    //Auto Login
    private SharedPreferences loginPreferences;
    private Boolean saveLogin;
    private SharedPreferences.Editor loginPrefsEditor;
    SessionManager sessionManager;
    ActivityVerifyOtpBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding= DataBindingUtil.setContentView(this, R.layout.activity_verify_otp);


this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_ST
ATE_ALWAYS_HIDDEN);
        context = activity = this;
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        initView();
        sessionManager = new SessionManager();

        loginPreferences = getSharedPreferences("loginPrefs", MODE_PRIVATE);
        loginPrefsEditor = loginPreferences.edit();
        saveLogin = loginPreferences.getBoolean("saveLogin", false);

        Intent o = getIntent();
        IntentNumber = o.getStringExtra("Number");
        IntentUserId = o.getStringExtra("UserId");
        IntentPassword = o.getStringExtra("Password");
        IntentActivity= o.getStringExtra("Activity");

        callResendOTPApi(true);

        if (IntentActivity.equals("Login")){
            callResendOTPApi(true);
        }
        }

        binding.tvOtpSendTo.setText("OTP sent to "+IntentNumber);



        countDownTimer =  new CountDownTimer(60000, 1000) {

            public void onTick(long millisUntilFinished) {
                //tv_Timer.setText("Resend OTP in: " + millisUntilFinished / 1000);
                binding.tvOtpTimer.setText("Didn't receive the OTP? Request for a new one in 
"+ String.format("%d:%d sec",
                        TimeUnit.MILLISECONDS.toMinutes( millisUntilFinished),
                        TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished) -

TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millisUntilFinis
hed))));
            }
            public void onFinish() {
                binding.tvOtpResend.setVisibility(View.VISIBLE);
                binding.tvOtpTimer.setVisibility(View.GONE);
            }

        }.start();

        binding.etOtp1.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp1.getText().toString().length() == 1)     //size as per your 
requirement
                {
                    binding.etOtp2.requestFocus();
                }
            }
            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });

        binding.etOtp2.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp2.getText().toString().length() == 1)     //size as per your
requirement
                {
                    binding.etOtp3.requestFocus();
                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });

        binding.etOtp3.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp3.getText().toString().length() == 1)     //size as per your
requirement
                {
                    binding.etOtp4.requestFocus();
                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });
        binding.etOtp4.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp4.getText().toString().length() == 1)     //size as per your
requirement
                {
                   OTP = GetOTP();
                   callVerifyOTPApi(true);

                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }
```

```
    public void afterTextChanged(Editable s) {
    }

});



binding.tvVerifyOTP.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        OTP  = GetOTP();

        callVerifyOTPApi(true);
        countDownTimer.cancel();

    }
});

binding.tvOtpResend.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        callResendOTPApi(true);

        binding.tvOtpTimer.setVisibility(View.VISIBLE);
        binding.tvOtpResend.setVisibility(View.GONE);
        countDownTimer =  new CountDownTimer(60000, 1000) {

            public void onTick(long millisUntilFinished) {
                //tv_Timer.setText("Resend OTP in: " + millisUntilFinished / 1000);
```

```
                    binding.tvOtpTimer.setText("Didn't receive the OTP? Request for a new
one in "+ String.format("%d:%d sec",
                            TimeUnit.MILLISECONDS.toMinutes( millisUntilFinished),
                            TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished) -

TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millisUntilFinis
hed))));
                }
                public void onFinish() {
                    binding.tvOtpResend.setVisibility(View.VISIBLE);
                    binding.tvOtpTimer.setVisibility(View.GONE);
                }

            }.start();

        }
    });

}

public void initView(){


    binding.head.imBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            try {
                countDownTimer.cancel();
            }
            catch (Exception e){
                e.printStackTrace();
            }
            onBackPressed();
        }
    });
    binding.head.tvHeaderName.setText("VERIFY OTP");



}

public String GetOTP(){
    String GETOTP = "";
    String Otp1 = binding.etOtp1.getText().toString();
    String Otp2 = binding.etOtp2.getText().toString();
    String Otp3 = binding.etOtp3.getText().toString();
    String Otp4 = binding.etOtp4.getText().toString();

    if (Otp1.equals("")){
        ShowToast(context,"Enter OTP");
    }
    else if (Otp2.equals("")){
        ShowToast(context,"Enter OTP");
    }else if (Otp3.equals("")){
        ShowToast(context,"Enter OTP");
    }else if (Otp4.equals("")){
        ShowToast(context,"Enter OTP");
    }
    else {
        GETOTP = Otp1+Otp2+Otp3+Otp4;
    }
    return GETOTP;
}



private BroadcastReceiver receiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equalsIgnoreCase("otp")) {
            final String message = intent.getStringExtra("message");

            char o1 = message.charAt(0);
            char o2 = message.charAt(1);
            char o3 = message.charAt(2);
            char o4 = message.charAt(3);



            binding.etOtp1.setText(o1+"");
            binding.etOtp2.setText(o2+"");
            binding.etOtp3.setText(o3+"");
            binding.etOtp4.setText(o4+"");

            GetOTP();
            callVerifyOTPApi(true);
            countDownTimer.cancel();

        }
    }
};

private void callVerifyOTPApi(boolean isShowLoader) {
    try {

        apiRequestManager.callAPI(VERIFYOTP,
                createRequestJson(), context, activity, VERIFYOTPTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

JSONObject createRequestJson() {
    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("mobile", IntentNumber);
        jsonObject.put("otp", OTP);
        jsonObject.put("user_id", IntentUserId);



    } catch (JSONException e) {
        e.printStackTrace();
    }
    return jsonObject;
}


private void callResendOTPApi(boolean isShowLoader) {
    try {
        JSONObject jsonObject = new JSONObject();
        jsonObject.put("user_id",IntentUserId);
        apiRequestManager.callAPI(RESENDOTP,
                jsonObject, context, activity, RESENDOTPTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

private void callLoginApi(boolean isShowLoader) {
    try {

        apiRequestManager.callAPI(LOGIN,
                createRequestJsonLogin(), context, activity, LOGINTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

JSONObject createRequestJsonLogin() {
    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("mobile", IntentNumber);
        jsonObject.put("password", IntentPassword);
        jsonObject.put("type", "Normal");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }




    @Override
    public void onResume() {
        try {
            LocalBroadcastManager.getInstance(this).registerReceiver(receiver, new
IntentFilter("otp"));
        }
        catch (Exception e){
            e.printStackTrace();
        }
        super.onResume();
    }

    @Override
    public void onPause() {
        super.onPause();
        try {
            LocalBroadcastManager.getInstance(this).unregisterReceiver(receiver);
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }
    @Override
    public void getResult(Context mContext, String type, String message, JSONObject
result) {

        ShowToast(context,message);
        if (type.equals(VERIFYOTPTYPE)) {
            callLoginApi(true);
        }
        else if (type.equals(LOGINTYPE)){

            loginPrefsEditor.putBoolean("saveLogin", true);
            loginPrefsEditor.commit();



            try {
                UserDetails userDetails = new UserDetails();
                userDetails.setUser_id(result.getString("user_id"));
                userDetails.setName(result.getString("name"));
                userDetails.setMobile(result.getString("mobile"));
                userDetails.setEmail(result.getString("email"));
                userDetails.setType(result.getString("type"));
                userDetails.setVerify(result.getString("verify"));
                userDetails.setReferral_code(result.getString("referral_code"));
                userDetails.setImage(result.getString("image"));

                sessionManager.setUser(context,userDetails);
            }
            catch (Exception e){
                e.printStackTrace();
            }



            Intent i = new Intent(activity, HomeActivity.class);
            startActivity(i);
        }
        else if (type.equals(RESENDOTPTYPE)){
        }
    }




    @Override
    public void onError(Context mContext, String type, String message) {
        if (type.equals(VERIFYOTP)){
            ShowToast(context,"Invalid OTP");
        }
        else if (type.equals(LOGINTYPE)) {
            ShowToast(context,"Number Verified Successfully. Please Login to Continue");
            Intent i = new Intent(activity, LoginActivity.class);
            startActivity(i);
        }
        else if (type.equals(RESENDOTPTYPE)){
            ShowToast(context,message);
        }

    }

    @Override
    public void onBackPressed() {
        Intent i = new Intent(activity, MainActivity.class);
        startActivity(i);
        finish();
    }
}
```

CREATE PASSWORD CODE

```
[10:10 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.CountDownTimer;

import androidx.databinding.DataBindingUtil;
import androidx.localbroadcastmanager.content.LocalBroadcastManager;
import androidx.appcompat.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.WindowManager;

import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.UserDetails;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityVerifyOtpBinding;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.concurrent.TimeUnit;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.LOGIN;
import static com.elevendreamer.APICallingPackage.Config.RESENDOTP;
import static com.elevendreamer.APICallingPackage.Config.VERIFYOTP;
import static com.elevendreamer.APICallingPackage.Constants.LOGINTYPE;
import static com.elevendreamer.APICallingPackage.Constants.RESENDOTPTYPE;
import static com.elevendreamer.APICallingPackage.Constants.VERIFYOTPTYPE;

public class VerifyOTPActivity extends AppCompatActivity implements
ResponseManager {
    ResponseManager responseManager;
    APIRequestManager apiRequestManager;
    Context context;
    VerifyOTPActivity activity;
    String OTP;
    String IntentNumber,IntentUserId,IntentPassword,IntentActivity;
    private static CountDownTimer countDownTimer;
    //Auto Login
    private SharedPreferences loginPreferences;
    private Boolean saveLogin;
    private SharedPreferences.Editor loginPrefsEditor;
    SessionManager sessionManager;
    ActivityVerifyOtpBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding= DataBindingUtil.setContentView(this, R.layout.activity_verify_otp);


this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);
        context = activity = this;
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        initView();
        sessionManager = new SessionManager();

        loginPreferences = getSharedPreferences("loginPrefs", MODE_PRIVATE);
        loginPrefsEditor = loginPreferences.edit();
        saveLogin = loginPreferences.getBoolean("saveLogin", false);

        Intent o = getIntent();
        IntentNumber = o.getStringExtra("Number");
        IntentUserId = o.getStringExtra("UserId");
        IntentPassword = o.getStringExtra("Password");
        IntentActivity= o.getStringExtra("Activity");

        callResendOTPApi(true);

        if (IntentActivity.equals("Login")){
            callResendOTPApi(true);
        }
        binding.tvOtpSendTo.setText("OTP sent to "+IntentNumber);



        countDownTimer =  new CountDownTimer(60000, 1000) {


            public void onTick(long millisUntilFinished) {
                //tv_Timer.setText("Resend OTP in: " + millisUntilFinished / 1000);
                binding.tvOtpTimer.setText("Didn't receive the OTP? Request for a new one in
"+ String.format("%d:%d sec",
                        TimeUnit.MILLISECONDS.toMinutes( millisUntilFinished),
                        TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished) -

TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millisUntilFinis
hed))));
            }
            public void onFinish() {
                binding.tvOtpResend.setVisibility(View.VISIBLE);
                binding.tvOtpTimer.setVisibility(View.GONE);
            }

        }.start();

        binding.etOtp1.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp1.getText().toString().length() == 1)     //size as per your
requirement
                {
                    binding.etOtp2.requestFocus();
                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });

        binding.etOtp2.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp2.getText().toString().length() == 1)     //size as per your
requirement
                {
                    binding.etOtp3.requestFocus();
                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });
        binding.etOtp3.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp3.getText().toString().length() == 1)     //size as per your
requirement
                {
                    binding.etOtp4.requestFocus();
                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });
        binding.etOtp4.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp4.getText().toString().length() == 1)     //size as per your
requirement
                {
                    OTP = GetOTP();
                    callVerifyOTPApi(true);

                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });



        binding.tvVerifyOTP.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                OTP  = GetOTP();

                callVerifyOTPApi(true);
                countDownTimer.cancel();

            }
        });

        binding.tvOtpResend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                callResendOTPApi(true);

                binding.tvOtpTimer.setVisibility(View.VISIBLE);
                binding.tvOtpResend.setVisibility(View.GONE);
                countDownTimer =  new CountDownTimer(60000, 1000) {
```

```
                    public void onTick(long millisUntilFinished) {
                        //tv_Timer.setText("Resend OTP in: " + millisUntilFinished / 1000);
                        binding.tvOtpTimer.setText("Didn't receive the OTP? Request for a new
one in "+ String.format("%d:%d sec",
                                TimeUnit.MILLISECONDS.toMinutes( millisUntilFinished),
                                TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished) -

TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millisUntilFinis
hed))));
                    }
                    public void onFinish() {
                        binding.tvOtpResend.setVisibility(View.VISIBLE);
                        binding.tvOtpTimer.setVisibility(View.GONE);
                    }

                }.start();

            }
        });

    }

    public void initView(){



        binding.head.imBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    countDownTimer.cancel();
            }
            catch (Exception e){
                e.printStackTrace();
            }
            onBackPressed();
        }
    });
    binding.head.tvHeaderName.setText("VERIFY OTP");



}

public String GetOTP(){
    String GETOTP = "";
    String Otp1 = binding.etOtp1.getText().toString();
    String Otp2 = binding.etOtp2.getText().toString();
    String Otp3 = binding.etOtp3.getText().toString();
    String Otp4 = binding.etOtp4.getText().toString();

    if (Otp1.equals("")){
        ShowToast(context,"Enter OTP");
    }
    else if (Otp2.equals("")){
        ShowToast(context,"Enter OTP");
    }else if (Otp3.equals("")){
        ShowToast(context,"Enter OTP");
    }else if (Otp4.equals("")){
        ShowToast(context,"Enter OTP");
    }
    else {
        GETOTP = Otp1+Otp2+Otp3+Otp4;
    }

    return GETOTP;
}



private BroadcastReceiver receiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equalsIgnoreCase("otp")) {
            final String message = intent.getStringExtra("message");

            char o1 = message.charAt(0);
            char o2 = message.charAt(1);
            char o3 = message.charAt(2);
            char o4 = message.charAt(3);



            binding.etOtp1.setText(o1+"");
            binding.etOtp2.setText(o2+"");
            binding.etOtp3.setText(o3+"");
            binding.etOtp4.setText(o4+"");

            GetOTP();
            callVerifyOTPApi(true);
            countDownTimer.cancel();

        }
    }
```

```
};

private void callVerifyOTPApi(boolean isShowLoader) {
   try {

       apiRequestManager.callAPI(VERIFYOTP,
               createRequestJson(), context, activity, VERIFYOTPTYPE,
               isShowLoader,responseManager);

   } catch (JSONException e) {
       e.printStackTrace();
   }
}

JSONObject createRequestJson() {
   JSONObject jsonObject = new JSONObject();
   try {
       jsonObject.put("mobile", IntentNumber);
       jsonObject.put("otp", OTP);
       jsonObject.put("user_id", IntentUserId);



   } catch (JSONException e) {
       e.printStackTrace();
   }
   return jsonObject;
}
```

```
private void callResendOTPApi(boolean isShowLoader) {
    try {
        JSONObject jsonObject = new JSONObject();
        jsonObject.put("user_id",IntentUserId);
        apiRequestManager.callAPI(RESENDOTP,
                jsonObject, context, activity, RESENDOTPTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

private void callLoginApi(boolean isShowLoader) {
    try {


        apiRequestManager.callAPI(LOGIN,
                createRequestJsonLogin(), context, activity, LOGINTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

JSONObject createRequestJsonLogin() {
    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("mobile", IntentNumber);
        jsonObject.put("password", IntentPassword);
```

```
            jsonObject.put("type", "Normal");



        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }




    @Override
    public void onResume() {
        try {
            LocalBroadcastManager.getInstance(this).registerReceiver(receiver, new
IntentFilter("otp"));
        }
        catch (Exception e){
            e.printStackTrace();
        }
        super.onResume();
    }

    @Override
    public void onPause() {
        super.onPause();
        try {
            LocalBroadcastManager.getInstance(this).unregisterReceiver(receiver);
        }
        catch (Exception e){
            e.printStackTrace();
        }

    }




    @Override

    public void getResult(Context mContext, String type, String message, JSONObject
result) {


        ShowToast(context,message);

        if (type.equals(VERIFYOTPTYPE)) {

            callLoginApi(true);

        }

        else if (type.equals(LOGINTYPE)){


            loginPrefsEditor.putBoolean("saveLogin", true);

            loginPrefsEditor.commit();



            try {

                UserDetails userDetails = new UserDetails();

                userDetails.setUser_id(result.getString("user_id"));

                userDetails.setName(result.getString("name"));

                userDetails.setMobile(result.getString("mobile"));

                userDetails.setEmail(result.getString("email"));

                userDetails.setType(result.getString("type"));

                userDetails.setVerify(result.getString("verify"));

                userDetails.setReferral_code(result.getString("referral_code"));

                userDetails.setImage(result.getString("image"));


                sessionManager.setUser(context,userDetails);
            }
            catch (Exception e){
                e.printStackTrace();
            }



            Intent i = new Intent(activity, HomeActivity.class);
            startActivity(i);
        }
        else if (type.equals(RESENDOTPTYPE)){
        }
    }




    @Override
    public void onError(Context mContext, String type, String message) {
        if (type.equals(VERIFYOTP)){
            ShowToast(context,"Invalid OTP");
        }
        else if (type.equals(LOGINTYPE)) {
            ShowToast(context,"Number Verified Successfully. Please Login to Continue");
            Intent i = new Intent(activity, LoginActivity.class);
            startActivity(i);
        }
        else if (type.equals(RESENDOTPTYPE)){
            ShowToast(context,message);
        }

    }
```

```
    @Override
    public void onBackPressed() {
        Intent i = new Intent(activity, MainActivity.class);
        startActivity(i);
        finish();
    }
}
[10:13 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import com.elevendreamer.databinding.ActivityNewPasswordBinding;

import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;

import android.view.View;

import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Class.Validations;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.R;

import org.json.JSONException;
import org.json.JSONObject;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.CHANGEPASSWORD;
import static
com.elevendreamer.APICallingPackage.Config.UPDATENEWPASSWORD;
import static
com.elevendreamer.APICallingPackage.Constants.CHANGEPASSWORDTPYE;
import static
com.elevendreamer.APICallingPackage.Constants.UPDATENEWPASSWORDTPYE;


public class NewPasswordActivity extends AppCompatActivity implements
ResponseManager {


    ResponseManager responseManager;
    APIRequestManager apiRequestManager;


    Context context;
    NewPasswordActivity activity;




    String UserId,IntentActivity;
    String NewPassword,ConfirmNewPassword,OldPassword;


    ActivityNewPasswordBinding binding;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding= DataBindingUtil.setContentView(this,R.layout.activity_new_password);
        context = activity = this;
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        initViews();
```

```
}

public void initViews(){

    binding.head.tvHeaderName.setText(getString(R.string.new_pass));
    binding.head.imBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            onBackPressed();
        }
    });



    UserId = getIntent().getStringExtra("UserId");
    IntentActivity = getIntent().getStringExtra("IntentActivity");

    if (IntentActivity.equals("ForgotPassword")){
        binding.inputOldPassword.setVisibility(View.GONE);
    }
    else if (IntentActivity.equals("ChangePassword")){
        binding.inputOldPassword.setVisibility(View.VISIBLE);
    }



    binding.tvSubmitNewPassword.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            NewPassword = binding.etNewPassword.getText().toString();
            ConfirmNewPassword = binding.etConfirmNewPassword.getText().toString();

            if (IntentActivity.equals("ChangePassword")){
                OldPassword = binding.etOldPassword.getText().toString();
                if (OldPassword.equals("")){
                    ShowToast(context,"Enter Old Password");
                }else if (OldPassword.length()<8&&
!Validations.isValidPassword(OldPassword)){

                    ShowToast(context,"Password Pattern Not Macthed");
                }

                else if (NewPassword.equals("")){
                    ShowToast(context,"Enter New Password");
                }
                else if (NewPassword.length()<8&&
!Validations.isValidPassword(NewPassword)){

                    ShowToast(context,"Password Pattern Not Macthed");
                }
                else if (ConfirmNewPassword.equals("")){
                    ShowToast(context,"Enter Confirm New Password");
                }
                else if (!NewPassword.equals(ConfirmNewPassword)){
                    ShowToast(context,"Confirm Password Not Match");
                }
                else {
                    callChangePasswordApi(true);
                }
            }
```

```
                else {


                    if (NewPassword.equals("")) {
                        ShowToast(context, "Enter New Password");
                    }else if (NewPassword.length()<8&&
!Validations.isValidPassword(NewPassword)){


                        ShowToast(context,"Password Pattern Not Macthed");
                    }
                    else if (ConfirmNewPassword.equals("")) {
                        ShowToast(context, "Enter Confirm New Password");
                    } else if (!NewPassword.equals(ConfirmNewPassword)) {
                        ShowToast(context, "Confirm Password Not Match");
                    } else {
                        callUpdatePasswordApi(true);
                    }
                }


            }
        });


    }


    private void callChangePasswordApi(boolean isShowLoader) {
        try {


            apiRequestManager.callAPI(CHANGEPASSWORD,
                    createRequestJsonForgotPassword(), context, activity,
CHANGEPASSWORDTPYE,
                    isShowLoader,responseManager);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    private void callUpdatePasswordApi(boolean isShowLoader) {
        try {

            apiRequestManager.callAPI(UPDATENEWPASSWORD,
                    createRequestJsonForgotPassword(), context, activity,
UPDATENEWPASSWORDTPYE,
                    isShowLoader,responseManager);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJsonForgotPassword() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("password", NewPassword);
            jsonObject.put("user_id", UserId);
            if (IntentActivity.equals("ChangePassword")) {
                jsonObject.put("old_password", OldPassword);
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }
    @Override
    public void getResult(Context mContext, String type, String message, JSONObject
result) {
        if (IntentActivity.equals("ChangePassword")){
            ShowToast(context,message);
            Intent i = new Intent(activity, HomeActivity.class);
            startActivity(i);
            finish();
        }
        else {
            ShowToast(context,message);
            Intent i = new Intent(activity, LoginActivity.class);
            startActivity(i);
        }

    }




    @Override
    public void onError(Context mContext, String type, String message) {
        ShowToast(context,message);
    }

    @Override
    public void onBackPressed() {

        if (IntentActivity.equals("ForgotPassword")){
            Intent i = new Intent(activity,LoginActivity.class);
            startActivity(i);
        }
        else {
            super.onBackPressed();
        }
    }
}
```

MY TEAM :

```
[10:10 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.CountDownTimer;

import androidx.databinding.DataBindingUtil;
import androidx.localbroadcastmanager.content.LocalBroadcastManager;
import androidx.appcompat.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.WindowManager;

import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.UserDetails;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityVerifyOtpBinding;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.concurrent.TimeUnit;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.LOGIN;
import static com.elevendreamer.APICallingPackage.Config.RESENDOTP;
import static com.elevendreamer.APICallingPackage.Config.VERIFYOTP;
import static com.elevendreamer.APICallingPackage.Constants.LOGINTYPE;
import static com.elevendreamer.APICallingPackage.Constants.RESENDOTPTYPE;
import static com.elevendreamer.APICallingPackage.Constants.VERIFYOTPTYPE;

public class VerifyOTPActivity extends AppCompatActivity implements
ResponseManager {
    ResponseManager responseManager;
    APIRequestManager apiRequestManager;
    Context context;
    VerifyOTPActivity activity;
    String OTP;
    String IntentNumber,IntentUserId,IntentPassword,IntentActivity;
    private static CountDownTimer countDownTimer;
    //Auto Login
    private SharedPreferences loginPreferences;
    private Boolean saveLogin;
    private SharedPreferences.Editor loginPrefsEditor;
    SessionManager sessionManager;
    ActivityVerifyOtpBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding= DataBindingUtil.setContentView(this, R.layout.activity_verify_otp);


this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);
        context = activity = this;
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        initView();
        sessionManager = new SessionManager();

        loginPreferences = getSharedPreferences("loginPrefs", MODE_PRIVATE);
        loginPrefsEditor = loginPreferences.edit();
        saveLogin = loginPreferences.getBoolean("saveLogin", false);

        Intent o = getIntent();
        IntentNumber = o.getStringExtra("Number");
        IntentUserId = o.getStringExtra("UserId");
        IntentPassword = o.getStringExtra("Password");
        IntentActivity= o.getStringExtra("Activity");

        callResendOTPApi(true);

        if (IntentActivity.equals("Login")){
            callResendOTPApi(true);
        }
        binding.tvOtpSendTo.setText("OTP sent to "+IntentNumber);



        countDownTimer =  new CountDownTimer(60000, 1000) {

            public void onTick(long millisUntilFinished) {
                //tv_Timer.setText("Resend OTP in: " + millisUntilFinished / 1000);
                binding.tvOtpTimer.setText("Didn't receive the OTP? Request for a new one in 
"+ String.format("%d:%d sec",
                    TimeUnit.MILLISECONDS.toMinutes( millisUntilFinished),
                    TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished) -

TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millisUntilFinis
hed))));
            }
            public void onFinish() {
                binding.tvOtpResend.setVisibility(View.VISIBLE);
                binding.tvOtpTimer.setVisibility(View.GONE);
            }

        }.start();

        binding.etOtp1.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp1.getText().toString().length() == 1)     //size as per your 
requirement
                {
                    binding.etOtp2.requestFocus();
                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });

        binding.etOtp2.addTextChangedListener(new TextWatcher() {

            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp2.getText().toString().length() == 1)     //size as per your
requirement
                {
                    binding.etOtp3.requestFocus();
                }
            }

            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });
        binding.etOtp3.addTextChangedListener(new TextWatcher() {


            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp3.getText().toString().length() == 1)     //size as per your
requirement
                {
                    binding.etOtp4.requestFocus();
                }
            }


            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }


            public void afterTextChanged(Editable s) {
            }


        });
        binding.etOtp4.addTextChangedListener(new TextWatcher() {


            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (binding.etOtp4.getText().toString().length() == 1)     //size as per your
requirement
                {
                   OTP = GetOTP();
                   callVerifyOTPApi(true);


                }
            }


            public void beforeTextChanged(CharSequence s, int start,
                                          int count, int after) {
            }

            public void afterTextChanged(Editable s) {
            }

        });



        binding.tvVerifyOTP.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                OTP  = GetOTP();

                callVerifyOTPApi(true);
                countDownTimer.cancel();

            }
        });

        binding.tvOtpResend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                callResendOTPApi(true);

                binding.tvOtpTimer.setVisibility(View.VISIBLE);
                binding.tvOtpResend.setVisibility(View.GONE);
                countDownTimer =  new CountDownTimer(60000, 1000) {
                       public void onTick(long millisUntilFinished) {
                          //tv_Timer.setText("Resend OTP in: " + millisUntilFinished / 1000);
                          binding.tvOtpTimer.setText("Didn't receive the OTP? Request for a new
        one in "+ String.format("%d:%d sec",
                                TimeUnit.MILLISECONDS.toMinutes( millisUntilFinished),
                                TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished) -

        TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millisUntilFinis
        hed))));
                       }
                       public void onFinish() {
                          binding.tvOtpResend.setVisibility(View.VISIBLE);
                          binding.tvOtpTimer.setVisibility(View.GONE);
                       }

                    }.start();

                 }
              });

           }

           public void initView(){



              binding.head.imBack.setOnClickListener(new View.OnClickListener() {
                 @Override
                 public void onClick(View view) {
                    try {
                       countDownTimer.cancel();
                    }
                    catch (Exception e){
                        e.printStackTrace();
                    }
                    onBackPressed();
                }
            });
            binding.head.tvHeaderName.setText("VERIFY OTP");



        }

        public String GetOTP(){
            String GETOTP = "";
            String Otp1 = binding.etOtp1.getText().toString();
            String Otp2 = binding.etOtp2.getText().toString();
            String Otp3 = binding.etOtp3.getText().toString();
            String Otp4 = binding.etOtp4.getText().toString();

            if (Otp1.equals("")){
                ShowToast(context,"Enter OTP");
            }
            else if (Otp2.equals("")){
                ShowToast(context,"Enter OTP");
            }else if (Otp3.equals("")){
                ShowToast(context,"Enter OTP");
            }else if (Otp4.equals("")){
                ShowToast(context,"Enter OTP");
            }
            else {
                GETOTP = Otp1+Otp2+Otp3+Otp4;
            }

            return GETOTP;
        }



        private BroadcastReceiver receiver = new BroadcastReceiver() {
            @Override
            public void onReceive(Context context, Intent intent) {
                if (intent.getAction().equalsIgnoreCase("otp")) {
                    final String message = intent.getStringExtra("message");

                    char o1 = message.charAt(0);
                    char o2 = message.charAt(1);
                    char o3 = message.charAt(2);
                    char o4 = message.charAt(3);



                    binding.etOtp1.setText(o1+"");
                    binding.etOtp2.setText(o2+"");
                    binding.etOtp3.setText(o3+"");
                    binding.etOtp4.setText(o4+"");

                    GetOTP();
                    callVerifyOTPApi(true);
                    countDownTimer.cancel();

                }
            }
```

```
};

private void callVerifyOTPApi(boolean isShowLoader) {
   try {


      apiRequestManager.callAPI(VERIFYOTP,
            createRequestJson(), context, activity, VERIFYOTPTYPE,
            isShowLoader,responseManager);


   } catch (JSONException e) {
      e.printStackTrace();
   }
}

JSONObject createRequestJson() {
   JSONObject jsonObject = new JSONObject();
   try {
      jsonObject.put("mobile", IntentNumber);
      jsonObject.put("otp", OTP);
      jsonObject.put("user_id", IntentUserId);




   } catch (JSONException e) {
      e.printStackTrace();
   }
   return jsonObject;
}
```

```
private void callResendOTPApi(boolean isShowLoader) {
    try {
        JSONObject jsonObject = new JSONObject();
        jsonObject.put("user_id",IntentUserId);
        apiRequestManager.callAPI(RESENDOTP,
                jsonObject, context, activity, RESENDOTPTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

private void callLoginApi(boolean isShowLoader) {
    try {


        apiRequestManager.callAPI(LOGIN,
                createRequestJsonLogin(), context, activity, LOGINTYPE,
                isShowLoader,responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

JSONObject createRequestJsonLogin() {
    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("mobile", IntentNumber);
        jsonObject.put("password", IntentPassword);
```

```
            jsonObject.put("type", "Normal");



        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }




    @Override
    public void onResume() {
        try {
            LocalBroadcastManager.getInstance(this).registerReceiver(receiver, new
IntentFilter("otp"));
        }
        catch (Exception e){
            e.printStackTrace();
        }
        super.onResume();
    }

    @Override
    public void onPause() {
        super.onPause();
        try {
            LocalBroadcastManager.getInstance(this).unregisterReceiver(receiver);
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }



    @Override
    public void getResult(Context mContext, String type, String message, JSONObject result) {

        ShowToast(context,message);
        if (type.equals(VERIFYOTPTYPE)) {
            callLoginApi(true);
        }
        else if (type.equals(LOGINTYPE)){

            loginPrefsEditor.putBoolean("saveLogin", true);
            loginPrefsEditor.commit();


            try {
                UserDetails userDetails = new UserDetails();
                userDetails.setUser_id(result.getString("user_id"));
                userDetails.setName(result.getString("name"));
                userDetails.setMobile(result.getString("mobile"));
                userDetails.setEmail(result.getString("email"));
                userDetails.setType(result.getString("type"));
                userDetails.setVerify(result.getString("verify"));
                userDetails.setReferral_code(result.getString("referral_code"));
                userDetails.setImage(result.getString("image"));

                sessionManager.setUser(context,userDetails);
            }
            catch (Exception e){
                e.printStackTrace();
            }



            Intent i = new Intent(activity, HomeActivity.class);
            startActivity(i);
        }
        else if (type.equals(RESENDOTPTYPE)){
        }
    }




    @Override
    public void onError(Context mContext, String type, String message) {
        if (type.equals(VERIFYOTP)){
            ShowToast(context,"Invalid OTP");
        }
        else if (type.equals(LOGINTYPE)) {
            ShowToast(context,"Number Verified Successfully. Please Login to Continue");
            Intent i = new Intent(activity, LoginActivity.class);
            startActivity(i);
        }
        else if (type.equals(RESENDOTPTYPE)){
            ShowToast(context,message);
        }

    }
```

```
    @Override
    public void onBackPressed() {
        Intent i = new Intent(activity, MainActivity.class);
        startActivity(i);
        finish();
    }
}
[10:13 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import com.elevendreamer.databinding.ActivityNewPasswordBinding;

import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;

import android.view.View;

import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Class.Validations;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.R;

import org.json.JSONException;
import org.json.JSONObject;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.CHANGEPASSWORD;
import static
com.elevendreamer.APICallingPackage.Config.UPDATENEWPASSWORD;
import static
com.elevendreamer.APICallingPackage.Constants.CHANGEPASSWORDTPYE;
import static
com.elevendreamer.APICallingPackage.Constants.UPDATENEWPASSWORDTPYE;


public class NewPasswordActivity extends AppCompatActivity implements
ResponseManager {


    ResponseManager responseManager;
    APIRequestManager apiRequestManager;


    Context context;
    NewPasswordActivity activity;




    String UserId,IntentActivity;
    String NewPassword,ConfirmNewPassword,OldPassword;


    ActivityNewPasswordBinding binding;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding= DataBindingUtil.setContentView(this,R.layout.activity_new_password);
        context = activity = this;
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        initViews();
```

```
}

public void initViews(){

    binding.head.tvHeaderName.setText(getString(R.string.new_pass));
    binding.head.imBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            onBackPressed();
        }
    });



    UserId = getIntent().getStringExtra("UserId");
    IntentActivity = getIntent().getStringExtra("IntentActivity");

    if (IntentActivity.equals("ForgotPassword")){
        binding.inputOldPassword.setVisibility(View.GONE);
    }
    else if (IntentActivity.equals("ChangePassword")){
        binding.inputOldPassword.setVisibility(View.VISIBLE);
    }



    binding.tvSubmitNewPassword.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            NewPassword = binding.etNewPassword.getText().toString();
            ConfirmNewPassword = binding.etConfirmNewPassword.getText().toString();

            if (IntentActivity.equals("ChangePassword")){
                OldPassword = binding.etOldPassword.getText().toString();
                if (OldPassword.equals("")){
                    ShowToast(context,"Enter Old Password");
                }else if (OldPassword.length()<8&&
!Validations.isValidPassword(OldPassword)){

                    ShowToast(context,"Password Pattern Not Macthed");
                }

                else if (NewPassword.equals("")){
                    ShowToast(context,"Enter New Password");
                }
                else if (NewPassword.length()<8&&
!Validations.isValidPassword(NewPassword)){

                    ShowToast(context,"Password Pattern Not Macthed");
                }
                else if (ConfirmNewPassword.equals("")){
                    ShowToast(context,"Enter Confirm New Password");
                }
                else if (!NewPassword.equals(ConfirmNewPassword)){
                    ShowToast(context,"Confirm Password Not Match");
                }
                else {
                    callChangePasswordApi(true);
                }
            }
```

```
                else {


                    if (NewPassword.equals("")) {
                        ShowToast(context, "Enter New Password");
                    }else if (NewPassword.length()<8&&
!Validations.isValidPassword(NewPassword)){


                        ShowToast(context,"Password Pattern Not Macthed");
                    }
                    else if (ConfirmNewPassword.equals("")) {
                        ShowToast(context, "Enter Confirm New Password");
                    } else if (!NewPassword.equals(ConfirmNewPassword)) {
                        ShowToast(context, "Confirm Password Not Match");
                    } else {
                        callUpdatePasswordApi(true);
                    }
                }


            }
        });


    }


    private void callChangePasswordApi(boolean isShowLoader) {
        try {


            apiRequestManager.callAPI(CHANGEPASSWORD,
                    createRequestJsonForgotPassword(), context, activity,
CHANGEPASSWORDTPYE,
                    isShowLoader,responseManager);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    private void callUpdatePasswordApi(boolean isShowLoader) {
        try {

            apiRequestManager.callAPI(UPDATENEWPASSWORD,
                    createRequestJsonForgotPassword(), context, activity,
UPDATENEWPASSWORDTPYE,
                    isShowLoader,responseManager);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJsonForgotPassword() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("password", NewPassword);
            jsonObject.put("user_id", UserId);
            if (IntentActivity.equals("ChangePassword")) {
                jsonObject.put("old_password", OldPassword);
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }
    @Override
    public void getResult(Context mContext, String type, String message, JSONObject
result) {
        if (IntentActivity.equals("ChangePassword")){
            ShowToast(context,message);
            Intent i = new Intent(activity, HomeActivity.class);
            startActivity(i);
            finish();
        }
        else {
            ShowToast(context,message);
            Intent i = new Intent(activity, LoginActivity.class);
            startActivity(i);
        }

    }



    @Override
    public void onError(Context mContext, String type, String message) {
        ShowToast(context,message);
    }

    @Override
    public void onBackPressed() {

        if (IntentActivity.equals("ForgotPassword")){
            Intent i = new Intent(activity,LoginActivity.class);
            startActivity(i);
        }
        else {
            super.onBackPressed();
        }
    }
}
```

PAYMENTS CODE:

```
package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.BeanAddCashOfferList;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityAddCashBinding;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.ADDAMOUNTOFFER;
import static
com.elevendreamer.APICallingPackage.Constants.ADDAMOUNTOFFERTYPE;

public class AddCashActivity extends AppCompatActivity implements ResponseManager
{

    AddCashActivity activity;
    Context context;
    ResponseManager responseManager;
    APIRequestManager apiRequestManager;

    SessionManager sessionManager;



    String FinalAmountToAdd;
    String EntryFee;
    AdapterAddCashOffertList adapterAddCashOfferList;
    public static String Activity = "";

    ActivityAddCashBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = DataBindingUtil.setContentView(this, R.layout.activity_add_cash);
        context = activity = this;
        sessionManager = new SessionManager();
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);



binding.head.tvHeaderName.setText(getResources().getString(R.string.add_cash_head));

        binding.head.imBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onBackPressed();
            }
        });



        binding.RVAddCashOffer.setHasFixedSize(true);
        binding.RVAddCashOffer.setNestedScrollingEnabled(false);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(activity);
```

```
binding.RVAddCashOffer.setLayoutManager(mLayoutManager);
//Rv_PlayerList.setItemAnimator(new DefaultItemAnimator());
binding.RVAddCashOffer.setItemAnimator(null);


try {
    EntryFee = getIntent().getStringExtra("EntryFee");

    if (EntryFee != null) {
        binding.etAddCashEnterAmount.setText(String.valueOf(EntryFee));
    }

} catch (Exception e) {
    e.printStackTrace();
}

try {
    Activity = getIntent().getStringExtra("Activity");
    System.out.print(Activity);
    if (Activity == null) {
        Activity = "";
    }
} catch (Exception e) {
    e.printStackTrace();
}


binding.tvAddCash.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
            FinalAmountToAdd = binding.etAddCashEnterAmount.getText().toString();

            if (FinalAmountToAdd.equals("")) {
                ShowToast(context, getResources().getString(R.string.enter_valid_amt));
                FinalAmountToAdd = "0";
            } else if (Integer.parseInt(FinalAmountToAdd) < 10) {
                ShowToast(context, getResources().getString(R.string.enter_min_amt));

            } else {
                Intent i = new Intent(activity, PaymentOptionActivity.class);
                i.putExtra("FinalAmount", FinalAmountToAdd);

                startActivity(i);
            }
        }
    });

    binding.tvOneHundred.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

binding.etAddCashEnterAmount.setText(getResources().getString(R.string.hundered));
        }
    });
    binding.tvTwoHundred.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

binding.etAddCashEnterAmount.setText(getResources().getString(R.string.two_hundred)
);
        }
    });
    binding.tvFiveHundred.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
binding.etAddCashEnterAmount.setText(getResources().getString(R.string.five_hundred)
);
        }
    });

    CallAddAmountOffer(true);

}

private void CallAddAmountOffer(boolean isShowLoader) {
    try {

        apiRequestManager.callAPI(ADDAMOUNTOFFER,
                createRequestJson(), context, activity, ADDAMOUNTOFFERTYPE,
                isShowLoader, responseManager);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

JSONObject createRequestJson() {
    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("user_id", sessionManager.getUser(context).getUser_id());
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return jsonObject;
}



@Override
public void getResult(Context mContext, String type, String message, JSONObject
result) {
    binding.RVAddCashOffer.setVisibility(View.VISIBLE);
    binding.LLAddCashOffer.setVisibility(View.VISIBLE);
    try {
        JSONArray jsonArray = result.getJSONArray("data");
        //Log.e("Data",jsonArray.toString());
        List<BeanAddCashOfferList> beanContestLists = new
Gson().fromJson(jsonArray.toString(),
                new TypeToken<List<BeanAddCashOfferList>>() {
                }.getType());
        adapterAddCashOfferList = new AdapterAddCashOffertList(beanContestLists,
activity);
        binding.RVAddCashOffer.setAdapter(adapterAddCashOfferList);

    } catch (Exception e) {
        e.printStackTrace();
    }

    adapterAddCashOfferList.notifyDataSetChanged();
}
@Override
public void onError(Context mContext, String type, String message) {


    binding.RVAddCashOffer.setVisibility(View.GONE);
    binding.LLAddCashOffer.setVisibility(View.GONE);


}


public class AdapterAddCashOffertList extends
RecyclerView.Adapter<AdapterAddCashOffertList.MyViewHolder> {
    private List<BeanAddCashOfferList> mListenerList;
    Context mContext;



    public AdapterAddCashOffertList(List<BeanAddCashOfferList> mListenerList,
Context context) {
        mContext = context;
        this.mListenerList = mListenerList;


    }


    public class MyViewHolder extends RecyclerView.ViewHolder {
        TextView tv_BonusCashLimit,tv_BonusOfferAmount;
        public MyViewHolder(View view) {
            super(view);
            tv_BonusCashLimit = view.findViewById(R.id.tv_BonusCashLimit);
            tv_BonusOfferAmount=view.findViewById(R.id.tv_BonusOfferAmount);
        }


    }
    @Override
    public int getItemCount() {
        return mListenerList.size();
    }

    @Override
    public AdapterAddCashOffertList.MyViewHolder onCreateViewHolder(ViewGroup
parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.adapter_add_cash_offer, parent, false);

        return new AdapterAddCashOffertList.MyViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(final AdapterAddCashOffertList.MyViewHolder
holder, final int position) {


        final String max_range = mListenerList.get(position).getMax_range();
        final  String min_range=mListenerList.get(position).getMin_range();
        final  String amount=mListenerList.get(position).getAmount();
        if (!max_range.equals(""))
        {
            holder.tv_BonusCashLimit.setText("Add Cash "+min_range+" ₹ to
"+max_range+" ₹");
            holder.tv_BonusOfferAmount.setText("Get "+amount+ " ₹ Bonus ");
        }

    }

}
```

```
}
```

PRICE CODE:

```
[10:15 pm, 31/05/2022] Ashuaa AAND: import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.BeanAddCashOfferList;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityAddCashBinding;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.List;


import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.ADDAMOUNTOFFER;
import static
com.elevendreamer.APICallingPackage.Constants.ADDAMOUNTOFFERTYPE;


public class AddCashActivity extends AppCompatActivity implements ResponseManager
{


    AddCashActivity activity;
    Context context;
    ResponseManager responseManager;
    APIRequestManager apiRequestManager;


    SessionManager sessionManager;




    String FinalAmountToAdd;
    String EntryFee;
    AdapterAddCashOffertList adapterAddCashOfferList;




    public static String Activity = "";


    ActivityAddCashBinding binding;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = DataBindingUtil.setContentView(this, R.layout.activity_add_cash);
        context = activity = this;
        sessionManager = new SessionManager();
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);



binding.head.tvHeaderName.setText(getResources().getString(R.string.add_cash_head));

        binding.head.imBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onBackPressed();
            }
        });



        binding.RVAddCashOffer.setHasFixedSize(true);
        binding.RVAddCashOffer.setNestedScrollingEnabled(false);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(activity);
        binding.RVAddCashOffer.setLayoutManager(mLayoutManager);
        //Rv_PlayerList.setItemAnimator(new DefaultItemAnimator());
        binding.RVAddCashOffer.setItemAnimator(null);



        try {
            EntryFee = getIntent().getStringExtra("EntryFee");

            if (EntryFee != null) {
                binding.etAddCashEnterAmount.setText(String.valueOf(EntryFee));
```

```
    }

} catch (Exception e) {
    e.printStackTrace();
}

try {
    Activity = getIntent().getStringExtra("Activity");
    System.out.print(Activity);
    if (Activity == null) {
        Activity = "";
    }
} catch (Exception e) {
    e.printStackTrace();
}


binding.tvAddCash.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        FinalAmountToAdd = binding.etAddCashEnterAmount.getText().toString();

        if (FinalAmountToAdd.equals("")) {
            ShowToast(context, getResources().getString(R.string.enter_valid_amt));
            FinalAmountToAdd = "0";
        } else if (Integer.parseInt(FinalAmountToAdd) < 10) {
            ShowToast(context, getResources().getString(R.string.enter_min_amt));

        } else {
            Intent i = new Intent(activity, PaymentOptionActivity.class);
```

```
                    i.putExtra("FinalAmount", FinalAmountToAdd);

                    startActivity(i);
                }
            }
        });

        binding.tvOneHundred.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

binding.etAddCashEnterAmount.setText(getResources().getString(R.string.hundered));
            }
        });
        binding.tvTwoHundred.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

binding.etAddCashEnterAmount.setText(getResources().getString(R.string.two_hundred)
);
            }
        });
        binding.tvFiveHundred.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

binding.etAddCashEnterAmount.setText(getResources().getString(R.string.five_hundred)
);
            }
        });

        CallAddAmountOffer(true);
```

```
    }

    private void CallAddAmountOffer(boolean isShowLoader) {
        try {

            apiRequestManager.callAPI(ADDAMOUNTOFFER,
                    createRequestJson(), context, activity, ADDAMOUNTOFFERTYPE,
                    isShowLoader, responseManager);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJson() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("user_id", sessionManager.getUser(context).getUser_id());

        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }


    @Override
    public void getResult(Context mContext, String type, String message, JSONObject
result) {
        binding.RVAddCashOffer.setVisibility(View.VISIBLE);
        binding.LLAddCashOffer.setVisibility(View.VISIBLE);
        try {
            JSONArray jsonArray = result.getJSONArray("data");
            //Log.e("Data",jsonArray.toString());
            List<BeanAddCashOfferList> beanContestLists = new Gson().fromJson(jsonArray.toString(),
                    new TypeToken<List<BeanAddCashOfferList>>() {
                    }.getType());
            adapterAddCashOfferList = new AdapterAddCashOffertList(beanContestLists, activity);
            binding.RVAddCashOffer.setAdapter(adapterAddCashOfferList);


        } catch (Exception e) {
            e.printStackTrace();
        }


        adapterAddCashOfferList.notifyDataSetChanged();
    }



    @Override
    public void onError(Context mContext, String type, String message) {


        binding.RVAddCashOffer.setVisibility(View.GONE);
        binding.LLAddCashOffer.setVisibility(View.GONE);


    }


    public class AdapterAddCashOffertList extends RecyclerView.Adapter<AdapterAddCashOffertList.MyViewHolder> {
        private List<BeanAddCashOfferList> mListenerList;
        Context mContext;



        public AdapterAddCashOffertList(List<BeanAddCashOfferList> mListenerList,
    Context context) {
            mContext = context;
            this.mListenerList = mListenerList;

        }

        public class MyViewHolder extends RecyclerView.ViewHolder {
            TextView tv_BonusCashLimit,tv_BonusOfferAmount;
            public MyViewHolder(View view) {
                super(view);
                tv_BonusCashLimit = view.findViewById(R.id.tv_BonusCashLimit);
                tv_BonusOfferAmount=view.findViewById(R.id.tv_BonusOfferAmount);
            }

        }
        @Override
        public int getItemCount() {
            return mListenerList.size();
        }

        @Override
        public AdapterAddCashOffertList.MyViewHolder onCreateViewHolder(ViewGroup
    parent, int viewType) {
            View itemView = LayoutInflater.from(parent.getContext())
                    .inflate(R.layout.adapter_add_cash_offer, parent, false);

            return new AdapterAddCashOffertList.MyViewHolder(itemView);
        }


        @Override
        public void onBindViewHolder(final AdapterAddCashOffertList.MyViewHolder holder, final int position) {



            final String max_range = mListenerList.get(position).getMax_range();
            final  String min_range=mListenerList.get(position).getMin_range();
            final  String amount=mListenerList.get(position).getAmount();
            if (!max_range.equals(""))
            {
                holder.tv_BonusCashLimit.setText("Add Cash "+min_range+" ₹ to "+max_range+" ₹");
                holder.tv_BonusOfferAmount.setText("Get "+amount+ " ₹ Bonus ");
            }

        }

    }
}
```

[10:16 pm, 31/05/2022] Ashuaa AAND: rank code

[10:16 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

```
import android.content.Context;
import android.os.Bundle;

import androidx.databinding.DataBindingUtil;
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;
import com.elevendreamer.databinding.ActivityGlobalRankBinding;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.eleve…
[10:17 pm, 31/05/2022] Ashuaa AAND: prize code
[10:17 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.elevendreamer.databinding.ActivitySelectPrizeCreateBinding;
import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.BeanRank;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.CREATECONTESTRANK;
import static com.elevendreamer.APICallingPackage.Config.CREATEOWNCONTEST;
import static
com.elevendreamer.APICallingPackage.Constants.CREATEOWNCONTESTTYPE;
import static com.elevendreamer.APICallingPackage.Constants.RANKLISTTYPE;

public class SelectPrizeCreateActivity extends AppCompatActivity implements
ResponseManager {

    String ContestName, ContestSize, ContestWinningAmount, EntryFees, MatchId;


    SelectPrizeCreateActivity activity;
    Context context;
    ResponseManager responseManager;
    APIRequestManager apiRequestManager;
    AdapterRankList adapterRankList;
    SessionManager sessionManager;
    String WinningBreakupID = "";

    ActivitySelectPrizeCreateBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = DataBindingUtil.setContentView(this,
R.layout.activity_select_prize_create);

        Intent i = getIntent();
        ContestName = i.getStringExtra("ContestName");
        ContestSize = String.valueOf(i.getIntExtra("ContestSize", 0));
        ContestWinningAmount = String.valueOf(i.getIntExtra("ContestWinningAmount",
0));
        EntryFees = String.valueOf(i.getDoubleExtra("EntryFees", 0));
        MatchId = i.getStringExtra("MatchId");

        context = activity = this;
        initViews();
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        sessionManager = new SessionManager();

        binding.tvCEntryFees.setText("₹" + EntryFees);
        binding.tvCPrizePool.setText("₹" + ContestWinningAmount);
        binding.tvCSize.setText("" + ContestSize);
        callRankList(true);
        binding.RLBottomFinalCreateMyContest.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (WinningBreakupID.equals("")) {
                    ShowToast(activity,
getResources().getString(R.string.select_winning_breakup) +
                            "");
                } else {
                    callCreateContest(true);
                }
            }
        });

    }

    public void initViews() {



        binding.head.tvHeaderName.setText("" + ContestName);
        binding.head.imBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onBackPressed();
            }
        });

        binding.RvRankList.setHasFixedSize(true);
        binding.RvRankList.setNestedScrollingEnabled(false);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(activity);
        binding.RvRankList.setLayoutManager(mLayoutManager);
        binding.RvRankList.setItemAnimator(new DefaultItemAnimator());



    }

    private void callRankList(boolean isShowLoader) {
        try {
            apiRequestManager.callAPI(CREATECONTESTRANK,
                    createRequestJson1(), context, activity, RANKLISTTYPE,
                    isShowLoader, responseManager);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJson1() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("team_size", ContestSize);
            jsonObject.put("price", ContestWinningAmount);

        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }

    private void callCreateContest(boolean isShowLoader) {
        try {
            apiRequestManager.callAPI(CREATEOWNCONTEST,
                    createRequestJson2(), context, activity, CREATEOWNCONTESTTYPE,
                    isShowLoader, responseManager);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJson2() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("user_id", sessionManager.getUser(context).getUser_id());
            jsonObject.put("userContestName", ContestName);
            jsonObject.put("userWinners", ContestWinningAmount);
            jsonObject.put("userTotalteam", ContestSize);
            jsonObject.put("userEntry", EntryFees);
            jsonObject.put("userMatchid", MatchId);
            jsonObject.put("breakupId", WinningBreakupID);

        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }

    @Override
    public void getResult(Context mContext, String type, String message, JSONObject
result) {
        if (type.equals(CREATEOWNCONTESTTYPE)) {
```

```
        try {
            ContestListActivity.ContestId = result.getString("user_Contestid");
            ContestListActivity.MyContestCode = result.getString("unique_code");
            Intent i = new Intent(activity, JoinContestActivity.class);
            i.putExtra("EntryFee", EntryFees);
            i.putExtra("ContestCode", ContestListActivity.MyContestCode);
            startActivity(i);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else {
        try {
            JSONArray jsonArray = result.getJSONArray("data");

            ArrayList<BeanRank> arr_beapb = new ArrayList<>();
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject userData = jsonArray.getJSONObject(i);
                String Id = userData.getString("id");
                String winners_count = userData.getString("winners_count");
                JSONArray rankingArray = userData.getJSONArray("ranking");

                BeanRank b = new BeanRank();
                b.setId(Id);
                b.setWinners_count(winners_count);
                b.setRanking(rankingArray);
                arr_beapb.add(b);

                adapterRankList = new AdapterRankList(arr_beapb, activity);
                binding.RvRankList.setAdapter(adapterRankList);
                }



            } catch (Exception e) {
                e.printStackTrace();
            }
            adapterRankList.notifyDataSetChanged();

        }



    }

    @Override
    public void onError(Context mContext, String type, String message) {
        ShowToast(context, message);
    }



    public class AdapterRankList extends
RecyclerView.Adapter<AdapterRankList.MyViewHolder> {
        private List<BeanRank> mListenerList;
        Context mContext;
        int selectedPosition=-1;
        int K= 0;

        public AdapterRankList(List<BeanRank> mListenerList, Context context) {
            mContext = context;
            this.mListenerList = mListenerList;

        }
```

```
public class MyViewHolder extends RecyclerView.ViewHolder {
    TextView tv_AdaWinnersCount;
    LinearLayout LL_AdawinnersList,LL_Item;

    public MyViewHolder(View view) {
        super(view);
        tv_AdaWinnersCount = view.findViewById(R.id.tv_AdaWinnersCount);
        LL_AdawinnersList = view.findViewById(R.id.LL_AdawinnersList);
        LL_Item = view.findViewById(R.id.LL_Item);
    }

}
@Override
public int getItemCount() {
    return mListenerList.size();
}

@Override
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View itemView = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.adapter_user_contest, parent, false);
    return new MyViewHolder(itemView);
}

@Override
public void onBindViewHolder(final MyViewHolder holder, final int position) {

    String Id = mListenerList.get(position).getId();
    String  WinnersCount = mListenerList.get(position).getWinners_count();
```

```
JSONArray  RankingArray = mListenerList.get(position).getRanking();

holder.tv_AdaWinnersCount.setText(WinnersCount+" Winners");

if (K==0) {
    if (RankingArray.length() > 0) {
        for (int i = 0; i < RankingArray.length(); i++) {

            View to_add =
LayoutInflater.from(context).inflate(R.layout.item_own_rank,
                    holder.LL_AdawinnersList, false);
            TextView tv_Rank = to_add.findViewById(R.id.tv_Rank);
            TextView tv_Percent = to_add.findViewById(R.id.tv_Percent);
            TextView tv_Price = to_add.findViewById(R.id.tv_Price);
            try {
                JSONObject userData = RankingArray.getJSONObject(i);
                String RankId = userData.getString("id");
                String rank = userData.getString("rank");
                String poolprice = userData.getString("poolprice");
                String price_percentage = userData.getString("price_percentage");
                tv_Rank.setText("Rank: " + rank);
                tv_Price.setText("₹ " + poolprice);
                tv_Percent.setText(price_percentage + "%");
                holder.LL_AdawinnersList.addView(to_add);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
            if(selectedPosition==position)

holder.LL_Item.setBackgroundResource(R.drawable.winning_breakup_selected);
            else {
                holder.LL_Item.setBackgroundResource(R.drawable.edittext_back);
            }

            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    WinningBreakupID = mListenerList.get(position).getId();
                    selectedPosition=position;
                    K=1;
                    notifyDataSetChanged();
                }
            });



        }

    }
}
```

API INTIGRATION :

```
[10:17 pm, 31/05/2022] Ashuaa AAND: package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.databinding.DataBindingUtil;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.elevendreamer.databinding.ActivitySelectPrizeCreateBinding;
import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.BeanRank;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;

import static com.elevendreamer.APICallingPackage.Class.Validations.ShowToast;
import static com.elevendreamer.APICallingPackage.Config.CREATECONTESTRANK;
import static com.elevendreamer.APICallingPackage.Config.CREATEOWNCONTEST;
import static
com.elevendreamer.APICallingPackage.Constants.CREATEOWNCONTESTTYPE;
import static com.elevendreamer.APICallingPackage.Constants.RANKLISTTYPE;

public class SelectPrizeCreateActivity extends AppCompatActivity implements
ResponseManager {

    String ContestName, ContestSize, ContestWinningAmount, EntryFees, MatchId;

    SelectPrizeCreateActivity activity;
    Context context;
    ResponseManager responseManager;
    APIRequestManager apiRequestManager;
    AdapterRankList adapterRankList;
    SessionManager sessionManager;
    String WinningBreakupID = "";

    ActivitySelectPrizeCreateBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = DataBindingUtil.setContentView(this,
R.layout.activity_select_prize_create);
```

```
        Intent i = getIntent();
        ContestName = i.getStringExtra("ContestName");
        ContestSize = String.valueOf(i.getIntExtra("ContestSize", 0));
        ContestWinningAmount = String.valueOf(i.getIntExtra("ContestWinningAmount",
0));
        EntryFees = String.valueOf(i.getDoubleExtra("EntryFees", 0));
        MatchId = i.getStringExtra("MatchId");

        context = activity = this;
        initViews();
        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);
        sessionManager = new SessionManager();

        binding.tvCEntryFees.setText("₹" + EntryFees);
        binding.tvCPrizePool.setText("₹" + ContestWinningAmount);
        binding.tvCSize.setText("" + ContestSize);
        callRankList(true);

        binding.RLBottomFinalCreateMyContest.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (WinningBreakupID.equals("")) {
                    ShowToast(activity,
getResources().getString(R.string.select_winning_breakup) +
                            "");
                } else {
                    callCreateContest(true);
                }
            }
        });

    }

    public void initViews() {



        binding.head.tvHeaderName.setText("" + ContestName);
        binding.head.imBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onBackPressed();
            }
        });

        binding.RvRankList.setHasFixedSize(true);
        binding.RvRankList.setNestedScrollingEnabled(false);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(activity);
        binding.RvRankList.setLayoutManager(mLayoutManager);
        binding.RvRankList.setItemAnimator(new DefaultItemAnimator());



    }

    private void callRankList(boolean isShowLoader) {
        try {
            apiRequestManager.callAPI(CREATECONTESTRANK,
                    createRequestJson1(), context, activity, RANKLISTTYPE,
                    isShowLoader, responseManager);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJson1() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("team_size", ContestSize);
            jsonObject.put("price", ContestWinningAmount);

        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }

    private void callCreateContest(boolean isShowLoader) {
        try {
            apiRequestManager.callAPI(CREATEOWNCONTEST,
                    createRequestJson2(), context, activity, CREATEOWNCONTESTTYPE,
                    isShowLoader, responseManager);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJson2() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("user_id", sessionManager.getUser(context).getUser_id());
            jsonObject.put("userContestName", ContestName);
            jsonObject.put("userWinners", ContestWinningAmount);
            jsonObject.put("userTotalteam", ContestSize);
            jsonObject.put("userEntry", EntryFees);
            jsonObject.put("userMatchid", MatchId);
            jsonObject.put("breakupId", WinningBreakupID);

        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }

    @Override
    public void getResult(Context mContext, String type, String message, JSONObject
result) {
        if (type.equals(CREATEOWNCONTESTTYPE)) {
            try {
                ContestListActivity.ContestId = result.getString("user_Contestid");
                ContestListActivity.MyContestCode = result.getString("unique_code");
                Intent i = new Intent(activity, JoinContestActivity.class);
                i.putExtra("EntryFee", EntryFees);
                i.putExtra("ContestCode", ContestListActivity.MyContestCode);
                startActivity(i);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            try {
                JSONArray jsonArray = result.getJSONArray("data");
            ArrayList<BeanRank> arr_beapb = new ArrayList<>();
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject userData = jsonArray.getJSONObject(i);
                String Id = userData.getString("id");
                String winners_count = userData.getString("winners_count");
                JSONArray rankingArray = userData.getJSONArray("ranking");

                BeanRank b = new BeanRank();
                b.setId(Id);
                b.setWinners_count(winners_count);
                b.setRanking(rankingArray);
                arr_beapb.add(b);

                adapterRankList = new AdapterRankList(arr_beapb, activity);
                binding.RvRankList.setAdapter(adapterRankList);

            }



        } catch (Exception e) {
            e.printStackTrace();
        }
        adapterRankList.notifyDataSetChanged();

    }



}
```

```
    @Override
    public void onError(Context mContext, String type, String message) {
        ShowToast(context, message);
    }



    public class AdapterRankList extends
RecyclerView.Adapter<AdapterRankList.MyViewHolder> {
        private List<BeanRank> mListenerList;
        Context mContext;
        int selectedPosition=-1;
        int K= 0;

        public AdapterRankList(List<BeanRank> mListenerList, Context context) {
            mContext = context;
            this.mListenerList = mListenerList;

        }

        public class MyViewHolder extends RecyclerView.ViewHolder {
            TextView tv_AdaWinnersCount;
            LinearLayout LL_AdawinnersList,LL_Item;

            public MyViewHolder(View view) {
                super(view);
                tv_AdaWinnersCount = view.findViewById(R.id.tv_AdaWinnersCount);
                LL_AdawinnersList = view.findViewById(R.id.LL_AdawinnersList);
                LL_Item = view.findViewById(R.id.LL_Item);
            }

        }
```

```
@Override
public int getItemCount() {
   return mListenerList.size();
}

@Override
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
   View itemView = LayoutInflater.from(parent.getContext())
         .inflate(R.layout.adapter_user_contest, parent, false);
   return new MyViewHolder(itemView);
}

@Override
public void onBindViewHolder(final MyViewHolder holder, final int position) {

   String Id = mListenerList.get(position).getId();
   String  WinnersCount = mListenerList.get(position).getWinners_count();
   JSONArray  RankingArray = mListenerList.get(position).getRanking();

   holder.tv_AdaWinnersCount.setText(WinnersCount+" Winners");

   if (K==0) {
      if (RankingArray.length() > 0) {
         for (int i = 0; i < RankingArray.length(); i++) {

            View to_add =
LayoutInflater.from(context).inflate(R.layout.item_own_rank,
                  holder.LL_AdawinnersList, false);
            TextView tv_Rank = to_add.findViewById(R.id.tv_Rank);
            TextView tv_Percent = to_add.findViewById(R.id.tv_Percent);
            TextView tv_Price = to_add.findViewById(R.id.tv_Price);
```

```
                        try {
                            JSONObject userData = RankingArray.getJSONObject(i);
                            String RankId = userData.getString("id");
                            String rank = userData.getString("rank");
                            String poolprice = userData.getString("poolprice");
                            String price_percentage = userData.getString("price_percentage");
                            tv_Rank.setText("Rank: " + rank);
                            tv_Price.setText("₹ " + poolprice);
                            tv_Percent.setText(price_percentage + "%");
                            holder.LL_AdawinnersList.addView(to_add);
                        } catch (Exception e) {
                            e.printStackTrace();
                        }
                    }
                }
            }



            if(selectedPosition==position)

holder.LL_Item.setBackgroundResource(R.drawable.winning_breakup_selected);
            else {
                holder.LL_Item.setBackgroundResource(R.drawable.edittext_back);
            }

            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    WinningBreakupID = mListenerList.get(position).getId();
                    selectedPosition=position;
                    K=1;
                    notifyDataSetChanged();
                }
            });



        }

    }
}
[10:18 pm, 31/05/2022] Ashuaa AAND: api integration
[10:18 pm, 31/05/2022] Ashuaa AAND: package
com.elevendreamer.APICallingPackage.Class;

import android.app.Activity;
import android.content.Context;

import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.APICallingPackage.Interface.ServerResponseListner;
import com.elevendreamer.APICallingPackage.Interface.VolleyRestClient;

import org.json.JSONException;
import org.json.JSONObject;


public class APIRequestManager implements ServerResponseListner {



    Context mContext;
    ResponseManager responseManager;
    private VolleyRestClient volleyRestClient;


    public APIRequestManager(Context mContext) {
        this.mContext = mContext;


    }



    public void callAPI(String url, JSONObject jsonObject, Context mContext, Activity
activity,
                        String type, boolean isShowProgress,
                        ResponseManager responseManager) throws JSONException {
        this.responseManager = responseManager;
        volleyRestClient = new VolleyApiCalling();
        volleyRestClient.callRestApi(url, jsonObject, mContext, activity, type,
                this, isShowProgress);


    }



    @Override
    public void onSucess(JSONObject response, String type, String message) {



        //Response only consist data object/array/string


        if (response != null && !response.equals("")) {
            try {


                responseManager.getResult(mContext,type,message,response);
            }
            catch (Exception e){
                e.printStackTrace();
            }
            }




    }

    @Override
    public void onError(String error, String type) {
            responseManager.onError(mContext,type,error);

    }
}
```

API RESPONSES :

```
package com.elevendreamer.APICallingPackage.Interface;

import android.content.Context;

import org.json.JSONObject;


public interface ResponseManager {
    public void getResult(Context mContext, String type, String message, JSONObject result);

    public void onError(Context mContext, String type, String message);

}
```

SERVER RESPONSE:

```
package com.elevendreamer.APICallingPackage.Interface;

    import org.json.JSONObject;

    public interface ServerResponseListner
    {
        public void onSucess(JSONObject response, String type, String message);
        public void onError(String error, String type);
    }
```

LOGOUT CODE ;

```
package com.elevendreamer.activity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import com.bumptech.glide.Glide;
import com.bumptech.glide.load.engine.DiskCacheStrategy;
import com.elevendreamer.APICallingPackage.Config;
import com.elevendreamer.databinding.ActivityMyJoinedContestListBinding;
import com.google.android.material.bottomsheet.BottomSheetDialog;
import androidx.databinding.DataBindingUtil;
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.elevendreamer.APICallingPackage.Class.APIRequestManager;
import com.elevendreamer.APICallingPackage.Class.Validations;
import com.elevendreamer.APICallingPackage.Interface.ResponseManager;
import com.elevendreamer.Bean.BeanMyJoinedContestList;
import com.elevendreamer.Bean.BeanWiningInfoList;
import com.elevendreamer.R;
import com.elevendreamer.utils.SessionManager;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;
import static com.elevendreamer.APICallingPackage.Config.MYJOINCONTESTLIST;
import static com.elevendreamer.APICallingPackage.Config.WINNINGINFOLIST;
import static
com.elevendreamer.APICallingPackage.Constants.MYJOINCONTESTLISTTYPE;
import static
com.elevendreamer.APICallingPackage.Constants.WINNINGINFOLISTTYPE;

public class MyJoinedFixtureContestListActivity extends AppCompatActivity
implements ResponseManager {

    MyJoinedFixtureContestListActivity activity;
    Context context;
    AdapterMyJoinedContestList adapterMyJoinedContestList;

    ResponseManager responseManager;
    APIRequestManager apiRequestManager;
    public static String
IntentMatchId,IntentTime,IntenTeamsName,IntentTeamOneName,IntentTeamTwoName,
IntentT1Image,IntentT2Image;
    BottomSheetDialog dialog;
    public static String ContestId,Matchid;
    List<BeanWiningInfoList> beanWinningLists;
    String prize_pool,contest_description;

    SessionManager sessionManager;

    ActivityMyJoinedContestListBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding=
DataBindingUtil.setContentView(this,R.layout.activity_my_joined_contest_list);

        context = activity = this;
        initViews();

        sessionManager = new SessionManager();
        IntentMatchId = getIntent().getStringExtra("MatchId");
        IntentTime = getIntent().getStringExtra("Time");
        IntenTeamsName = getIntent().getStringExtra("TeamsName");
        IntentTeamOneName = getIntent().getStringExtra("TeamsOneName");
        IntentTeamTwoName = getIntent().getStringExtra("TeamsTwoName");
        IntentT1Image = getIntent().getStringExtra("T1Image");
        IntentT2Image = getIntent().getStringExtra("T2Image");

        binding.inclVsBck.tvHeadTeamOneName.setText(IntentTeamOneName);
        binding.inclVsBck.tvHeadTeamTwoName.setText(IntentTeamTwoName);
        binding.inclVsBck.tvContestTimer.setText(IntentTime);
        Glide.with(activity).load(Config.TEAMFLAGIMAGE +IntentT1Image)
                .diskCacheStrategy(DiskCacheStrategy.ALL)
                .into(binding.inclVsBck.imTeam1);
        Glide.with(activity).load(Config.TEAMFLAGIMAGE +IntentT2Image)
                .diskCacheStrategy(DiskCacheStrategy.ALL)
                .into(binding.inclVsBck.imTeam2);

        responseManager = this;
        apiRequestManager = new APIRequestManager(activity);



        binding.RvMyJoinedContestList.setHasFixedSize(true);
        binding.RvMyJoinedContestList.setNestedScrollingEnabled(false);
```

```
    RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(activity);
    binding.RvMyJoinedContestList.setLayoutManager(mLayoutManager);
    binding.RvMyJoinedContestList.setItemAnimator(new DefaultItemAnimator());



    Validations.CountDownTimer(IntentTime,binding.inclVsBck.tvContestTimer);


    binding.swipeRefreshLayout.post(new Runnable() {
                @Override
                public void run() {
                  binding.swipeRefreshLayout.setRefreshing(true);
                  callMyJoinedContestList(false);
                }
              }
    );


    binding.swipeRefreshLayout.setOnRefreshListener(new
SwipeRefreshLayout.OnRefreshListener() {
      @Override
      public void onRefresh() {


        callMyJoinedContestList(false);
      }
    });




  }


  public void initViews(){
        binding.head.tvHeaderName.setText("JOIN CONTESTS");
        binding.head.imBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                onBackPressed();
            }
        });



    }



    private void callMyJoinedContestList(boolean isShowLoader) {
        try {

            apiRequestManager.callAPI(MYJOINCONTESTLIST,
                    createRequestJson(), context, activity, MYJOINCONTESTLISTTYPE,
                    isShowLoader,responseManager);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJson() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("match_id", IntentMatchId);
            jsonObject.put("user_id", sessionManager.getUser(context).getUser_id());



        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonObject;
    }

    private void callWinningInfoList(boolean isShowLoader) {
        try {

            apiRequestManager.callAPI(WINNINGINFOLIST,
                    createRequestJsonWin(), context, activity, WINNINGINFOLISTTYPE,
                    isShowLoader,responseManager);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    JSONObject createRequestJsonWin() {
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("contest_id", ContestId);



        } catch (JSONException e) {
            e.printStackTrace();
        }
```

```
        return jsonObject;
    }



    @Override
    public void getResult(Context mContext, String type, String message, JSONObject
result) {




        if (type.equals(WINNINGINFOLISTTYPE)){
            try {
                JSONArray jsonArray = result.getJSONArray("data");

                beanWinningLists = new Gson().fromJson(jsonArray.toString(),
                        new TypeToken<List<BeanWiningInfoList>>() {
                        }.getType());

                dialog = new BottomSheetDialog(activity);
                dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
                dialog.setContentView(R.layout.dialog_winning_breakups);
                final TextView tv_DClose = dialog.findViewById(R.id.tv_DClose);
                final TextView tv_DTotalWinning
=dialog.findViewById(R.id.tv_DTotalWinning);
                final TextView tv_DBottomNote
=dialog.findViewById(R.id.tv_DBottomNote);
                final LinearLayout
LL_WinningBreackupList=dialog.findViewById(R.id.LL_WinningBreackupList);
                dialog.show();
                tv_DTotalWinning.setText("₹ "+prize_pool);
                tv_DBottomNote.setText("Note: "+contest_description);
                tv_DClose.setOnClickListener(new View.OnClickListener() {
```

```
                @Override
                public void onClick(View view) {
                    dialog.cancel();
                }
            });
            for (int i = 0; i <beanWinningLists .size(); i++) {


                View to_add =
LayoutInflater.from(context).inflate(R.layout.item_winning_breakup,
                        LL_WinningBreackupList,false);
                TextView tv_Rank = to_add.findViewById(R.id.tv_Rank);
                TextView tv_Price = to_add.findViewById(R.id.tv_Price);


                tv_Rank.setText("Rank: "+beanWinningLists.get(i).getRank());
                tv_Price.setText("₹ "+beanWinningLists.get(i).getPrice());


                LL_WinningBreackupList.addView(to_add);
            }



        }
        catch (Exception e){
            e.printStackTrace();
        }





    }
    else {
        binding.tvNoDataAvailable.setVisibility(View.GONE);
        binding.RvMyJoinedContestList.setVisibility(View.VISIBLE);
        binding.swipeRefreshLayout.setRefreshing(false);

        try {
            JSONArray jsonArray = result.getJSONArray("data");
            List<BeanMyJoinedContestList> beanContestLists = new
Gson().fromJson(jsonArray.toString(),
                    new TypeToken<List<BeanMyJoinedContestList>>() {
                    }.getType());
            adapterMyJoinedContestList = new
AdapterMyJoinedContestList(beanContestLists, activity);
            binding.RvMyJoinedContestList.setAdapter(adapterMyJoinedContestList);

        } catch (Exception e) {
            e.printStackTrace();
        }

        adapterMyJoinedContestList.notifyDataSetChanged();

    }
}

@Override
public void onError(Context mContext, String type, String message) {
    if (type.equals(MYJOINCONTESTLISTTYPE)){

        binding.tvNoDataAvailable.setVisibility(View.VISIBLE);
        binding.RvMyJoinedContestList.setVisibility(View.GONE);
        binding.swipeRefreshLayout.setRefreshing(false);
    }
}


public class AdapterMyJoinedContestList extends
RecyclerView.Adapter<AdapterMyJoinedContestList.MyViewHolder> {
    private List<BeanMyJoinedContestList> mListenerList;
    Context mContext;



    public AdapterMyJoinedContestList(List<BeanMyJoinedContestList>
mListenerList, Context context) {
        mContext = context;
        this.mListenerList = mListenerList;

    }

    public class MyViewHolder extends RecyclerView.ViewHolder {
        TextView
tv_LiveContestName,tv_LiveContestDesc,tv_TotalPrice,tv_WinnersCount,tv_EntryFees,t
v_TeamLeftCount,tv_TotalTeamCount
                    ,tv_JoinContest,tv_MyJoinedTeamCount;

        ProgressBar PB_EntryProgress;



        public MyViewHolder(View view) {
            super(view);
            tv_LiveContestName = view.findViewById(R.id.tv_contestName);
            tv_LiveContestDesc = view.findViewById(R.id.tv_LiveContestDesc);
            tv_TotalPrice = view.findViewById(R.id.tv_TotalPrice);
            tv_WinnersCount = view.findViewById(R.id.tv_WinnersCount);
            tv_EntryFees = view.findViewById(R.id.tv_EntryFees);
            tv_TeamLeftCount = view.findViewById(R.id.tv_TeamLeftCount);
            tv_TotalTeamCount = view.findViewById(R.id.tv_TotalTeamCount);
            PB_EntryProgress = view.findViewById(R.id.PB_EntryProgress);
            tv_MyJoinedTeamCount = view.findViewById(R.id.tv_MyJoinedTeamCount);



        }

    }
    @Override
    public int getItemCount() {
        return mListenerList.size();
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.adapter_my_joined_contest_list, parent, false);

        return new MyViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int position) {

        //holder.tv_JoinContest.setVisibility(View.INVISIBLE);
        final String contest_id = mListenerList.get(position).getContest_id();
        String contest_name= mListenerList.get(position).getContest_name();
        String contest_tag= mListenerList.get(position).getContest_tag();
        String winners= mListenerList.get(position).getWinners();
```

```
   prize_pool= mListenerList.get(position).getPrize_pool();
   String total_team= mListenerList.get(position).getTotal_team();
   String join_team= mListenerList.get(position).getJoin_team();
   String entry= mListenerList.get(position).getEntry();

   String contest_note1 = mListenerList.get(position).getContest_note1();
   String contest_note2= mListenerList.get(position).getContest_note2();
   String match_id= mListenerList.get(position).getMatch_id();
   String type= mListenerList.get(position).getType();
   String remaining_team= mListenerList.get(position).getRemaining_team();
   String joinedteamcount= mListenerList.get(position).getTeam_count();

   holder.tv_LiveContestName.setText(contest_name);
   holder.tv_LiveContestDesc.setText(contest_tag);
   holder.tv_MyJoinedTeamCount.setText("Joined with "+joinedteamcount+"
Team");
   holder.tv_TotalPrice.setText("₹ "+prize_pool);
   holder.tv_WinnersCount.setText(winners);
   holder.tv_EntryFees.setText("₹ "+entry);

   holder.tv_TeamLeftCount.setText(remaining_team+" Spots Left");
   holder.tv_TotalTeamCount.setText(total_team+" Teams");
   holder.PB_EntryProgress.setMax(Integer.parseInt(total_team));
   holder.PB_EntryProgress.setProgress(Integer.parseInt(join_team));

   holder.tv_WinnersCount.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View view) {
           ContestId = mListenerList.get(position).getContest_id();
           contest_description= mListenerList.get(position).getContest_description();
           prize_pool= mListenerList.get(position).getPrize_pool();
```

```
                    callWinningInfoList(true);
                }
            });

            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Matchid = IntentMatchId;
                    ContestId = mListenerList.get(position).getContest_id();
                    Intent i = new Intent(activity, MyFixtureContestDetailsActivity.class);
                    startActivity(i);
                }
            });

        }

    }

}
```