Table of Content

- Certificate
- Acknowledgement…….. [i]
- Abstract……...[ii]

1. Introduction…….…….….3
    1.1. Features
2. Software Development Model……...….4-6
    2.1. Software Engineering
    2.2. Software Life Cycle Model
    2.3. Need of SDLC
    2.4. SDLC Cycle
3. Water Fall Model…….…….….7-8
    3.1. Waterfall Model – Design
    3.2. Use of SDLC Model
    3.3. Application of Waterfall Model
    3.4. Advantages of Waterfall Model
    3.5. Disadvantages of Waterfall Model
4. Frontend…….9-17
    4.1. Java
    4.2. History of Java
    4.3. Application of Java
    4.4. Java (Keywords, Control Statement and Loops)
    4.5. Android Tutorial
    4.6. Activity and Intent
    4.7. Gradle Script and XML Layout
5. Backend (Database)…….....18-23
    5.1. Firebase
    5.2. History of Firebase
    5.3. Realtime Database with Java
    5.4. Object Oriented Programming Concept with Java
    5.5. Collection and Framework (for Android)
6. Flowchart……...24
7. Screenshot…….....25-31
    7.1. Slash Activity
    7.2. Main Activity
    7.3. Account Activity
    7.4. Card Activity
    7.5 Travel Activity
    7.6 Profile Activity
    7.7 Send_money Activity
    7.8 Payment_Sucess Activity
    7.9 Payment_Fail Activity
    7.10 Login_register Activity
    7.11 Login Activity
    7.12 Register Activity
    7.13 Validation Activity
    7.14 Change_pin Activity

8. Source Code…….....32-78
9.Conclusion……...….79
10.Future Scope……...80
11.Reference…….....81

List of Figures

1.SDLC……5
2.Waterfall model……....7
3. Android Architecture……..……13
4. Android Activity Lifecycle methods ……16

1. INTRODUCTION

- I am developing a Gramin Ebank android application which is very user friendly to use and it is also offer all service online and no need to visit branch. Service In this application you can track your banking activity (like credit, debit and other transactions).
- In the realm of Android app development, utilizing Java as the primary programming language and integrating Firebase as a backend service represents a powerful combination for creating robust and dynamic applications. This project report aims to encapsulate the journey of developing a native Android app, highlighting the seamless integration of Java's versatility with Firebase's real-time database, authentication, and hosting capabilities.
- Throughout this report, we will delve into the intricacies of leveraging Java's object-oriented principles to craft efficient and user-friendly interfaces, while harnessing Firebase's cloud-based services to enhance the app's functionality and performance. From user authentication and data storage to real-time updates and analytics, this project showcases the synergy between traditional programming languages and cutting-edge cloud technologies.

1.1. Features

- The innovative world of E-Bank, where convenience meets cutting-edge technology in the palm of your hand.
- Our Android application redefines banking by offering a seamless and user-friendly experience for all your financial needs.
- Say goodbye to long queues and branch visits - with E-Bank, all banking services are just a tap away.
- Track your banking activity effortlessly, from monitoring credit and debit transactions to staying updated on all financial interactions.
- Enjoy the convenience of accessing your account information anytime, anywhere, with real-time updates at your fingertips.
- Experience the power of online banking with secure and encrypted transactions for peace of mind.
- E-Bank empowers you to take control of your finances with comprehensive tracking and reporting features.
- Simplify your financial management with a range of services available at your convenience, eliminating the need for physical visits to the bank.

2. SOFTWARE DEVELOPMENT MODEL

2.1. Software Engineering:

- Software engineering is the engineering of development of software in a systematic method. Software engineering is the process of analysing user needs and designing, constructing, and testing end-user applications that will satisfy these needs through the use of software programming languages. Placation of engineering principles to software development.
-
- It is ape griming; software engineering is used for larger and more complex software systems, which is used as critical systems for businesses and organizations. Software engineering involves a number of fields that cover the process of engineering software and certification including: requirements gathering, software design, software construction, software maintenance, software configuration management, software engineering management, software development process management and creation, software engineering models and methods, software quality, software engineering professional practices as well as foundational computing and mathematical and engineering study.

- The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software

2.2. Software Life Cycle Model:

- A software life cycle model (also termed process model) is a pictorial and diagrammatic representation of the software life cycle. A life cycle model represents all the methods required to make a software product transit through itslife cycle stages. It also captures the structure in which these methods are to be undertaken.
- In other words, a life cycle model maps the various activities performed on a software product from its inception to retirement. Different life cycle models may plan the necessary development activities to phases in different ways. Thus, no element which life cycle model is followed, the essential activities are contained in all life cycle models though the action may be carried out in distinct orders in different life cycle models. During any life cycle stage, more than one activity may also be carried out.

2.3. Need of SDLC:

- The development team must determine a suitable life cycle model for a particular plan and then observe it.

- Without using an exact life cycle model, the development of a software product would not be in a systematic and disciplined manner. When a team is developing a software product, there must be a clear understanding among team representatives about when and what to do. Otherwise, it would point to chaos and project failure.

- This problem can be defined by using an example. Suppose a software development issue is divided into various parts and the parts are assigned to the team members. From then on, suppose the team representative is allowed the freedom to develop the roles assigned to them in whatever way they like. It is possible that one representative might start writing the code for his part, another

might choose to prepare the test documents first, and some other engineer might begin with the design phase of the roles assigned to him. This would be one of the perfect methods for project failure.

- A software life cycle model describes entry and exit criteria for each phase. A phase can begin only if its stage-entry criteria have been fulfilled. So without a software life cycle model, the entry and exit criteria for a stage cannot be recognized. Without software life cycle models, it becomes tough for software project managers to monitor the progress of the project.

2.4. SDLC Cycle:

- SDLC Cycle represents the process of developing software. SDLC framework includes the following steps:
- The stages of SDLC are as follows:

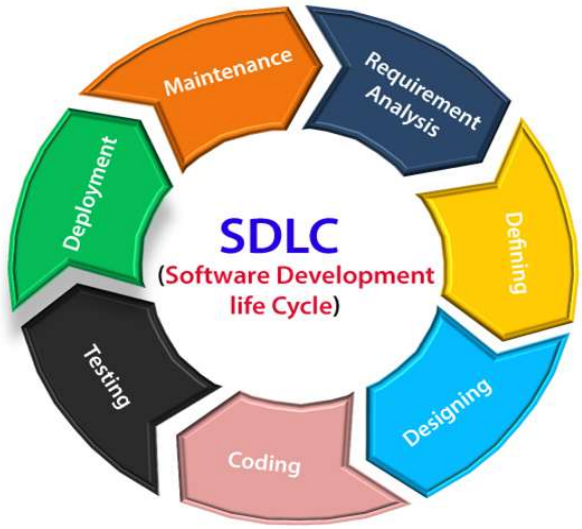


Figure1: -SDLC [1]

Stage1 Planning and requirement analysis:

- Requirement Analysis is the most important and necessary stage in SDLC.

- For Example: -A client wants to have an application which concerns money transactions. In this method, the requirement has to be precise like what kind of operations will be done, how it will be done, in which currency it will be done, etc.

Stage2 Defining Requirements:

- Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.
- This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.

Stage3 Designing the Software:

- The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.
-

Stage4 Developing the project:

- In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins with writing code.
- Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

Stage5 Testing:

- After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage.
- During this stage, unit testing, integration testing, system testing, acceptance testing are done.

Stage6 Deployment:

- Once the software is certified, and no bugs or errors are stated, then it is deployed.
- Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.
- After the software is deployed, then its maintenance begins.

Stage7 Maintenance:

- Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.
- This procedure where the care is taken for the developed product is known as maintenance.

3.WATER FALL MODEL

- The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The Waterfall model is the earliest SDLC approach that was used for software development.
- The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

3.1. Waterfall Model – Design:

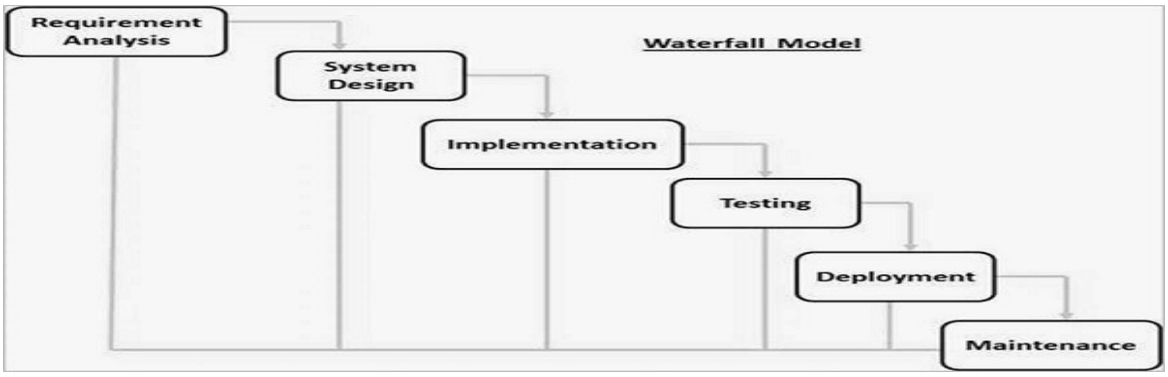


Figure 2: -Waterfall model [2]

1. Requirements analysis and specification phase:

- The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions, performance, and interfacing requirements of the software
- It describes the "what" of the system to be produced and not "how." In this phase, a large document called Software Requirement Specification (SRS) document is created which contained a detailed description of what the system will do in the common language.

2. Design Phase:

- This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language.
- It defines the overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

3. Implementation and unit testing:

- During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD.
- During testing, the code is thoroughly examined and modified. Small modules are tested in isolation initially. After that these modules are tested by writing some overhead code to check the interaction between these modules and the flow of intermediate output.

4. Integration and System Testing:

- This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results.
- Unit testing determines the efficiency of individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.

5. Deployment of system:

- Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

6. Operation and maintenance phase:

- There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

3.2. Use of Waterfall model:

When the requirements are constant and not changed regularly

- A project is short
- The situation is calm
- Where the tools and technology used is consistent and is not changing
- When resources are well prepared and are available to use.

3.3. Application of waterfall model:

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are –

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

3.4. Advantages of Waterfall model:

- This model is simple to implement and the number of resources that are required for it is minimal.
- The requirements are simple and explicitly declared; they remain unchanged during the entire project development.
- Easy to arrange tasks.

3.5. Disadvantages of Waterfall model:

- In this model, the risk factor is higher, so this model is not suitable for more significant and complex projects.
- This model cannot accept the changes in requirements during development.
- It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.

4. FRONTEND

4.1. Java:

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by *Sun Microsystems* (which is now the subsidiary of Oracle) in the year 1995. *James Gosling* is known as the father of Java. Before Java, its name was *Oak*. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

Java is an object-oriented programming language that enables you to run your code on many different hardware platforms, including desktops, mobile devices, and Raspberry PI.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

4.2. History of Java

The history of Java is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of Java starts with the Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc. However, it was best suited for internet programming. Later, Java technology was incorporated by Netscape.

Java Version History

Many java versions have been released till now. The current stable release of Java is Java SE 10.

1. JDK Alpha and Beta (1995)
2. JDK 1.0 (23rd Jan 1996)
3. JDK 1.1 (19th Feb 1997)
4. J2SE 1.2 (8th Dec 1998)
5. J2SE 1.3 (8th May 2000)
6. J2SE 1.4 (6th Feb 2002)
7. J2SE 5.0 (30th Sep 2004)
8. Java SE 6 (11th Dec 2006)
9. Java SE 7 (28th July 2011)
10. Java SE 8 (18th Mar 2014)
11. Java SE 9 (21st Sep 2017)
12. Java SE 10 (20th Mar 2018)
13. Java SE 11 (September 2018)
14. Java SE 12 (March 2019)
15. Java SE 13 (September 2019)
16. Java SE 14 (Mar 2020)
17. Java SE 15 (September 2020)
18. Java SE 16 (Mar 2021)
19. Java SE 17 (September 2021)
20. Java SE 18 (to be released by March 2022)

Since Java SE 8 release, the Oracle corporation follows a pattern in which every even version is release in March month and an odd version released in September month.

4.3. Application of Java

According to Sun, 3 billion devices run Java. There are many devices where Java is currently used. Some of them are as follows:

1. Desktop Applications such as acrobat reader, media player, antivirus, etc.
2. Web Applications such as irctc.co.in, javatpoint.com, etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games, etc.

Types of Java Applications

Web Application An application that runs on the server side and creates a dynamic page is called a web application. Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF, etc. technologies are used for creating web applications in Java.

Mobile Application An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

- **Java Platforms / Editions**

There are 4 platforms or editions of Java:

1) Java SE (Java Standard Edition)

It is a Java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.

2) Java EE (Java Enterprise Edition)

It is an enterprise platform that is mainly used to develop web and enterprise applications. It is built on top of the Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA, etc.

3) Java ME (Java Micro Edition)

It is a micro platform that is dedicated to mobile applications.

4) JavaFX

It is used to develop rich internet applications. It uses a lightweight user interface API

4.4. Java (Keywords, Control Statement and Loops)

- **Java Keywords**

Java keywords are also known as reserved words. Keywords are particular words that act as a key to a code. These are predefined words by Java so they cannot be used as a variable or object name or class name.

List of Java Keywords

A list of Java keywords or reserved words are given below:

1. abstract**:** Java abstract keyword is used to declare an abstract class. An abstract class can provide the implementation of the interface. It can have abstract and non-abstract methods.
2. boolean: Java boolean keyword is used to declare a variable as a boolean type. It can hold True and False values only.
3. break**:** Java break keyword is used to break the loop or switch statement. It breaks the current flow of the program at specified conditions.
4. do**:** Java do keyword is used in the control statement to declare a loop. It can iterate a part of the program several times.
5. double**:** Java double keyword is used to declare a variable that can hold 64-bit floating-point number.

6. else**:** Java else keyword is used to indicate the alternative branches in an if statement.
7. final**:** Java final keyword is used to indicate that a variable holds a constant value. It is used with a variable. It is used to restrict the user from updating the value of the variable.
8. for**:** Java for keyword is used to start a for loop. It is used to execute a set of instructions/functions repeatedly when some condition becomes true. If the number of iteration is fixed, it is recommended to use for loop.
9. if**:** Java if keyword tests the condition. It executes the if block if the condition is true.
10. implements**:** Java implements keyword is used to implement an interface.
11. import**:** Java import keyword makes classes and interfaces available and accessible to the current source code.
12. int**:** Java int keyword is used to declare a variable that can hold a 32-bit signed integer.
13. new**:** Java new keyword is used to create new objects.
14. null**:** Java null keyword is used to indicate that a reference does not refer to anything. It removes the garbage value.
15. package**:** Java package keyword is used to declare a Java package that includes the classes.
16. private**:** Java private keyword is an access modifier. It is used to indicate that a method or variable may be accessed only in the class in which it is declared.
17. protected**:** Java protected keyword is an access modifier. It can be accessible within the package and outside the package but through inheritance only. It can't be applied with the class.
18. public**:** Java public keyword is an access modifier. It is used to indicate that an item is accessible anywhere. It has the widest scope among all other modifiers.
19. return**:** Java return keyword is used to return from a method when its execution is complete.
20. static**:** Java static keyword is used to indicate that a variable or method is a class method. The static keyword in Java is mainly used for memory management.
21. super**:** Java super keyword is a reference variable that is used to refer to parent class objects. It can be used to invoke the immediate parent class method.
22. switch**:** The Java switch keyword contains a switch statement that executes code based on test value. The switch statement tests the equality of a variable against multiple values.
23. try**:** Java try keyword is used to start a block of code that will be tested for exceptions. The try block must be followed by either catch or finally block.
24. **void:** Java void keyword is used to specify that a method does not have a return value.
25. while**:** Java while keyword is used to start a while loop. This loop iterates a part of the program several times. If the number of iteration is not fixed, it is recommended to use the while loop.

- **Java Control Statements | Control Flow in Java**

Java compiler executes the code from top to bottom. The statements in the code are executed according to the order in which they appear. However, Java provides statements that can be used to control the flow of Java code. Such statements are called control flow statements. It is one of the fundamental features of Java, which provides a smooth flow of program.
Java provides three types of control flow statements.

1. Decision Making statements
    - if statements
    - switch statement
2. Loop statements
    - do while loop
    - while loop
    - for loop
    - for-each loop
3. Jump statements
    - break statement
    - continue statement

- **What is Android?**

Before learning all topics of android, it is required to know what is android. Android is a software package and linux based operating system for mobile devices such as tablet computers and smartphones.It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used. The goal of android project is to create a successful real-world product that improves the mobile experience for end users.There are many code names of android such as Lollipop, Kitkat, Jelly Bean, Ice cream Sandwich, Froyo, Ecliar, Donut etc which is covered in next page.

What is Open Handset Alliance (OHA)?

It's a consortium of 84 companies such as google, samsung, AKM, synaptics, KDDI, Garmin, Teleca, Ebay, Intel etc. It was established on 5th November, 2007, led by Google. It is committed to advance open standards, provide services and deploy handsets using the Android Plateform.

- **Features of Android**

After learning what is android, let's see the features of android. The important features of android are given below:

1) It is open-source.

2) Anyone can customize the Android Platform.

3) There are a lot of mobile applications that can be chosen by the consumer.

4) It provides many interesting features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc.It provides support for messaging services(SMS and MMS), web browser, storage (SQLite), connectivity (GSM, CDMA, Blue Tooth, Wi-Fi etc.), media, handset layout etc.

Categories of Android applications

There are many android applications in the market. The top categories are:

- Entertainment
- Personalization
- Music and Audio
- Social
- Media and Video
- Travel and Local etc.

Android Architecture

android architecture or **Android software stack** is categorized into five parts:

1. linux kernel
2. native libraries (middleware),
3. Android Runtime
4. Application Framework
5. Applications

Let's see the android architecture first.

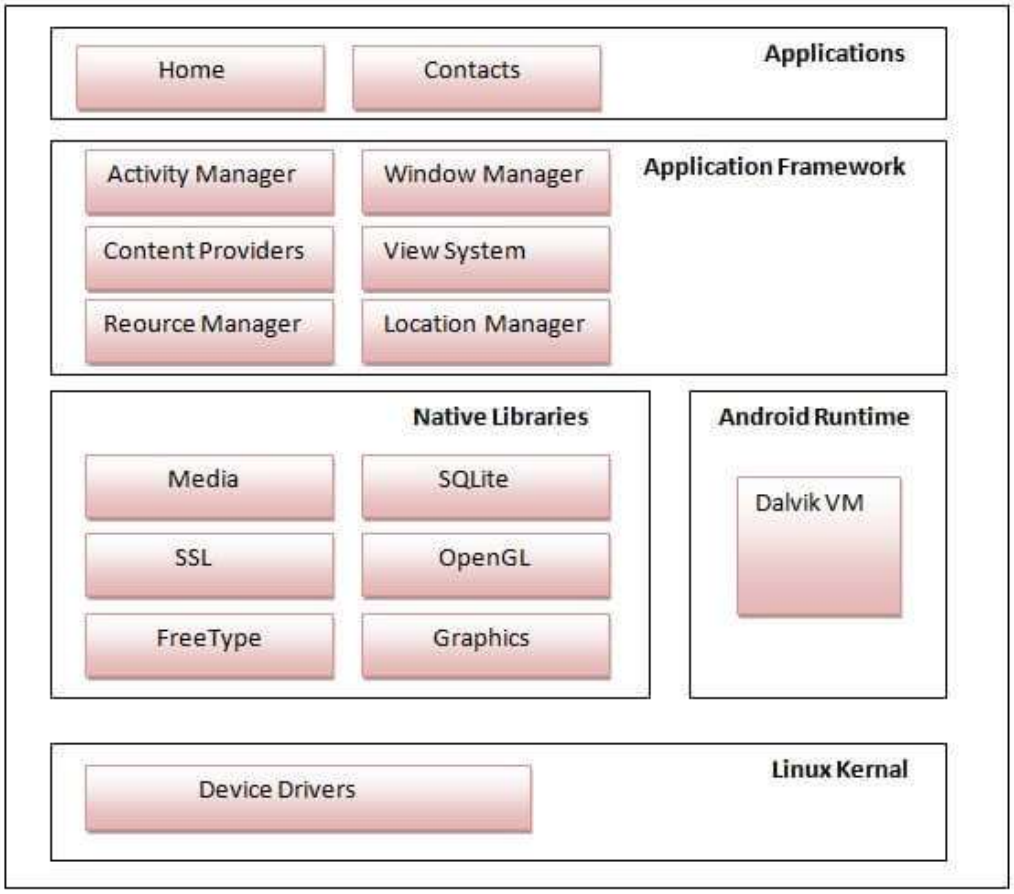


Figure 3: -android architecture [3]

1) Linux kernel

It is the heart of android architecture that exists at the root of android architecture. Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.

2) Native Libraries

On the top of linux kernel, their are Native libraries such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc.

3) Android Runtime

In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.

4) Android Framework

On the top of Native libraries and android runtime, there is android framework. Android framework includes Android API's such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.

5) Applications

On the top of android framework, there are applications. All applications such as home,

contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using linux kernal.

- **AndroidManifest.xml file in android**

  The AndroidManifest.xml file contains information of your package, including components of the application such as activities, services, broadcast receivers, content providers etc.

  It performs some other tasks also:

  - It is responsible to protect the application to access any protected parts by providing the permissions.
  - It also declares the android api that the application is going to use.
  - It lists the instrumentation classes. The instrumentation classes provides profiling and other informations. These informations are removed just before the application is published etc.

  This is the required xml file for all the android application and located inside the root directory.

  A simple AndroidManifest.xml file looks like this:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools">

   <uses-permission android:name="android.permission.INTERNET" />

   <application
      android:allowBackup="true"
      android:dataExtractionRules="@xml/data_extraction_rules"
      android:fullBackupContent="@xml/backup_rules"
      android:icon="@drawable/appicon"
      android:label="@string/app_name"
      android:roundIcon="@drawable/bnkl"
      android:supportsRtl="true"
      android:theme="@style/Theme.Ebank"
      tools:targetApi="31">
      <activity
         android:name=".change_pin"
         android:exported="false" />
      <activity
         android:name=".Failpayment"
         android:exported="false" />
      <activity
         android:name=".Sucess_payment"
         android:exported="false" />
      <activity
         android:name=".PaymentsucessActivity2"
         android:exported="false" />
      <activity
         android:name=".SendmoneyActivity2"
         android:exported="false" />
      <activity
         android:name=".ValidationActivity"
         android:exported="false" />
      <activity
         android:name=".TravelActivity"
         android:exported="false" />
      <activity
         android:name=".CardActivity"
         android:exported="false" />
      <activity
```

```
        android:name=".AccountActivity"
        android:exported="false" />
    <activity
        android:name=".Profile"
        android:exported="false" />
    <activity
        android:name=".Register"
        android:exported="false" />
    <activity
        android:name=".Login"
        android:exported="false" />
    <activity
        android:name=".Loginregister"
        android:exported="false" />
    <activity
        android:name=".SlashActivity2"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".MainActivity"
        android:exported="true" />
  </application>
```

</manifest>Elements of the AndroidManifest.xml file

- **Android R.java file**
- **Android R.java** is *an auto-generated file by aapt* (Android Asset Packaging Tool) that contains resource IDs for all the resources of res/ directory.
- If you create any component in the activity_main.xml file, id for the corresponding component is automatically created in this file. This id can be used in the activity source file to perform any action on the component.
- Android Toast Example
  Toast class is used to show notification for a particular interval of time. After sometime it disappears. It doesn't block the user interaction.
  Constants of Toast class
  There are only 2 constants of Toast class which are given below.

Constant	Description
public static final int LENGTH_LONG	displays view for the long duration of time.
public static final int LENGTH_SHORT	displays view for the short duration of time.

Methods of Toast class
The widely used methods of Toast class are given below.

Method	Description
public static Toast makeText(Context context, CharSequence text, int duration)	makes the toast containing text and duration.
public void show()	displays toast.
public void setMargin (float horizontalMargin, float verticalMargin)	changes the horizontal and vertical margin difference.

Android Toast Example

```
Toast.makeText(getApplicationContext(),"Hello Javatpoint",Toast.LENGTH_SHORT).show();
```

4.6. Activity and Intent

Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class. The android Activity is the subclass of ContextThemeWrapper class.

An activity is the single screen in android. It is like window or frame of Java.

By the help of activity, you can place all your UI components or widgets in a single screen.

The 7 lifecycle method of Activity describes how activity will behave at different states.

Android Activity Lifecycle methods

Let's see the 7 lifecycle methods of android activity.

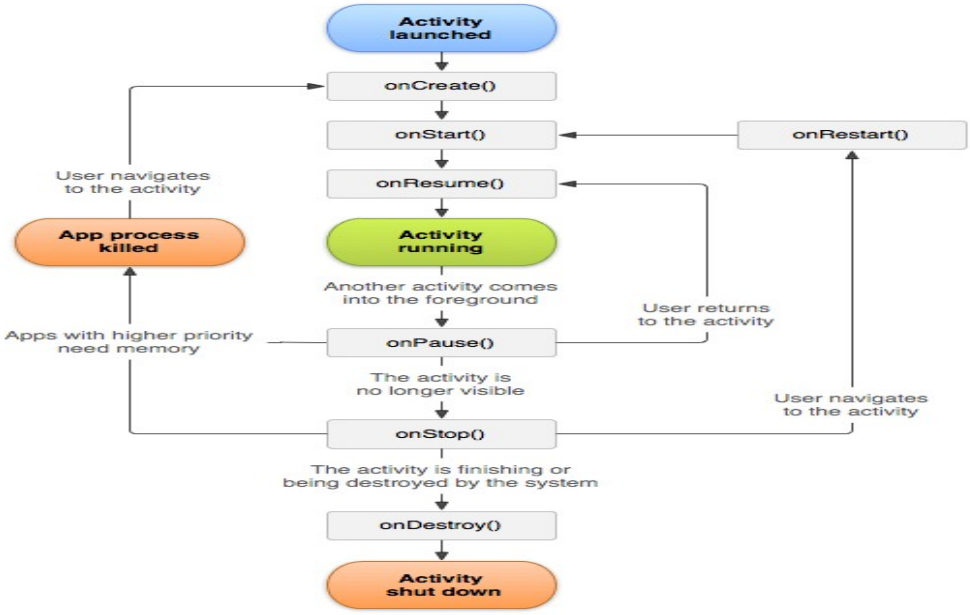


Figure 4: - Android Activity [4]

Android StartActivityForResult Example

By the help of android startActivityForResult() method, we can get result from another activity.

By the help of android startActivityForResult() method, we can send information from one activity to another and vice-versa. The android **startActivityForResult** method, requires a result from the second activity (activity to be invoked).

In such case, we need to override the onActivityResult method that is invoked automatically when second activity returns result.

Method Signature

There are two variants of startActivityForResult() method.

1. public void startActivityForResult (Intent intent, int requestCode)
2. public void startActivityForResult (Intent intent, int requestCode, Bundle options)

4.7. Gradle Script and XML Layout

Gradle is the build automation tool used in Android app development. It is responsible for managing project dependencies, building the app, and running various tasks. The Gradle scripts are written in Groovy or Kotlin DSL (Domain-Specific Language).

Project-level build.gradle file: This file defines the Gradle configuration for the entire project, including the Gradle version, build settings, and dependencies that are shared across all modules.

Module-level build.gradle file: This file defines the Gradle configuration for a specific module (e.g., the app module or a library module).

It specifies the Android plugin version, app properties (package name, app name, etc.), dependencies (libraries and external modules), build types, and more.

XML Layouts: In Android app development, the user interface is defined using XML layout files. These files describe the structure, layout, and appearance of the UI components, such as views, buttons, text fields, and more.

Activity Layout: This layout file defines the main UI for an Activity (a single screen in an Android app).

Layout Resources and Styles: Android provides various layout resources, such as dimensions, strings, colors, and styles, which can be defined in separate XML files and referenced throughout the app.

Styles allow developers to define reusable styles for UI components, promoting consistency and ease of maintenance.

Gradle scripts and XML layouts work together to build and define the structure and appearance of Android apps. Gradle scripts handle the build process, dependency management, and various tasks, while XML layouts define the UI components and their arrangement within the app's screens.

5.BACKEND (DATABASE)

Backend (Database) refers to the underlying system that stores and manages data for applications or websites. It is an essential component of any modern software system, as it provides a structured and organized way to store, retrieve, and manipulate data. The backend (database) is responsible for ensuring data integrity, consistency, and availability.

There are several types of database management systems (DBMS) available, each with its own strengths and weaknesses. The most common types are:

1. Relational Databases: These databases store data in tables with rows and columns, where each row represents a record, and each column represents a specific piece of information. Examples of relational databases include MySQL, PostgreSQL, Oracle, and Microsoft SQL Server.
2. NoSQL Databases: Unlike relational databases, NoSQL databases store data in a non-tabular way, such as key-value pairs, documents, or graphs. These databases are often used for handling large amounts of unstructured or semi-structured data. Examples include MongoDB, Cassandra, and Couchbase.
3. In-Memory Databases: As the name suggests, these databases store data entirely in memory, which allows for extremely fast read and write operations. However, they are typically more expensive and have limited storage capacity. Examples include Redis and Memcached.

When developing an application or website, developers interact with the backend (database) through a programming language or framework, such as PHP, Python, Ruby, or Node.js. These languages provide libraries or APIs that allow developers to execute database operations like creating, reading, updating, and deleting data.

The backend (database) plays a crucial role in ensuring data integrity and consistency. It enforces rules and constraints to prevent data corruption or inconsistencies. For example, it can ensure that duplicate records are not inserted, or that certain fields are mandatory and cannot be left blank.

In modern software development practices, databases are often integrated with other components of the application stack, such as web servers, caching layers, and message queues. This allows for efficient data flow, caching of frequently accessed data, and asynchronous processing of data-intensive operations.

In the context of Android development (or any app development), the backend typically refers to the server-side components of the application responsible for managing data and business logic. The backend database is where data is stored persistently.

5.1. Firebase

Firebase is a product of Google that helps developers build, manage, and grow their apps easily. It enables developers to build their apps faster and in a more secure way, without requiring any programming on the Firebase side. This makes it easy to use its features efficiently. Firebase provides services to Android, iOS, web, and Unity, and offers cloud storage. It uses NoSQL for the database, which is ideal for storing data.

Firebase initially started as an online chat service provider to various websites through API and was known as Envolve. However, developers used it to exchange application data, such as game states, in real-time across their users, which led to the separation of the Envolve architecture and its chat system. The Envolve architecture was further evolved by its founders, James Tamplin and Andrew Lee, to what modern-day Firebase is, in the year 2012.

Firebase provides its services in three main categories: Build better applications, Improve app quality, and Grow web and app project.
Under the "Build better applications" category, Firebase offers several services, including:
Realtime Database: A cloud-based NoSQL database that manages data at blazing speeds, similar to a big JSON file.
Authentication: A service that provides easy-to-use UI libraries and SDKs to authenticate users to an app, reducing the effort required to develop and maintain user authentication services.

5.2. History of Firebase

- Firebase, a product of Google, has a rich history that began in 2011 as a real-time data synchronization platform called Envolve. Envolve was an online chat service provider to various websites through API, but developers found it useful for exchanging application data, such as game states, in real-time across their users. This led to the separation of the Envolve architecture and its chat system, which was further evolved by its founders, James Tamplin and Andrew Lee, into what modern-day Firebase is today.
- Firebase is a Backend-as-a-Service (BaaS) platform that provides developers with a set of tools and infrastructure to build, develop, and grow their apps. It offers a real-time database, authentication services, cloud messaging, storage, and hosting, among other features. Firebase is designed to help developers build high-quality apps quickly and efficiently, with minimal backend development experience required.
- The platform is built around several key features, including the Realtime Database, Cloud Firestore, Authentication, Remote Config, Cloud Functions, Firebase Storage, Firebase Cloud Messaging (FCM), and Firebase Hosting. These features are designed to work together seamlessly, providing developers with a powerful set of tools to build and manage their apps.
- The Realtime Database is a cloud-hosted NoSQL database that allows developers to store and sync data in real-time. It uses a JSON-like data structure and provides offline support, making it an ideal choice for building collaborative and social apps. Cloud Firestore, on the other hand, is a flexible, scalable NoSQL cloud database that allows developers to store and sync data for client- and server-side development.
- Firebase Authentication provides developers with a simple way to authenticate users to their apps. It supports a variety of authentication methods, including email and password, phone number, and social media accounts. Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that allows developers to send messages and notifications to their users on multiple devices.

In conclusion, Firebase is a powerful and versatile platform that provides developers with a range of tools and services to build, improve, and grow their apps. With its real-time database, authentication services, cloud messaging, storage, and hosting, Firebase is an ideal choice for developers who want to build high-quality apps quickly and efficiently. Its growing community, range of tools and services, and accessible pricing make it a popular choice for developers of all skill levels.

5.3. Realtime Database with Java

Realtime databases play a crucial role in modern Android app development, enabling real-time data synchronization and updates across multiple clients. One of the most popular realtime database solutions for Android apps is Firebase Realtime Database, provided by Google. In this section, we'll explore how to work with Firebase Realtime Database in Android apps using Java.
1. Setting up Firebase Realtime Database:

Before you can start working with Firebase Realtime Database in your Android app, you need to set up a Firebase project and enable the Realtime Database service. This can be done through the Firebase Console, where you can create a new project, register your Android app, and download the necessary configuration files (google-services.json).

2. Adding Firebase to your Android project:

After setting up the Firebase project, you need to add the Firebase Android SDK to your app's dependencies. This can be done by adding the necessary dependencies in your app-level build.gradle file. For example, to include the Firebase Realtime Database library, you can add the following line:

```
implementation 'com.google.firebase:firebase-database:19.7.0'
```

Don't forget to sync your project with the Gradle files after making changes.

3. Initializing Firebase Realtime Database:

In your Android app's code, typically in the MainActivity or a separate class, you need to initialize the Firebase Realtime Database instance. This is usually done by obtaining a reference to the database using the `getInstance()` method from the `FirebaseDatabase` class.

```java
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("your_data_path");
```

The `getReference()` method allows you to specify the path where your data will be stored or retrieved from within the database.

4. Writing data to Firebase Realtime Database:

To write data to the Firebase Realtime Database, you can use the `setValue()` method on a `DatabaseReference` object. This method takes the data you want to write as an argument, which can be a Java object, a primitive value, or a collection.

```java
User user = new User("John Doe", "john@example.com");
myRef.child("users").child("user123").setValue(user);
```

In this example, we're creating a `User` object and writing it to the "users/user123" path in the database.

5. Reading data from Firebase Realtime Database:

To read data from the Firebase Realtime Database, you can use the `addValueEventListener()` method on a `DatabaseReference` object. This method takes a `ValueEventListener` as an argument, which will be called whenever the data at the specified path changes.

```java
myRef.child("users").child("user123").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // Get the User object from the DataSnapshot
        User user = dataSnapshot.getValue(User.class);
        // Update the UI with the retrieved data
    }
   @Override
    public void onCancelled(DatabaseError databaseError) {
        // Handle any errors that occurred
    }
});
```

In this example, we're attaching a `ValueEventListener` to the "users/user123" path. Whenever the data at this path changes, the `onDataChange()` method will be called with a `DataSnapshot` containing the updated data.

6. Realtime updates:

One of the key features of Firebase Realtime Database is its ability to provide real-time updates. When data changes in the database, all clients subscribed to that data location will automatically receive the updated data, thanks to the persistent WebSocket connection established between the client and the server.

7. Handling data conflicts:

In realtime scenarios, multiple clients may attempt to modify the same data simultaneously. Firebase Realtime Database handles data conflicts using operational transformations and conflict resolution strategies, ensuring data consistency and preventing data loss or corruption.

8. Security rules:

Firebase Realtime Database provides a set of security rules that allow you to control access to data. These rules are defined using a specific syntax and can be configured to grant or deny read and write permissions based on various conditions, such as user authentication, data structure, and data values.

9. Realtime Database listeners and lifecycle management:

When working with Firebase Realtime Database in Android apps, it's essential to manage the lifecycle of database listeners. Listeners should be detached or removed when the associated Activity or Fragment is no longer in use to prevent memory leaks and unnecessary data transfers. Android provides lifecycle methods like `onStart()`, `onStop()`, `onPause()`, and `onResume()` where you can attach and detach listeners accordingly.

10Data structure and modeling:

When working with a realtime database like Firebase Realtime Database, it's crucial to design an efficient and scalable data structure. Consider techniques like data flattening, data normalization, and denormalization to optimize data retrieval and storage.

12. Integration with other Firebase services:

Firebase Realtime Database can be seamlessly integrated with other Firebase services, such as Firebase Authentication for user authentication, Firebase Cloud Messaging for push

5.4. Object Oriented Programming Concept with Java

The concept of OOPs (Object-Oriented Programming) is a programming paradigm that revolves around the creation and manipulation of objects. It is a way of organizing and structuring code by bundling related data and functions together into objects. The main principles and concepts of OOPs are:

1. Classes and Objects: A class is a blueprint or template that defines the properties (data members) and behaviors (methods) of an object. An object is an instance of a class, created at runtime, with its own set of data and methods.

2. Encapsulation: Encapsulation is the process of binding data and functions together into a single unit (class). It helps in achieving data abstraction and hiding implementation details from the outside world, providing a layer of security and preventing unauthorized access to data.

3. Inheritance: Inheritance is a mechanism that allows a new class (derived class or subclass) to be based on an existing class (base class or superclass). The derived class inherits properties and methods from the base class, allowing code reuse and promoting the concept of hierarchical classification.

4. Polymorphism: Polymorphism refers to the ability of an object to take on many forms. It allows objects of different classes to be treated as objects of a common superclass. This is achieved through method overriding (redefining a method in the subclass with the same

signature as the superclass method) and method overloading (having multiple methods with the same name but different parameters).
5. Abstraction: Abstraction is the process of hiding unnecessary details and exposing only the essential features of an object or system. It helps in managing complexity by breaking down complex systems into simpler concepts and focusing on the essential aspects.
6. Association: Association is a relationship between two or more classes that enables them to communicate and interact with each other. It can be one-to-one, one-to-many, many-to-one, or many-to-many.
7. Aggregation: Aggregation is a special type of association where one class owns a collection of objects of another class. The owned objects have their own lifecycle and can exist independently of the owning class.
8. Composition: Composition is a stronger form of aggregation, where the owned objects are considered part of the owning object and cannot exist independently. If the owning object is destroyed, the owned objects are also destroyed.
9. Interfaces: An interface is a contract that defines a set of methods that a class must implement. It specifies what an object can do, but not how it does it. Interfaces promote abstraction and allow for multiple inheritance of methods (but not data).
10. Constructors and Destructors: Constructors are special methods used to initialize objects when they are created. Destructors are used to release resources and perform cleanup operations when an object is no longer needed and is about to be destroyed.
OOPs provides several benefits, including code reusability, modularity, encapsulation of data and code, and easier maintenance and debugging. It is widely used in various programming languages, such as Java, C++, Python, and C#, and is a fundamental concept in modern software development.

5.5. Collection and Framework (for Android)

Collections and frameworks play a crucial role in Android app development, providing reusable and efficient data structures and utilities that simplify common programming tasks. In the context of Android development using Java, here are some important collections and frameworks:

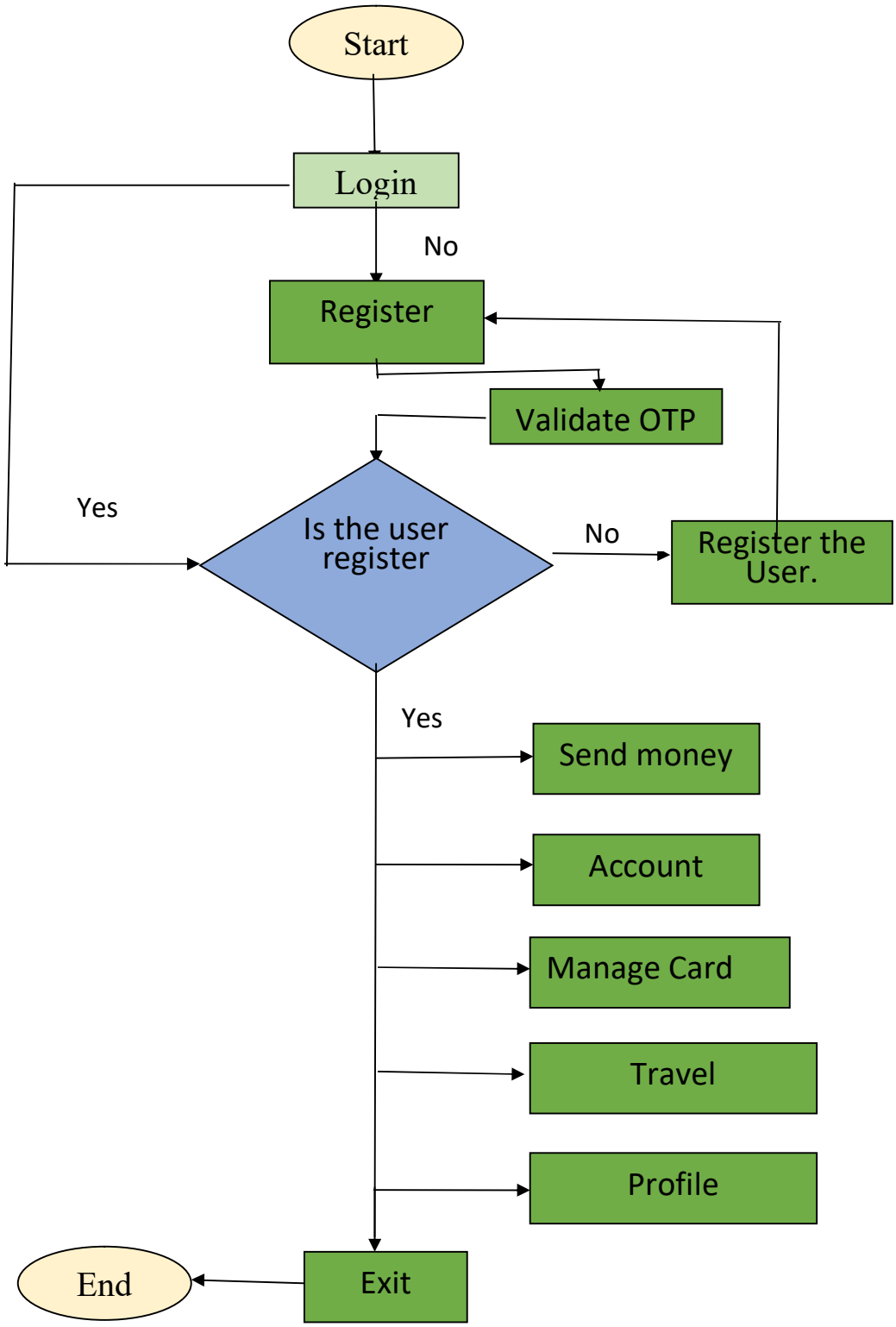
1. **Java Collections Framework (JCF)**: The Java Collections Framework is a unified architecture for representing and manipulating collections in Java. It provides several interfaces and classes for different types of collections, such as lists, sets, maps, and queues. In Android development, these collections are widely used for storing and manipulating data. Some commonly used classes and interfaces include:
    - `HashMap`: An implementation of the `Map` interface, which stores key-value pairs.
2. **Android Support Libraries**: Android Support Libraries are a set of libraries that provide backward-compatible versions of Android APIs and additional utilities. These libraries are essential for ensuring compatibility with older Android versions while still taking advantage of newer features. Some notable support libraries include:
    - `androidx.recyclerview`: Provides a more advanced and flexible version of the `ListView` for efficiently displaying large sets of data.
    - `androidx.appcompat`: Provides backward-compatible versions of Android APIs and UI components, enabling apps to have a consistent look and feel across different Android versions.
    - `androidx.fragment`: Enables modular app design by allowing the creation of reusable UI components called fragments.
3. **Android Architecture Components**: Android Architecture Components are a collection of libraries that help design robust, testable, and maintainable apps. These components follow the principles of modern app architecture, such as separating

concerns and promoting a reactive programming style. Some key components include:

- `ViewModel`: Stores and manages UI-related data, surviving configuration changes such as screen rotations.
- `LiveData`: An observable data holder class that follows the observer pattern and respects the Android lifecycle.
- `Room`: A persistence library that provides an abstraction layer over SQLite,

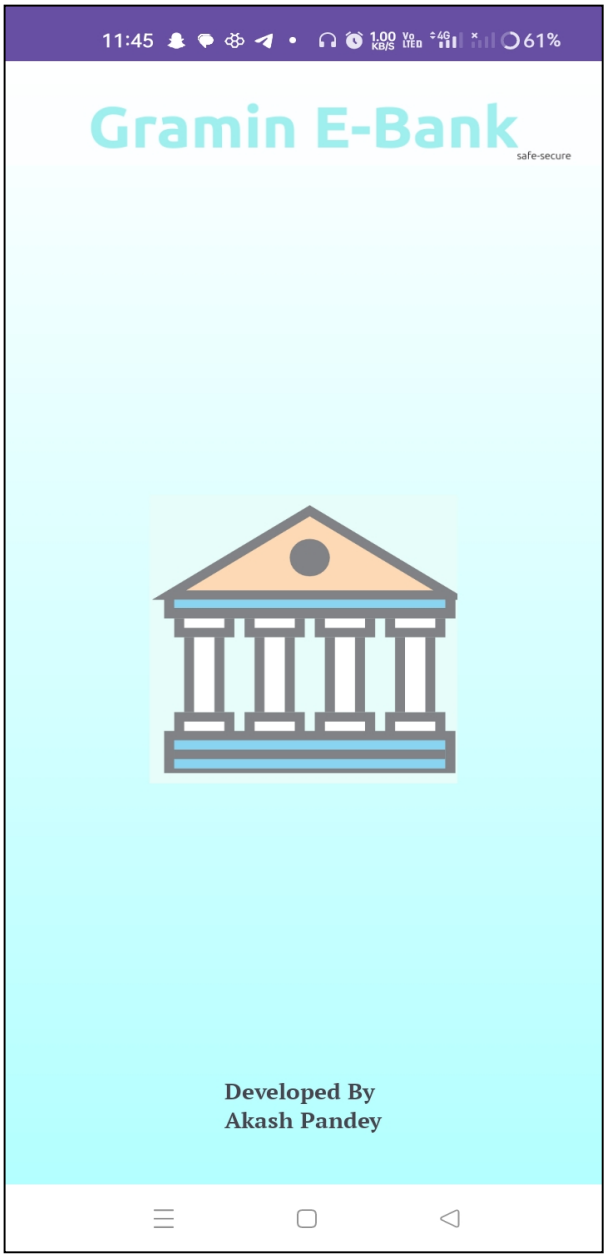
4. **Rx Java and Rx Android**: Rx Java is a popular library for implementing reactive programming in Java, while Rx Android provides a set of extensions for Android app development. These libraries enable developers to write asynchronous and event-based code in a more declarative and composable manner, simplifying the handling of asynchronous operations, such as network requests, database operations, and user input events.

6. FLOWCHART


Start
Login
No
Register
Validate OTP
Yes
Is the user register
No
Register the User.
Yes
Send money
Account
Manage Card
Travel
Profile
End
Exit

7.SCREENSHOT

7.1. Slash Activity



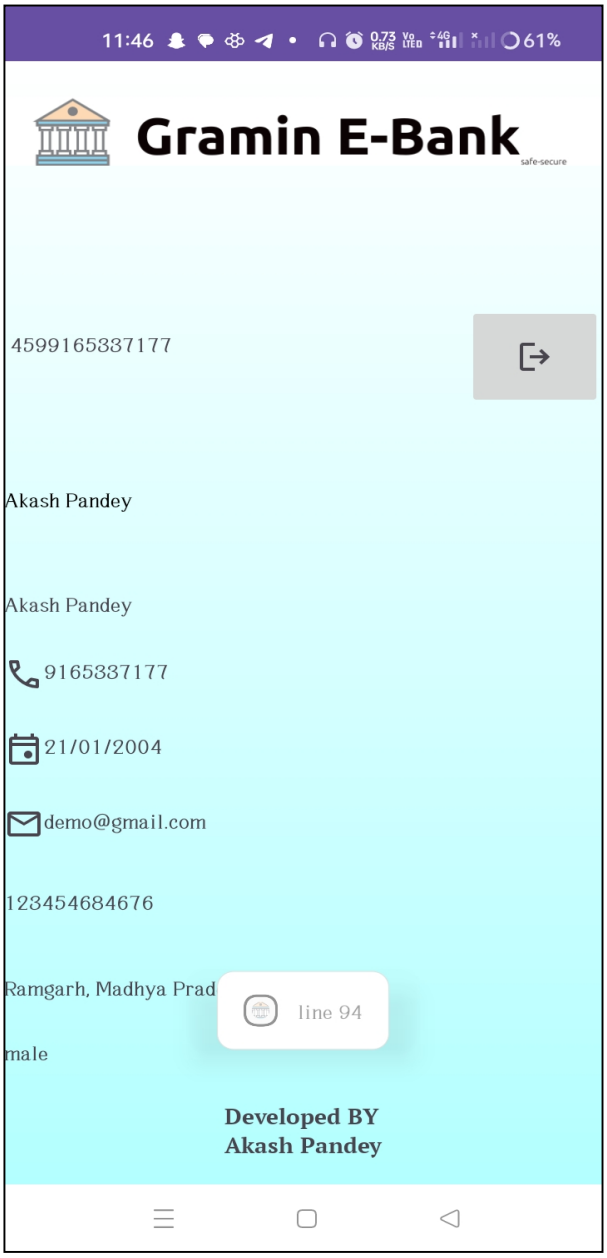
7.2. Main Activity

7.3. Account Activity

7.4. Card Activity

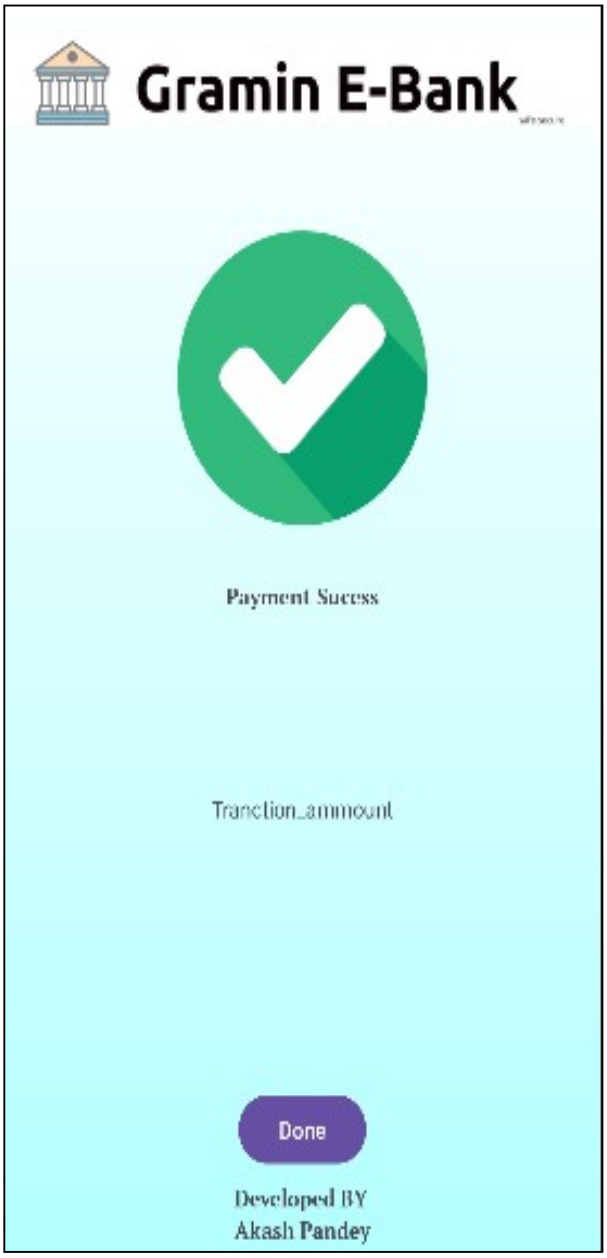
7.5 Travel Activity

7.6 Profile Activity

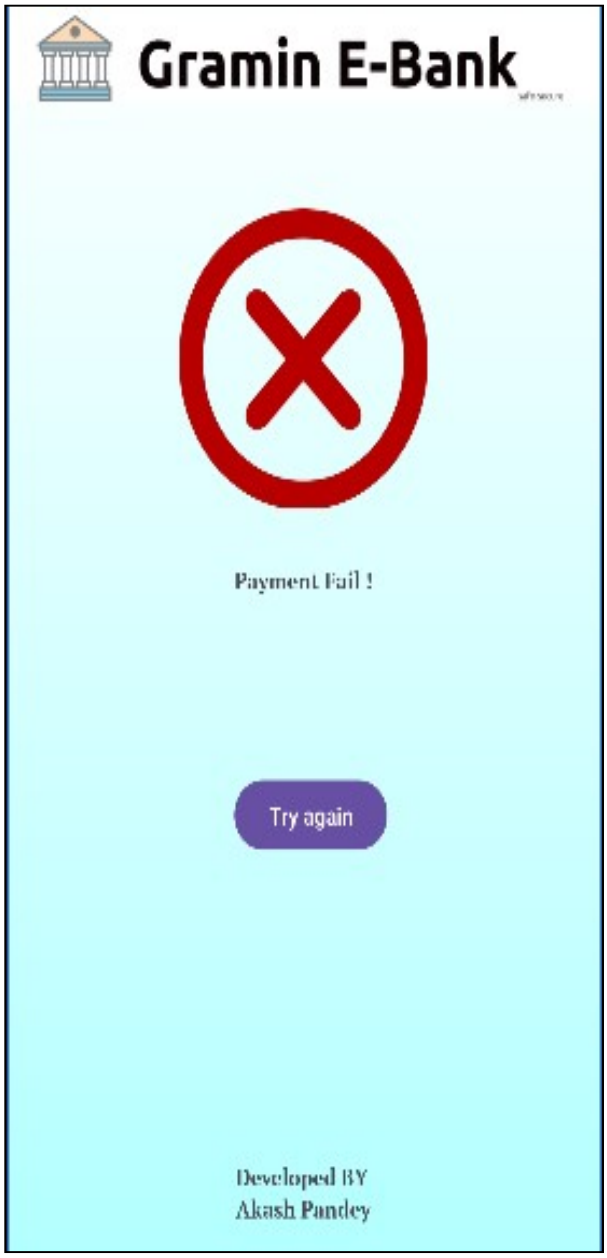


7.7 Send_money Activity

7.8 Payment_Sucess Activity

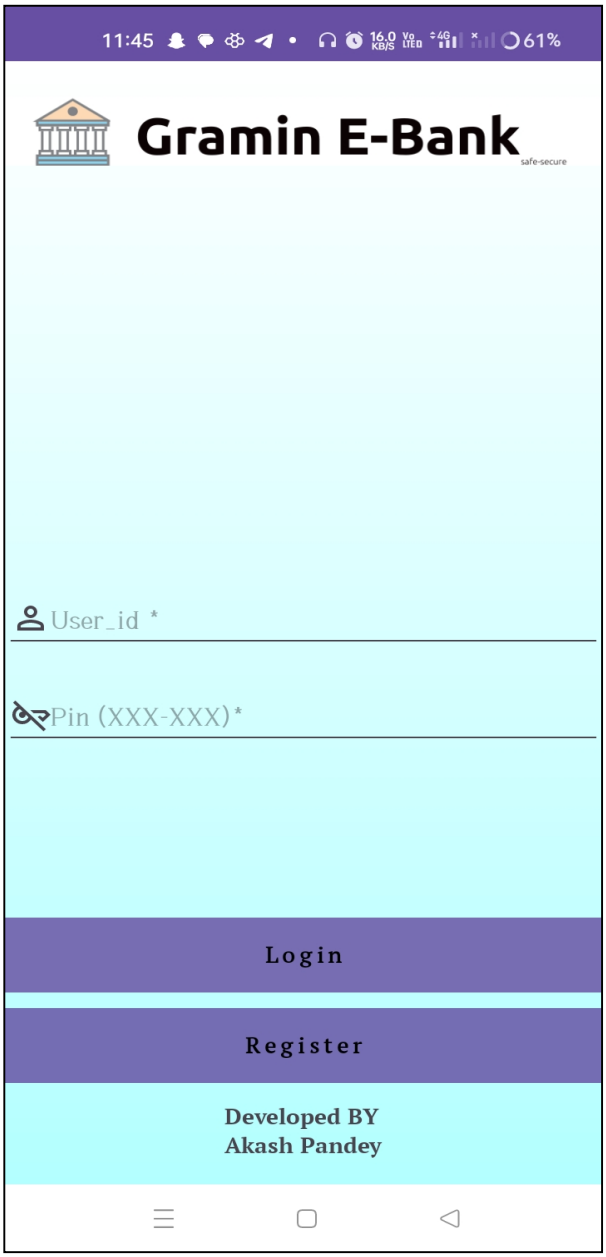


7.9 Payment_Fail Activity



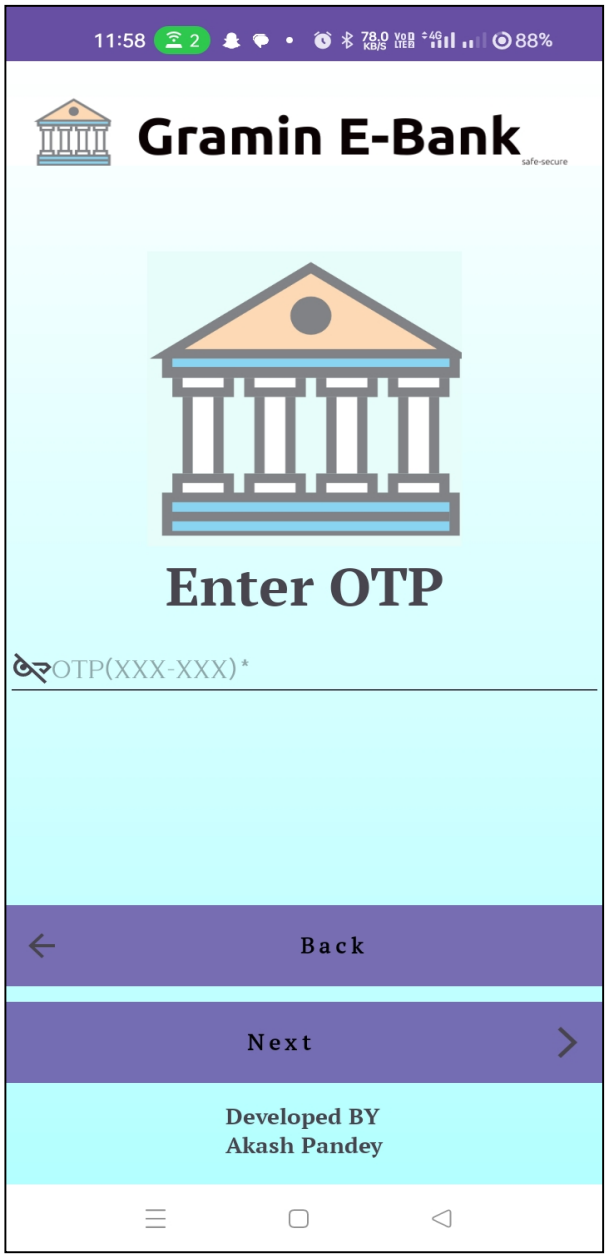
7.10. Login_register Activity

7.11. Login Activity

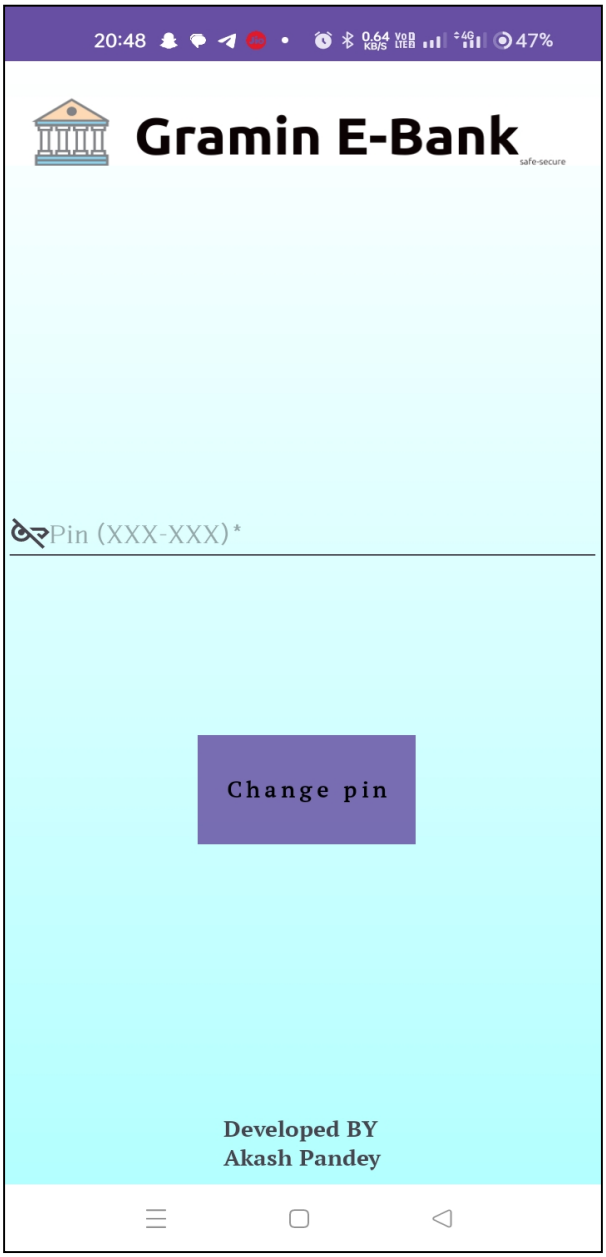


7.12. Register Activity

7.13. Validation Activity



7.14 Change_pin Activity



8. SOURCE CODE

- ***Account Activity (Java)***

```
package com.example.ebank;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
public class AccountActivity extends AppCompatActivity {
   Button Back;
   TextView number,bal,greetwithname;
   RecyclerView history;
   myadapter adapter;
   public DatabaseReference Userreference;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_account);
      Back=findViewById(R.id.Back);
      number=findViewById(R.id.number);
      bal=findViewById(R.id.bal);
      greetwithname=findViewById(R.id.greetwithname);
      Intent iNtnent = getIntent();
      String loggin_id = iNtnent.getStringExtra("log_id");
      FetchDataFromFirebase(loggin_id);
      Back.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Intent intent=new Intent(AccountActivity.this,MainActivity.class);
            startActivity(intent);
         } });
      history=(RecyclerView)findViewById(R.id.history);
      history.setLayoutManager(new LinearLayoutManager(this));
      FirebaseRecyclerOptions <model> options=
            new FirebaseRecyclerOptions.Builder<model>()
.setQuery(FirebaseDatabase.getInstance().getReference().child("User_detail").child(loggin_id).child("Transctio
n"),model.class)
                  .build();
      adapter=new myadapter(options);
      history.setAdapter(adapter);
```

```
    }
    @Override
    protected void onStart(){
        super.onStart();
        adapter.startListening();
  } @Override
    protected void onStop(){
        super.onStop();
        adapter.stopListening();
    }
   public void FetchDataFromFirebase(String loggin_id) {
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference().child("User_detail");
        reference.orderByChild("account_no");
        reference.addListenerForSingleValueEvent(new ValueEventListener() {
   @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                for (DataSnapshot userSnapshot : snapshot.getChildren()) {
                    String accountNo = userSnapshot.getKey();
                    if (accountNo.equals(loggin_id)) {
                         Toast.makeText(AccountActivity.this, "line 101", Toast.LENGTH_SHORT).show();
                        String name = userSnapshot.child("name").getValue(String.class);
                        String account_no = userSnapshot.child("account_no").getValue(String.class);
                        String balance = userSnapshot.child("balance").getValue(String.class);
                        float ac_bal= Float.parseFloat(balance);
                        greetwithname.setText(" Hello "+name);
                        number.setText(account_no);
                        bal.setText(" ₹ "+ac_bal);
                        break; // Exit the loop since we found the desired record
                    } }   }
        @Override
            public void onCancelled(@NonNull DatabaseError error) {
                // Getting data failed
                Toast.makeText(AccountActivity.this, "Failed to load data: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }  });   }}
```

- ***Account Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    android:textAlignment="center"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/imageView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <Button
        android:id="@+id/Back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:background="#D565C1E8"
        android:drawableLeft="@drawable/back"
```

```
    android:fontFamily="@font/regular"
    android:gravity="center"
    android:letterSpacing="0.2"
    android:padding="10dp"
    android:text="Back"
    android:textColor="#000000"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="4dp"
    android:layout_marginTop="100dp"
    android:drawableLeft="@drawable/account"
    android:text="Account Section"
    android:textSize="16dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/history"
    android:layout_width="319dp"
    android:layout_height="258dp"
    android:layout_marginStart="50dp"
    android:layout_marginTop="160dp"
    android:layout_marginEnd="50dp"
    android:layout_marginBottom="60dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.943" />
<TextView
    android:id="@+id/dispaly_history"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="272dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="24dp"
    android:fontFamily="monospace"
    android:text="History"
    android:textAlignment="center"
    android:textDirection="firstStrong"
    app:layout_constraintBottom_toTopOf="@+id/history"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView8" />
<TextView
    android:id="@+id/number"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="138dp"
    android:layout_marginTop="82dp"
    android:layout_marginEnd="22dp"
    android:fontFamily="@font/sbold"
    android:text="acc_number"
    app:layout_constraintEnd_toEndOf="parent"
```

```
        app:layout_constraintStart_toEndOf="@+id/Account_no"
        app:layout_constraintTop_toBottomOf="@+id/imageView8" />
    <TextView
        android:id="@+id/Account_no"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="22dp"
        android:layout_marginTop="48dp"
        android:fontFamily="@font/regular"
        android:text="Account_no"
        app:layout_constraintEnd_toStartOf="@+id/number"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView6" />
    <TextView
        android:id="@+id/bal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="138dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="22dp"
        android:fontFamily="@font/sbold"
        android:text="acc_bal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/Account_bal"
        app:layout_constraintTop_toBottomOf="@+id/number" />
    <TextView
        android:id="@+id/Account_bal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="22dp"
        android:layout_marginTop="50dp"
        android:fontFamily="@font/regular"
        android:text="Account_bal"
        app:layout_constraintEnd_toStartOf="@+id/bal"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Account_no" />
    <TextView
android:id="@+id/greetwithname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="45dp"
        android:fontFamily="sans-serif-medium"
        android:text="loading.....!"
        android:textAlignment="center"
        app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imageView8" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Card_data (Java)***

```
package com.example.ebank;
public class Card_data {
    public String  card_num;
   public String card_cvv;
    public String card_exp;
    // constructor
    public Card_data(String card_num, String card_cvv, String card_exp) {
        this.card_num = card_num;
        this.card_cvv = card_cvv;
        this.card_exp = card_exp;
    }
```

```
    public Card_data() {
    }
    //getter &  setter
    public String getCard_num() {
        return card_num;
    }
    public void setCard_num(String card_num) {
        this.card_num = card_num;
    }
    public String getCard_cvv() {
        return card_cvv;
    }
    public void setCard_cvv(String card_cvv) {
        this.card_cvv = card_cvv; }
    public String getCard_exp() {
        return card_exp;  }
    public void setCard_exp(String card_exp) {
        this.card_exp = card_exp;
    }}
```

- ***Card Activity (Java)***

```
package com.example.ebank;
importstatic com.google.android.material.color.utilities.MaterialDynamicColors.error;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.annotation.NonNull;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.ClosedSubscriberGroupInfo;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.time.LocalDate;
import java.time.Year;
import java.util.HashMap;
import kotlin.random.Random;
public class CardActivity extends AppCompatActivity {
    Button Back1,Change_pin;
    TextView card_number,card_cvv,card_exp,cardholder_name;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_card);
        //Recive current login user id from Login Activity
        Intent iNtent=getIntent();
        String login_id=iNtent.getStringExtra("log_id");
        //Find the id
        Back1=findViewById(R.id.Back1);
        Change_pin=findViewById(R.id.Change_Pin);
        card_number =findViewById(R.id.card_number);
        card_cvv=findViewById(R.id.card_cvv);
        card_exp=findViewById(R.id.card_exp);
        cardholder_name=findViewById(R.id.cardholder_name);
```

```
        Back1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(CardActivity.this,MainActivity.class);
                startActivity(intent);
            }});
Change_pin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(CardActivity.this, change_pin.class);
                intent.putExtra("login_id",login_id);
                startActivity(intent);
                finish();
            }  });
        FetchDataFromFirebase(login_id);
    }
    public void FetchDataFromFirebase(String login_id) {
        DatabaseReference reference =
FirebaseDatabase.getInstance().getReference().child("User_detail").child(login_id);
        reference.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                Toast.makeText(CardActivity.this, "75", Toast.LENGTH_SHORT).show();
                if (snapshot.exists()) {
                    // Toast.makeText(CardActivity.this, "line 104", Toast.LENGTH_SHORT).show();
                    String name=snapshot.child("name").getValue(String.class);
                    String cardNum = snapshot.child("card").child("card_num").getValue(String.class);
                    String cardCvv = snapshot.child("card").child("card_cvv").getValue(String.class);
                    String cardexp = snapshot.child("card").child("card_exp").getValue(String.class);
                    try {
                   //     Toast.makeText(CardActivity.this, "109", Toast.LENGTH_SHORT).show();
                        card_number.setText("4048 54"+cardNum);
                    //    Toast.makeText(CardActivity.this, "111", Toast.LENGTH_SHORT).show();
                        card_cvv.setText("CVV "+cardCvv);
                        card_exp.setText("MM/YY  "+"12/"+cardexp);
                        cardholder_name.setText(name);
                    } catch (Exception e) {
                        Toast.makeText(CardActivity.this, "No data load for display", Toast.LENGTH_SHORT).show();
                    }
                } else {
////                    // Data does not exist
                    Toast.makeText(CardActivity.this, "No data found", Toast.LENGTH_SHORT).show();
                }    }@Override
            public void onCancelled (@NonNull DatabaseError error){
                Toast.makeText(CardActivity.this, "Failed to Show Account detail Please try again later: " +
error.getMessage(), Toast.LENGTH_SHORT).show();
            }  }); }}
```

- ***Card Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".CardActivity">
    <ImageView
        android:id="@+id/imageView9"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <Button
        android:id="@+id/Back1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:background="#D565C1E8"
        android:drawableLeft="@drawable/back"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Back"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="4dp"
        android:layout_marginTop="120dp"
        android:drawableLeft="@drawable/card24px"
        android:text="Card Section"
        android:textSize="16dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/Change_Pin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="128dp"
        android:layout_marginBottom="4dp"
        android:text="Change Pin"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/Back1" />
    <RelativeLayout
        android:layout_width="363dp"
        android:layout_height="214dp"
        android:layout_marginTop="220dp"
        android:layout_marginBottom="152dp"
        android:background="@drawable/card_color"
        android:padding="16dp"
        app:layout_constraintBottom_toTopOf="@+id/Change_Pin"
        app:layout_constraintTop_toBottomOf="@+id/imageView9"
        app:layout_constraintVertical_bias="1.0">
    <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="8dp"
            android:elevation="8dp">
            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
```

```
android:background="@drawable/grdnt">
<ImageView
    android:id="@+id/card_type_icon"
    android:layout_width="match_parent"
    android:layout_height="32dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:src="@drawable/headerlogo" />
<TextView
    android:id="@+id/card_number"
    android:layout_width="294dp"
    android:layout_height="wrap_content"
    android:layout_alignTop="@id/card_type_icon"
    android:layout_marginLeft="-319dp"
    android:layout_marginTop="48dp"
    android:layout_toRightOf="@id/card_type_icon"
    android:text="1234 5678 9876 5432"
    android:textAlignment="center"
    android:textSize="16sp"
    android:textStyle="bold" />
<TextView
    android:id="@+id/cardholder_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/card_number"
    android:layout_marginLeft="-286dp"
    android:layout_marginTop="63dp"
    android:layout_toRightOf="@id/card_type_icon"
    android:text="CARDHOLDER"
    android:textSize="16sp" />
<TextView
    android:id="@+id/card_cvv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_marginTop="94dp"
    android:layout_marginRight="45dp"
    android:text="199"
    android:textSize="16sp"
    android:textStyle="bold" />
<TextView
    android:id="@+id/card_exp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_marginTop="93dp"
    android:layout_marginRight="189dp"
    android:text="12/99"
    android:textSize="16sp"
    android:textStyle="bold" />
<ImageView
    android:id="@+id/imageView17"
    android:layout_width="89dp"
    android:layout_height="37dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="2dp"
    android:layout_marginBottom="1dp"
```

```
            app:srcCompat="@drawable/visaaa" />
        </RelativeLayout>
    </androidx.cardview.widget.CardView>
  </RelativeLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Change_pin Activity (Java)***

```
package com.example.ebank;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import org.checkerframework.common.subtyping.qual.Bottom;
import java.util.HashMap;
public class change_pin extends AppCompatActivity {
EditText re_pin;
   Button Change_pin;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_change_pin);
      // find the all id
      Change_pin = findViewById(R.id.Change_pin);
      re_pin = findViewById(R.id.re_pin);
      Intent itent=getIntent();
      String loggin_id=itent.getStringExtra("login_id");
      Change_pin.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            final String new_pin = re_pin.getText().toString().trim();
            if (!new_pin.isEmpty()) {
               FirebaseDatabase database = FirebaseDatabase.getInstance();
               // Reference to the database node where data will be updated
               DatabaseReference myRef = database.getReference("User_detail").child(loggin_id);
               HashMap hashMap = new HashMap();
               hashMap.put("pin", new_pin);
               myRef.updateChildren(hashMap);
               Toast.makeText(change_pin.this, "Pin Change Sucessfully....!!",
Toast.LENGTH_SHORT).show();
               // Start back activity after a short delay
               new Handler().postDelayed(new Runnable() {
                  @Override
                  public void run() {
                     startActivity(new Intent(change_pin.this, CardActivity.class));
                     finish();
                  } }, 5000); // Delay for 5 seconds
```

```
                } else {
                    Toast.makeText(change_pin.this, "Fill all the field", Toast.LENGTH_SHORT).show();
                }}  });}}
```

- ***Change_pin Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".change_pin">
    <ImageView
        android:id="@+id/imageView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="16dp"
        tools:layout_editor_absoluteY="-32dp">
        <EditText
            android:id="@+id/re_pin"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_marginTop="280dp"
            android:drawableLeft="@drawable/password"
            android:ems="06"
            android:hint="Pin (XXX-XXX)*"
            android:inputType="numberPassword" />
<Button
            android:id="@+id/Change_pin"
            android:layout_width="182dp"
            android:layout_height="81dp"
            android:layout_below="@+id/re_pin"
            android:layout_alignParentStart="true"
            android:layout_alignParentEnd="true"
            android:layout_alignParentBottom="true"
            android:layout_marginStart="117dp"
            android:layout_marginTop="107dp"
            android:layout_marginEnd="112dp"
            android:layout_marginBottom="218dp"
            android:background="#D565C1E8"
            android:fontFamily="@font/regular"
            android:gravity="center"
            android:letterSpacing="0.2"
            android:padding="10dp"
            android:text="Change pin"
            android:textColor="#000000"
            android:textStyle="bold" />
    </RelativeLayout>
    <TextView
        android:id="@+id/textView4"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Failpayment Activity (Java)***

```
package com.example.ebank;
import android.os.Bundle;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
public class Failpayment extends AppCompatActivity {
  @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_failpayment);
        finish();
    }}
```

- ***Failpayment Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".Failpayment">
    <ImageView
        android:id="@+id/imageView14"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <TextView
        android:id="@+id/textView11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <ImageView
        android:id="@+id/imageView16"
        android:layout_width="176dp"
        android:layout_height="172dp"
        android:layout_marginTop="132dp"
        app:layout_constraintEnd_toEndOf="parent"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/failpay" />
    <TextView
        android:id="@+id/textView19"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:fontFamily="@font/sbold"
        android:text="Payment Fail !"
        app:layout_constraintEnd_toEndOf="@+id/imageView16"
        app:layout_constraintStart_toStartOf="@+id/imageView16"
        app:layout_constraintTop_toBottomOf="@+id/imageView16" />
    <Button
        android:id="@+id/Back5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Try again"
        app:layout_constraintEnd_toEndOf="@+id/textView19"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="@+id/textView19"
        tools:layout_editor_absoluteY="456dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Login Activity (Java)***

```
package com.example.ebank;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.content.SharedPreferences;
import com.example.ebank.*;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
public class Login extends AppCompatActivity {
    Button Register2, Login2;
    EditText User_id, User_pin2;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        User_id = findViewById(R.id.User_id);
        User_pin2 = findViewById(R.id.User_pin2);
        Register2 = findViewById(R.id.Register2);
        Register2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Login.this, Register.class);
                startActivity(intent); }});
       Login2 = findViewById(R.id.Login2);
        Login2.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View v) {
            final String Userid = User_id.getText().toString().trim();
            final String Userpin = User_pin2.getText().toString().trim();
            DatabaseReference Userreference =
FirebaseDatabase.getInstance().getReference().child("User_detail");
            Query checkUser_detail = Userreference.orderByChild("account_no").equalTo(Userid);
            checkUser_detail.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    if (snapshot.exists()) {
                        String passwordfromdb = snapshot.child(Userid).child("pin").getValue(String.class);
                        if (passwordfromdb.equals(Userpin)) {
                            Toast.makeText(Login.this, "Login Successfully",
Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent(Login.this, MainActivity.class);
                            //pass currtent login userid to another activity
                            intent.putExtra("Userid",Userid);
                            startActivity(intent);
                            finish();
                        } else {
                            Toast.makeText(Login.this,"Wrongpassword", Toast.LENGTH_SHORT).show();
                        }
                    } else {
                        Toast.makeText(Login.this, "User not found click to register ",
Toast.LENGTH_SHORT).show();
                    }
                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    Toast.makeText(Login.this, "Database Error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
                } });} });
    }
}
```

- ***Login Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".Login">
    <ImageView
        android:id="@+id/imageView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <EditText
        android:id="@+id/User_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="245dp"
        android:drawableLeft="@drawable/user"
```

```
        android:hint="User_id *"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView5" />
    <EditText
        android:id="@+id/User_pin2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="225dp"
        android:drawableLeft="@drawable/password"
        android:ems="06"
        android:hint="Pin (XXX-XXX)*"
        android:inputType="numberPassword"
        app:layout_constraintBottom_toTopOf="@+id/textView3" />
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <Button
        android:id="@+id/Login2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_marginBottom="10dp"
        android:background="#D565C1E8"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Login"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/Register2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />
    <Button
        android:id="@+id/Register2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_marginBottom="12dp"
        android:background="#D565C1E8"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Register"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/textView3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Loginregister Activity (Java)***

```
package com.example.ebank;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class Loginregister extends AppCompatActivity {
Button Login,Register;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_loginregister);
Login=findViewById(R.id.Login);
Login.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent=new Intent(Loginregister.this,Login.class);
startActivity(intent);
} });
Register=findViewById(R.id.Register);
Register.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent=new Intent(Loginregister.this,Register.class);
startActivity(intent);
} });    }}
```

- ***Loginregister Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@drawable/grdnt"
   tools:context=".Loginregister">
   <ImageView
      android:id="@+id/imageView3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:srcCompat="@drawable/headerlogo" />
   <ImageView
      android:id="@+id/imageView4"
      android:layout_width="194dp"
      android:layout_height="189dp"
      app:layout_constraintBottom_toTopOf="@+id/Login"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/imageView3"
      app:srcCompat="@drawable/bnkl" />
   <TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginBottom="16dp"
```

```
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <Button
        android:id="@+id/Login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_marginBottom="10dp"
        android:background="#D565C1E8"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Login"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/Register"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />
    <Button
        android:id="@+id/Register"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_marginBottom="12dp"
        android:background="#D565C1E8"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Register"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/textView2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Main Activity (Java)***

```
package com.example.ebank;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.example.ebank.Profile;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import android.app.AppComponentFactory;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
```

```
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    Button Send_money, Account, cards, Travel;
    ImageButton Profile;
    TextView name,morning_ev,ac_number,accountbal;
    public    DatabaseReference Userreference;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Recive current login user id from Login Activity
        Intent iNtent=getIntent();
        String log_id=iNtent.getStringExtra("Userid");
        name= findViewById(R.id.name);
        morning_ev = findViewById(R.id.morning_ev);
        ac_number = findViewById(R.id.ac_number);
        accountbal = findViewById(R.id.accountbal);
        //function for greating masseage
        greetingMessage();
        //fetch data from firebase
        FetchDataFromFirebase(log_id);
        Account = findViewById(R.id.Account);
        Account.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, AccountActivity.class);
                //pass currtent login userid to another activity
                intent.putExtra("log_id",log_id);
                startActivity(intent);
            }
        });
        cards = findViewById(R.id.cards);
        cards.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, CardActivity.class);
                //pass currtent login userid to another activity
                intent.putExtra("log_id",log_id);
                startActivity(intent);
            }
        });
        Travel = findViewById(R.id.Travel);
        Travel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, TravelActivity.class);
                //pass currtent login userid to another activity
                intent.putExtra("log_id",log_id);
                startActivity(intent);
            }
        });
        Profile = findViewById(R.id.Profile);
        Profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(com.example.ebank.MainActivity.this, "line 97",
Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(MainActivity.this, Profile.class);
                //pass currtent login userid to another activity
```

```
                intent.putExtra("log_id",log_id);
                startActivity(intent);
            }
        });
        Send_money = findViewById(R.id.Send_money);
        Send_money.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, SendmoneyActivity2.class);
                //pass currtent login userid to another activity
                intent.putExtra("log_id",log_id);
                startActivity(intent);
            }
        });
    }
    public void greetingMessage() {
        LocalDateTime currentDateTime = LocalDateTime.now();
        // Define date time formatter
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
        // Format and print current date and time
        String formattedDateTime = currentDateTime.format(formatter);
        String Transction_time = String.valueOf(formattedDateTime);
        System.out.println("Current Date and Time: " + Transction_time);
        int hour = currentDateTime.getHour();
        if (hour >= 5 && hour < 12) {
            //System.out.println("Good Morning!");
            morning_ev.setText("Good Morning!");
        } else if (hour >= 12 && hour < 17) {
            // System.out.println("Good Afternoon!");
            morning_ev.setText("Good Afternoon");
        } else {
            // System.out.println("Good Evening!");
            morning_ev.setText("Good Evening!");
        }  }
    public void FetchDataFromFirebase(String log_id ) {
        Userreference = FirebaseDatabase.getInstance().getReference().child("User_detail");
        Userreference.orderByChild("account_no");
        Userreference.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                for (DataSnapshot userSnapshot : snapshot.getChildren()) {
                    String accountNo = userSnapshot.getKey();
                    if (accountNo.equals(log_id))
                    {
                        String acc_number = userSnapshot.child("account_no").getValue(String.class);
                        String account_balance = userSnapshot.child("balance").getValue(String.class);
                        String account_name =
userSnapshot.child("account_holder_name").getValue(String.class);
                        float balance = Float.parseFloat(account_balance);
                        accountbal.setText("Account balance\n₹" + balance);
                        ac_number.setText( acc_number);
                        name.setText( account_name);
                        break; // Exit the loop since we found the desired record  }
            } }@Override
            public void onCancelled(@NonNull DatabaseError error) {
                // Getting data failed
                Toast.makeText(MainActivity.this, "Failed to Show Account detail Please try again later: " +
error.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
```

}}

- ***Main Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".MainActivity">
  <ImageView
        android:id="@+id/imageView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <ImageButton
        android:id="@+id/Profile"
       android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_marginTop="156dp"
        android:layout_marginEnd="359dp"
        android:textAlignment="center"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/user" />
    <Button
        android:id="@+id/home"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_marginStart="2dp"
        android:background="#D565C1E8"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Home"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/Account"
        app:layout_constraintStart_toStartOf="parent" />
    <Button
        android:id="@+id/Account"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:background="#D565C1E8"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Account"
        android:textColor="#000000"
        android:textStyle="bold"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/cards"
    app:layout_constraintStart_toEndOf="@+id/home" />
<Button
    android:id="@+id/cards"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:background="#D565C1E8"
    android:fontFamily="@font/regular"
    android:gravity="center"
    android:letterSpacing="0.2"
    android:padding="10dp"
    android:text="Manage Card"
    android:textColor="#000000"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/Travel"
    app:layout_constraintStart_toEndOf="@+id/Account" />
<Button
    android:id="@+id/Travel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:background="#D565C1E8"
    android:fontFamily="@font/regular"
    android:gravity="center"
    android:letterSpacing="0.2"
    android:padding="10dp"
    android:text="Travel"
    android:textStyle="bold"
    android:textColor="#000000"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/cards" />
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/sbold"
    android:text="Welcome to Gramin E-Bank "
    android:textColor="#9057F4"
    android:textSize="25dp"
    app:autoSizeMaxTextSize="25dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView7"
    tools:ignore="MissingConstraints" />
<Button
    android:id="@+id/Send_money"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Send Money"
    app:layout_constraintBottom_toTopOf="@+id/home"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Profile"
    app:layout_constraintVertical_bias="0.835" />
<TextView
    android:id="@+id/morning_ev"
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="32dp"
        android:fontFamily="sans-serif-light"
        android:text="TextView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Profile" />
    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:fontFamily="sans-serif-thin"
        android:text="loading_name"
        app:layout_constraintTop_toBottomOf="@+id/morning_ev"
        tools:layout_editor_absoluteX="31dp" />
    <TextView
        android:id="@+id/accountn_o"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="56dp"
        android:fontFamily="sans-serif-thin"
        android:text="Account no:"
        app:layout_constraintTop_toBottomOf="@+id/name"
        tools:layout_editor_absoluteX="32dp" />
    <TextView
        android:id="@+id/accountbal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:fontFamily="sans-serif-thin"
        android:text="balance"
        app:layout_constraintTop_toBottomOf="@+id/accountn_o"
        tools:layout_editor_absoluteX="50dp" />
    <TextView
        android:id="@+id/textView13"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="307dp"
        android:text="IFSC: GEBX00098767"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/accountbal"
        app:layout_constraintTop_toBottomOf="@+id/textView5" />
    <TextView
        android:id="@+id/ac_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="248dp"
        android:text="TextView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/accountn_o"
        app:layout_constraintTop_toBottomOf="@+id/textView5" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Model class (java)***

```
package com.example.ebank;
public class model {
    String Transction_time,Transction_to, Transction_from, Transction_amount,Transction_id,Transction_idd;
    //Constructor Empty
    public model() {
    }
```

```
    //Constructor
    public model(String transction_time, String transction_to, String transction_from, String transction_amount,
String transction_id) {
        Transction_time = transction_time;
        Transction_to = transction_to;
        Transction_from = transction_from;
        Transction_amount = transction_amount;
        Transction_id = transction_id;
        Transction_idd=Transction_idd;
    }
    //Getter setter
    public String getTransction_time() {
        return Transction_time;
    }
    public String getTransction_idd() {
        return Transction_idd;
    }
  public void setTransction_idd(String transction_idd) {
        Transction_idd = transction_idd;   }
   public void setTransction_time(String transction_time) {
        Transction_time = transction_time;}
    public String getTransction_to() {
        return Transction_to;
    }
    public void setTransction_to(String transction_to) {
        Transction_to = transction_to;
    }
    public String getTransction_from() {
        return Transction_from;
    }
    public void setTransction_from(String transction_from) {
        Transction_from = transction_from;
    }
    public String getTransction_amount() {
        return Transction_amount;
    }
    public void setTransction_amount(String transction_amount) {Transction_amount = transction_amount;}
    public String getTransction_id() {
        return Transction_id;
    }
    public void setTransction_id(String transction_id) { Transction_id = transction_id;
    }}
```

- ***Myadapter class (Java)***

```
package com.example.ebank;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
//import com.bumptech.glide.Glide;
```

```
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
public class myadapter extends FirebaseRecyclerAdapter<model,myadapter.myviewholder> {
    public myadapter(@NonNull FirebaseRecyclerOptions<model> options) {
        super(options); }@Override
    protected void onBindViewHolder(@NonNull myviewholder holder, int position, @NonNull model model) {
        holder.Trans_id.setText("Transction ID :"+model.getTransction_id());
        holder.Trans_time.setText("Transction Time :"+model.getTransction_time());
        holder.Trans_to.setText("Transction To :"+model.getTransction_to());
        holder.Trans_from.setText("Transction From :"+model.getTransction_from());
        holder.Trans_ammount.setText("Transction Ammount :"+model.getTransction_amount());
        holder.Trans_idd.setText(model.getTransction_idd());
    }
    @NonNull
    @Override
    public myviewholder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view= LayoutInflater.from(parent.getContext()).inflate(R.layout.singlerow,parent,false);
        return new myviewholder(view) ;
    }
    class myviewholder extends RecyclerView.ViewHolder{
        TextView Trans_id,Trans_time,Trans_to,Trans_from,Trans_ammount,Trans_idd;
        public myviewholder(@NonNull View itemView) {
            super(itemView);
            Trans_id=(TextView)itemView.findViewById(R.id.Trans_id);
            Trans_idd=(TextView)itemView.findViewById(R.id.Trans_idd);
            Trans_time=(TextView)itemView.findViewById(R.id.Trans_time);
            Trans_to=(TextView)itemView.findViewById(R.id.Trans_to);
            Trans_from=(TextView)itemView.findViewById(R.id.Trans_from);
            Trans_ammount=(TextView)itemView.findViewById(R.id.Trans_ammount);
        }}}
```

- ***Profile Activity (Java)***

```
package com.example.ebank;
import android.app.AppComponentFactory;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;
import android.content.SharedPreferences;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.example.ebank.Loginregister;
import com.example.ebank.Register;
import com.example.ebank.Login;
import com.example.ebank.*;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
```

```
public class Profile extends AppCompatActivity {
   ImageButton logout;
   TextView User_name1, User_fathername1, User_phone_no1, User_email1, User_dob1, User_address1,
User_gender1, User_addhar1, account_no1;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_profile);
       //Recive current login user id from Login Activity
       Intent iNtent=getIntent();
       String login_id=iNtent.getStringExtra("log_id");
       // Finding the views
       User_name1 = findViewById(R.id.User_name1);
       User_fathername1 = findViewById(R.id.User_Fathername1);
       User_dob1 = findViewById(R.id.User_dob1);
       User_phone_no1 = findViewById(R.id.User_phone_no1);
       account_no1 = findViewById(R.id.account_no1);
       User_email1 = findViewById(R.id.User_email1);
       User_addhar1 = findViewById(R.id.User_addhar1);
       User_gender1 = findViewById(R.id.User_gender1);
       User_address1 = findViewById(R.id.User_address1);
       // Function for fetching data from db
       FetchDataFromFirebase(login_id);
       //  Logout button
       logout = findViewById(R.id.logout);
       logout.setOnClickListener(new View.OnClickListener() {
             @Override
               public void onClick(View v) {
             Intent intent = new Intent(Profile.this, Loginregister.class);
             startActivity(intent);
                finish();
             Toast.makeText(Profile.this, "Logout ", Toast.LENGTH_SHORT).show();
            finish();  }});  }
   public void FetchDataFromFirebase(String login_id) {
  DatabaseReference reference = FirebaseDatabase.getInstance().getReference().child("User_detail");
        reference.orderByChild("account_no");
        reference.addListenerForSingleValueEvent(new ValueEventListener() {
           @Override
           public void onDataChange(@NonNull DataSnapshot snapshot) {
               Toast.makeText(Profile.this, "line 94", Toast.LENGTH_SHORT).show();
               for (DataSnapshot userSnapshot : snapshot.getChildren()) {
                   String accountNo = userSnapshot.getKey();
                   if (accountNo.equals(login_id)) {
                       String name = userSnapshot.child("name").getValue(String.class);
                       String fathername = userSnapshot.child("father_name").getValue(String.class);
                       String mobile_no = userSnapshot.child("phone_no").getValue(String.class);
                       String email = userSnapshot.child("email").getValue(String.class);
                       String account_no = userSnapshot.child("account_no").getValue(String.class);
                       String dob = userSnapshot.child("dob").getValue(String.class);
                       String address = userSnapshot.child("address").getValue(String.class);
                       String gender = userSnapshot.child("gender").getValue(String.class);
                       String addhar = userSnapshot.child("addhar").getValue(String.class);
                       User_name1.setText(name);
                       Toast.makeText(Profile.this, "line 143", Toast.LENGTH_SHORT).show();
                       User_fathername1.setText(fathername);
                       User_dob1.setText(dob);
                       User_phone_no1.setText(mobile_no);
                       account_no1.setText(account_no);
                       User_email1.setText(email);
                       User_addhar1.setText(addhar);
```

```
                User_gender1.setText(gender);
                User_address1.setText(address);
                break; // Exit the loop since we found the desired record
            }} }   @Override
        public void onCancelled(@NonNull DatabaseError error) {
            // Getting data failed
            Toast.makeText(Profile.this, "Failed to load data: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
        }});}}
```

- ***Profile Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".Register">
    <ImageView
        android:id="@+id/imageView13"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
<ImageButton
        android:id="@+id/logout"
        android:layout_width="82dp"
        android:layout_height="67dp"
        android:layout_marginTop="156dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/logout" />
    <TextView
        android:id="@+id/User_name1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:text=""
        android:textColor="@color/black"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/logout" />
    <TextView
        android:id="@+id/User_Fathername1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:text=""
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/User_name1" />
    <TextView
        android:id="@+id/User_phone_no1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
    android:layout_marginTop="24dp"
    android:drawableLeft="@drawable/phone"
    android:text=""
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/User_Fathername1" />
<TextView
    android:id="@+id/User_email1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:drawableLeft="@drawable/mail"
    android:text=""
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/User_dob1" />
<TextView
    android:id="@+id/User_dob1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:drawableLeft="@drawable/dob"
    android:text=""
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/User_phone_no1" />
<TextView
    android:id="@+id/User_address1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:text=""
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/User_addhar1" />
<TextView
    android:id="@+id/User_gender1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text=""
    app:layout_constraintBottom_toTopOf="@+id/textView10"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/User_address1" />
<TextView
    android:id="@+id/User_addhar1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:text=""
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/User_email1" />
<TextView
    android:id="@+id/textView10"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/account_no1"
        android:layout_width="263dp"
        android:layout_height="42dp"
        android:text="account_no"
        app:layout_constraintBottom_toTopOf="@+id/User_name1"
        app:layout_constraintEnd_toStartOf="@+id/logout"
        app:layout_constraintHorizontal_bias="0.242"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView13"
        app:layout_constraintVertical_bias="0.586" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Register Activity (Java)***

```
package com.example.ebank;
import static android.app.ProgressDialog.show;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.hbb20.CountryCodePicker;
import java.lang.ref.Cleaner;
import java.time.LocalDate;
import java.util.Random;
public class Register extends AppCompatActivity {
Button Login3,Register3;
CountryCodePicker ccp;
private EditText
User_name,User_fathername,User_phone_no,User_pin,User_email,User_dob,User_address,User_gender,User_
addhar;
public    DatabaseReference Userreference;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_register);
// Find id of Edittext field
User_name=findViewById(R.id.User_name);
User_fathername=findViewById(R.id.User_Fathername);
User_dob=findViewById(R.id.User_dob);
User_phone_no=findViewById(R.id.User_phone_no);
User_pin=findViewById(R.id.User_pin);
User_email=findViewById(R.id.User_email);
User_addhar=findViewById(R.id.User_addhar);
User_gender=findViewById(R.id.User_gender);
User_address=findViewById(R.id.User_address);
ccp=(CountryCodePicker)findViewById(R.id.ccp);
ccp.registerCarrierNumberEditText(User_phone_no);
Login3=findViewById(R.id.Login3);
```

```
Login3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent=new Intent(Register.this,Login.class);
startActivity(intent);
}
});
Register3=findViewById(R.id.Register3);
Register3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();
Userreference = FirebaseDatabase.getInstance().getReference().child("User_detail");
String account_holder_name = User_name.getText().toString().trim();
String father_name  = User_fathername.getText().toString().trim();
String dob = User_dob.getText().toString().trim();
String phone_no =User_phone_no.getText().toString().trim();
String pin =  User_pin.getText().toString().trim();
String email = User_email.getText().toString().trim();
String address = User_address.getText().toString().trim();
String addhar = User_addhar.getText().toString().trim();
String gender= User_gender.getText().toString().trim();
String  account_no= ("459"+phone_no).toString().trim();
String balance= ("5000").toString().trim();
if (!account_holder_name.isEmpty() && !father_name.isEmpty()  && !dob.isEmpty()  &&
!phone_no.isEmpty() && !pin.isEmpty()  && !email.isEmpty()  && !address.isEmpty()  && !addhar.isEmpty()
&& !gender.isEmpty())
{
// Create a new User object
Userdata data=new
Userdata(account_holder_name,account_no,balance,father_name,dob,phone_no,pin,email,address,addhar,gende
r); // Assuming User is your data model class
// Push the data to Firebase Realtime Database
Userreference.child(account_no).setValue(data);
// genrating cards value and number
// firebase database
DatabaseReference reference =
FirebaseDatabase.getInstance().getReference().child("User_detail").child(account_no).child("card");
//Transction id genrator using random function
Random random = new Random();
int randomNumber6 = 100000 + random.nextInt(900000);
int randomNumber3 = 100 + random.nextInt(900);
//System.out.println("Random 6-digit number: " + card_number);
String card_number= String.valueOf(randomNumber6);
String card_cvv= String.valueOf(randomNumber3);
LocalDate currentDate = LocalDate.now();
int currentYear = currentDate.getYear();
int card_exp = currentYear+5;
String card_expp= String.valueOf(card_exp);
Card_data carddata=new Card_data(card_number,card_cvv,card_expp);
// Push the data to Firebase Realtime Database
//reference.orderByChild("account_no");
reference.setValue(carddata);
Toast.makeText(Register.this, "Data Loaded in Firebase DB", Toast.LENGTH_SHORT).show();
//Switch to another activity
Intent intent=new Intent(Register.this,ValidationActivity.class);
intent.putExtra("mobile",ccp.getFullNumberWithPlus().replace(" ",""));
startActivity(intent);} else {
Toast.makeText(Register.this, "Please fill all fields", Toast.LENGTH_SHORT).show();
}}});}}
```

- ***Main Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".Register">
    <ImageView
        android:id="@+id/imageView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <com.hbb20.CountryCodePicker
        android:id="@+id/ccp"
        android:layout_width="131dp"
        android:layout_height="48dp"
        android:layout_marginStart="2dp"
        android:layout_marginBottom="8dp"
        app:layout_constraintBottom_toTopOf="@+id/User_pin"
        app:layout_constraintEnd_toStartOf="@+id/User_phone_no"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent" />
    <EditText
        android:id="@+id/User_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="15dp"
        android:hint="Name"
        app:layout_constraintBottom_toTopOf="@+id/User_Fathername"
        tools:layout_editor_absoluteX="0dp" />
    <EditText
        android:id="@+id/User_Fathername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="12dp"
        android:hint="Father name"
        app:layout_constraintBottom_toTopOf="@+id/User_phone_no"
        tools:layout_editor_absoluteX="123dp" />
    <EditText
        android:id="@+id/User_phone_no"
        android:layout_width="264dp"
        android:layout_height="49dp"
        android:layout_marginBottom="4dp"
        android:drawableLeft="@drawable/phone"
        android:hint="Phone (XXX-XXX-XXXX)"
        android:inputType="phone"
        app:layout_constraintBottom_toTopOf="@+id/User_pin"
        app:layout_constraintEnd_toEndOf="parent" />
    <EditText
        android:id="@+id/User_pin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="15dp"
        android:drawableLeft="@drawable/password"
```

```
    android:ems="06"
    android:hint="Pin(XXX-XXX)"
    android:inputType="numberPassword"
    app:layout_constraintBottom_toTopOf="@+id/User_email" />
<EditText
    android:id="@+id/User_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="15dp"
    android:drawableLeft="@drawable/mail"
    android:hint="Email"
    android:inputType="textEmailAddress"
    app:layout_constraintBottom_toTopOf="@+id/User_dob"
    tools:layout_editor_absoluteX="-16dp" />
<EditText
    android:id="@+id/User_dob"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="15dp"
    android:drawableLeft="@drawable/dob"
    android:hint="Date of Birth (MM/DD/YYYY)"
    android:inputType="date"
    app:layout_constraintBottom_toTopOf="@+id/User_addhar" />
<EditText
    android:id="@+id/User_address"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:hint="Address"
    app:layout_constraintTop_toBottomOf="@+id/User_addhar"
    tools:layout_editor_absoluteX="16dp" />
<EditText
    android:id="@+id/User_gender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="gender"
    app:layout_constraintBottom_toTopOf="@+id/textView4"
    app:layout_constraintTop_toBottomOf="@+id/User_address"
    tools:layout_editor_absoluteX="16dp" />
<EditText
    android:id="@+id/User_addhar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="135dp"
    android:hint="Addhar no."
    android:inputType="number"
    app:layout_constraintBottom_toTopOf="@+id/textView4"
    tools:layout_editor_absoluteX="-36dp" />
<Button
    android:id="@+id/Register3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginBottom="16dp"
    android:drawableLeft="@drawable/register"
    android:fontFamily="@font/regular"
    android:text="Register"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/textView4" />
```

```
    <Button
        android:id="@+id/Login3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="16dp"
        android:drawableLeft="@drawable/login"
        android:fontFamily="@font/regular"
        android:text="Login"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/textView4"
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Sendmoney Activity (Java)***

```
package com.example.ebank;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
public class SendmoneyActivity2 extends AppCompatActivity {
    TextView account_number, acc_balance;
    Button Check, Pay;
    EditText Reciver_account_no, Re_Reciver_account_no, User_pin, Amount;
    public DatabaseReference Userreference;
   @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sendmoney2);
        //Recive current login user id from Login Activity
        Intent iNtent = getIntent();
```

```
        String login_id = iNtent.getStringExtra("log_id");
        //Function for database
        FetchDataFromFirebase(login_id);
        // Finding the views
        Pay = findViewById(R.id.Pay);
        account_number = findViewById(R.id.account_number);
        acc_balance = findViewById(R.id.acc_balance);
        Reciver_account_no = findViewById(R.id.Reciver_account_no);
        Re_Reciver_account_no = findViewById(R.id.Re_Reciver_account_no);
        User_pin = findViewById(R.id.User_pin);
        Amount = findViewById(R.id.Amount);
        Pay.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //string
                final String reaccount_number = Reciver_account_no.getText().toString().trim();
                final String re_account_number = Re_Reciver_account_no.getText().toString().trim();
                final String send_ammount = Amount.getText().toString().trim();
                final String m_pin = User_pin.getText().toString().trim();
                Toast.makeText(SendmoneyActivity2.this, "64", Toast.LENGTH_SHORT).show();
                //if for fill all field
               if (!reaccount_number.isEmpty() && !re_account_number.isEmpty() && !send_ammount.isEmpty()
&& !m_pin.isEmpty()) {
                    //macting account number to re enter acccount number
                    if (reaccount_number.equalsIgnoreCase(re_account_number)) {
                        // Get a reference to the Firebase Realtime Database
                        //Create a refrence of Database
                        DatabaseReference reference =
FirebaseDatabase.getInstance().getReference().child("User_detail");
                        // Toast.makeText(Profile.this, "line 88", Toast.LENGTH_SHORT).show();
                        reference.child("account_no");
                        reference.addListenerForSingleValueEvent(new ValueEventListener() {
                            @Override
                            public void onDataChange(@NonNull DataSnapshot snapshot) {
                                Toast.makeText(SendmoneyActivity2.this, "line 90", Toast.LENGTH_SHORT).show();
                                if (snapshot.hasChild(re_account_number)) {
                                    Toast.makeText(SendmoneyActivity2.this, "data exist ",
Toast.LENGTH_SHORT).show();
                                    String passwordfromdb = snapshot.child(login_id).child("pin").getValue(String.class);
                                    Toast.makeText(SendmoneyActivity2.this, "79", Toast.LENGTH_SHORT).show();
                                    if (passwordfromdb.equals(m_pin)) {
                                        Toast.makeText(SendmoneyActivity2.this, "82", Toast.LENGTH_SHORT).show();
                                        String Fundfromdb = snapshot.child(login_id).child("balance").getValue(String.class);
                                        String Fundfromdb_re =
snapshot.child(reaccount_number).child("balance").getValue(String.class);
                                        float fund = Float.valueOf(Fundfromdb);
                                        float fund_re = Float.valueOf(Fundfromdb_re);
                                        float send_fund = Float.parseFloat(send_ammount);
                                        if (fund >= send_fund) {
                                            int send_bal = Integer.parseInt(send_ammount);
                                            int se_cu_bal = (int) (fund - send_bal);
                                            int re_cu_bal = (int) (fund_re + send_bal);
                                            String re_Cu_balance = String.valueOf(re_cu_bal);
                                            String se_Cu_balance = String.valueOf(se_cu_bal);
                                            Toast.makeText(SendmoneyActivity2.this, "Hold a minute and don't quit the app",
Toast.LENGTH_SHORT).show();
                                        //update the balance in a/c of reciver and sender
                                        //HashMap updatedata = new HashMap();
                                        //  updatedata.put("balance", se_Cu_balance);
```

```
                                Toast.makeText(SendmoneyActivity2.this, "Please Wait it take some time ",
Toast.LENGTH_SHORT).show();
                                updateData(login_id, reaccount_number, re_Cu_balance, se_Cu_balance );
                                usertransaction(login_id, reaccount_number, send_ammount); //call after getting the
refrence of db otherwise pass the refrence
                                String re_Fundfromdb =
snapshot.child(login_id).child("balance").getValue(String.class);
                                String re_Fundfromdb_re =
snapshot.child(reaccount_number).child("balance").getValue(String.class);
                              // here we call sucess failure activity
                                if (!re_Fundfromdb.equalsIgnoreCase(Fundfromdb) &&
!re_Fundfromdb_re.equalsIgnoreCase(Fundfromdb_re)) {
                                   Toast.makeText(SendmoneyActivity2.this, "Payment Fail",
Toast.LENGTH_SHORT).show();
                                  // stsatr activity
                                  new Handler().postDelayed(new Runnable() {
                                     @Override
                                     public void run() {
                                        startActivity(new Intent(SendmoneyActivity2.this, Failpayment.class));
finish();
                                     }}, 1000); }
                              else{// stsatr activity
                                  Intent intEnt = new Intent(SendmoneyActivity2.this, Sucess_payment.class);
                                  //pass currtent login userid to another activity
                                 intEnt.putExtra("send_ammount",send_ammount);
                                  startActivity(intEnt);
                                }          } else {Toast.makeText(SendmoneyActivity2.this, "Iffucient Fund",
Toast.LENGTH_SHORT).show();}} else {
                             Toast.makeText(SendmoneyActivity2.this, "Incorrect Pin",
Toast.LENGTH_SHORT).show();}
 } else {Toast.makeText(SendmoneyActivity2.this, "User not exist ", Toast.LENGTH_SHORT).show();
                        }}@Override
                      public void onCancelled(@NonNull DatabaseError error) {
                        // Getting data failed
         Toast.makeText(SendmoneyActivity2.this, "Failed to load data: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
                      }});} else {Toast.makeText(SendmoneyActivity2.this, "account no. and Re account no. not
match", Toast.LENGTH_SHORT).show();
                } } else {
                Toast.makeText(SendmoneyActivity2.this, "Fill all field For making payment",
Toast.LENGTH_SHORT).show();
            }}}); }
     public void FetchDataFromFirebase(String login_id) {
  DatabaseReference reference = FirebaseDatabase.getInstance().getReference().child("User_detail");
      reference.orderByChild("account_no");
      reference.addListenerForSingleValueEvent(new ValueEventListener() {
         @Override
         public void onDataChange(@NonNull DataSnapshot snapshot) {
            for (DataSnapshot userSnapshot : snapshot.getChildren()) {
               String accountNo = userSnapshot.getKey();
               if (accountNo.equals(login_id)) {
                  // Toast.makeText(SendmoneyActivity2.this, "line 110", Toast.LENGTH_SHORT).show();
                  String acc_number = userSnapshot.child("account_no").getValue(String.class);
                  String account_balance = userSnapshot.child("balance").getValue(String.class);
float balance = Float.parseFloat(account_balance);
                  acc_balance.setText("Account balance\n₹" + balance);
                  account_number.setText("Account Number\n" + acc_number);
                  break; // Exit the loop since we found the desired record
               }}}
  @Override
```

```
            public void onCancelled(@NonNull DatabaseError error) {
                // Getting data failed
                Toast.makeText(SendmoneyActivity2.this, "Failed to Show Account detail Please try again later: " +
error.getMessage(), Toast.LENGTH_SHORT).show();
            }});}DatabaseReference Userreference =
FirebaseDatabase.getInstance().getReference().child("User_detail").child(login_id).child("Transction");
//Transction id genrator using random function
        Random random;
        random = new Random();
        StringBuilder sb = new StringBuilder();
        // Generate the first 12 digits randomly
        for (int i = 0; i < 12; i++) {
            sb.append(random.nextInt(10));  }
// Append a random non-zero digit for the 13th digit
        sb.append(random.nextInt(9) + 1);
        long randomNumber = Long.parseLong(sb.toString());
      //  System.out.println("Random 13-digit number: " + randomNumber);
        // Output the generated random number
        String Transction_id = String.valueOf(randomNumber);
        // Get current date and time
        LocalDateTime currentDateTime = LocalDateTime.now();
        // Define date time formatter
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
        // Format and print current date and time
        String formattedDateTime = currentDateTime.format(formatter);
        String Transction_time = String.valueOf(formattedDateTime);
        // Transction details
        SendmoneyActivity2 transction = new SendmoneyActivity2();
        String Transction_to = re_acc_no;
       String Transction_from = login_id;
        String Transction_amount = (amount+"\t-");
        Transctions transction_data = new Transctions(Transction_time, Transction_to, Transction_from,
Transction_amount,Transction_id); // Assuming User is your data model class
        // Push the data to Firebase Realtime Database
        Userreference.child(Transction_id).setValue(transction_data);
        DatabaseReference reference =
FirebaseDatabase.getInstance().getReference().child("User_detail").child(login_id).child("Transction");
        reference.child(Transction_id).setValue(transction_data);
        //for recivert
        Transction_amount = ( amount+" \t+");
        Transctions reciver_transction_data = new Transctions(Transction_time, Transction_to, Transction_from,
Transction_amount,Transction_id); // Assuming User is your data model class
        DatabaseReference re_reference =
FirebaseDatabase.getInstance().getReference().child("User_detail").child(re_acc_no).child("Transction");
        re_reference.child(Transction_id).setValue(reciver_transction_data); }
    //login_id,reaccount_number,re_Cu_balance,se_Cu_balance
    public void updateData(String login_id, String reAccountNumber, String re_Cu_balance, String
seCuBalance) {
        // Get instance of Firebase database
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        // Reference to the database node where data will be updated
        DatabaseReference myRef = database.getReference("User_detail").child(login_id);
        DatabaseReference myRref = database.getReference("User_detail").child(reAccountNumber);
        HashMap hashMap = new HashMap();
        HashMap hashMap1 = new HashMap();
        hashMap.put("balance", re_Cu_balance);
        hashMap1.put("balance", seCuBalance);
        myRef.updateChildren(hashMap1);
        myRref.updateChildren(hashMap);
    }}
```

- ***Sendmoney Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".SendmoneyActivity2">
    <ImageView
        android:id="@+id/imageView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/headerlogo" />
    <EditText
        android:id="@+id/User_pin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:drawableLeft="@drawable/password"
        android:ems="06"
        android:hint="Pin (XXX-XXX)"
        android:inputType="numberPassword"
        app:layout_constraintTop_toBottomOf="@+id/Amount"
        tools:layout_editor_absoluteX="0dp" />
    <EditText
        android:id="@+id/Amount"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:ems="06"
        android:hint="Enter Amount"
        android:inputType="numberDecimal"
        app:layout_constraintTop_toBottomOf="@+id/Re_Reciver_account_no"
        tools:layout_editor_absoluteX="0dp" />
    <EditText
        android:id="@+id/Re_Reciver_account_no"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:drawableLeft="@drawable/account"
        android:hint="Re-account Number(XXX-XXX-XXXX)"
        android:inputType="phone"
        app:layout_constraintTop_toBottomOf="@+id/Reciver_account_no"
        tools:layout_editor_absoluteX="0dp" />
    <EditText
        android:id="@+id/Reciver_account_no"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="200dp"
        android:drawableLeft="@drawable/account"
        android:hint="Account Number(XXX-XXX-XXXX)"
        android:inputType="phone"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView6" />
    <Button
        android:id="@+id/Pay"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="32dp"
        android:fontFamily="@font/regular"
        android:text="Pay"
        app:layout_constraintBottom_toTopOf="@+id/textView4"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/acc_balance"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="75dp"
        android:layout_marginEnd="82dp"
        android:text="TextView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView6" />
    <TextView
        android:id="@+id/account_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="75dp"
        android:text="TextView"
        app:layout_constraintEnd_toStartOf="@+id/acc_balance"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView6" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Slash Activity (Java)***

```
package com.example.ebank;
import androidx.appcompat.app.AppCompatActivity;
import android.os.*;
import android.content.Intent;
public class SlashActivity2 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Set the content view to the splash screen layout
        setContentView(R.layout.activity_slash2);
        // Start the main activity after a short delay
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                startActivity(new Intent(SlashActivity2.this, Loginregister.class));
                finish();
            }
        }, 5000); // Delay for 5 seconds
    }}
```

- ***Slash Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".SlashActivity2">
    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="190dp"
        android:layout_height="185dp"
        app:layout_constraintBottom_toTopOf="@+id/textView1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2"
        app:srcCompat="@drawable/bnkl" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="32dp"
        android:fontFamily="@font/sbold"
        android:text="Developed By\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/slashlogo" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Sucesspayment Activity (Java)***

```
package com.example.ebank;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
public class Sucess_payment extends AppCompatActivity {
    Button Done;
    TextView Tranction_ammount;
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sucess_payment);
        Tranction_ammount = findViewById(R.id.Tranction_ammount);
        Done = findViewById(R.id.Done);
        Intent intenT = getIntent();
        String send_ammount = intenT.getStringExtra("send_ammount");
        Tranction_ammount.setText(send_ammount);
        Done.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Sucess_payment.this, MainActivity.class);
                startActivity(intent);
          }});}}
```

- ***Successpayment Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".Sucess_payment">
    <ImageView
        android:id="@+id/imageView15"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <TextView
        android:id="@+id/textView16"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Done" />
    <ImageView
        android:id="@+id/imageView16"
        android:layout_width="176dp"
        android:layout_height="172dp"
        android:layout_marginTop="132dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/imageView15"
        app:srcCompat="@drawable/checked" />
    <TextView
        android:id="@+id/textView19"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:fontFamily="@font/sbold"
```

```
        android:text="Payment Sucess"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView16" />
    <TextView
        android:id="@+id/Tranction_ammount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="105dp"
        android:text="Tranction_ammount"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView19" />

<Button
        android:id="@+id/Done"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="154dp"
        android:text="Done"
        android:textColorLink="#673AB7"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Tranction_ammount" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Transction Class (Java)***

```
package com.example.ebank;
public class Transctions {
    public String transction_time;
    public String transction_to;
    public String transction_from;
    public String transction_amount;
    public String transction_id;
//Constructor
    public Transctions(String transction_time, String transction_to, String transction_from, String
transction_amount, String transction_id) {
        this.transction_time = transction_time;
        this.transction_to = transction_to;
        this.transction_from = transction_from;
        this.transction_amount = transction_amount;
        this.transction_id=transction_id;
    }
    //Empty Constructor
    public Transctions() {}
    public String getTransction_amount() {
        return transction_amount;
    }
    public void setTransction_amount(String transction_amount) {this.transction_amount = transction_amount;}
    public String getTransction_time() {
        return transction_time;
    }
    public void setTransction_time(String transction_time) {this.transction_time = transction_time;}
    public String getTransction_to() {
        return transction_to;
    }
    public void setTransction_to(String transction_to) {
        this.transction_to = transction_to;
    }
    public String getTransction_from() {
        return transction_from;
```

```
    }
    public void setTransction_from(String transction_from) {this.transction_from = transction_from;}
    public String getTransction_id() {
        return transction_id;
    }
    public void setTransction_id(String transction_id) {
        this.transction_id = transction_id;
    }}
```

- ***Travel Activity (Java)***

```
package com.example.ebank;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
public class TravelActivity extends AppCompatActivity {
    ImageButton rail,flight;
    Button Back2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_travel);
        rail = findViewById(R.id.rail);
        rail.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.irctc.co.in/nget/train-
search"));
                startActivity(intent);
            }
        });
        flight = findViewById(R.id.flight);
        flight.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.air.irctc.co.in/"));
                startActivity(intent);
            }
        });
        Back2=findViewById(R.id.Back2);
        Back2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(TravelActivity.this,MainActivity.class);
                startActivity(intent);
            }});}}
```

- ***Travel Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".TravelActivity">
    <ImageView
        android:id="@+id/imageView10"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <ImageButton
        android:id="@+id/rail"
        android:layout_width="136dp"
        android:layout_height="115dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="195dp"
        android:src="@drawable/railimg"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/Back2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:background="#D565C1E8"
        android:drawableLeft="@drawable/back"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Back"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <ImageButton
        android:id="@+id/flight"
        android:layout_width="127dp"
        android:layout_height="123dp"
        android:layout_marginStart="19dp"
        android:layout_marginTop="148dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/rail"
        app:srcCompat="@drawable/flightimg" />
    <TextView
        android:id="@+id/rail_ticket"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:fontFamily="@font/sbold"
        android:text="Railway Ticket"
        android:textSize="19dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/rail" />
    <TextView
        android:id="@+id/flight_ticket"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="28dp"
        android:fontFamily="@font/sbold"
        android:text="Flight Ticket"
        android:textSize="19dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/flight" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***User_data Class (Java)***

```
package com.example.ebank;
public class Userdata {
     private String account_holder_name;
     private String father_name;
     private String dob;
    private String phone_no;
    private String pin;
    private String email;
    private String address;
    private String addhar;
    public String account_no;
    public String balance;
    private String gender;
    // Constructor
    public Userdata(String account_holder_name, String account_no,String balance,String father_name, String
dob, String phone_no, String pin, String email, String address, String addhar, String gender) {
        this.account_holder_name = account_holder_name;
        this.account_no = account_no;
        this.father_name = father_name;
        this.dob = dob;
        this.phone_no = phone_no;
        this.pin = pin;
        this.balance=balance;
        this.email = email;
        this.address = address;
        this.addhar = addhar;
        this.gender = gender;
    }
 public Userdata() { }
 // Getter and setter
    public String getBalance() {
        return balance;  }
 public void setBalance(String balance) {
        this.balance = balance;  }
    public String getName() {
        return account_holder_name; }
    public void setName(String name) {
        this.account_holder_name = name;
    }
    public String getAccount_no() {
        return account_no;   }
    public void setAccount_no(String account_no) {
        this.account_no = account_no;
    }
    public String getFather_name() {
        return father_name;
    }
    public void setFather_name(String father_name) {
        this.father_name = father_name;
    }
    public String getDob() {
        return dob;
    }
    public void setDob(String dob) {
        this.dob = dob;
    }
    public String getPhone_no() {
        return phone_no;
    }
```

```
    public void setPhone_no(String phone_no) {
        this.phone_no = phone_no;
    }
    public String getPin() {
        return pin;
    }
    public void setPin(String pin) {
        this.pin = pin;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getAddhar() {
        return addhar;
    }
    public void setAddhar(String addhar) {
        this.addhar = addhar;
    }
    public String getGender() {
        return gender;
    }public void setGender(String gender) {
        this.gender = gender;  }}
```

- ***Validation Activity (java)***

```
package com.example.ebank;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.FirebaseException;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthProvider;
import java.util.concurrent.TimeUnit;
public class ValidationActivity extends AppCompatActivity {
Button back3,next;
    String phonenumber;
    EditText Otp;
    String otpid;
    FirebaseAuth mAuth;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_validation);
        Otp=findViewById(R.id.Otp);
        next=findViewById(R.id.next);
        phonenumber=getIntent().getStringExtra("mobile").toString();
        mAuth=FirebaseAuth.getInstance();
        initiateotp();
        next.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(Otp.getText().toString().isEmpty())
                    Toast.makeText(getApplicationContext(),"Blank Field can not be
processed",Toast.LENGTH_LONG).show();
                else if(Otp.getText().toString().length()!=6)
                    Toast.makeText(getApplicationContext(),"INvalid OTP",Toast.LENGTH_LONG).show();
                else
                { PhoneAuthCredential credential=PhoneAuthProvider.getCredential(otpid,Otp.getText().toString());
                    signInWithPhoneAuthCredential(credential);
                }} });
        back3=findViewById(R.id.back3);
        back3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(ValidationActivity.this, Register.class);
                startActivity(intent);
                finish();
            } }); }
    private void initiateotp()
    {
        PhoneAuthProvider.getInstance().verifyPhoneNumber(
                phonenumber,        // Phone number to verify
                60,                 // Timeout duration
                TimeUnit.SECONDS,   // Unit of timeout
                this,               // Activity (for callback binding)
                new PhoneAuthProvider.OnVerificationStateChangedCallbacks()
                {@Override
                    public void onCodeSent(String s, PhoneAuthProvider.ForceResendingToken
forceResendingToken)
                    {
                        otpid=s;   }
                    @Override
                    public void onVerificationCompleted(PhoneAuthCredential phoneAuthCredential)
                    { signInWithPhoneAuthCredential(phoneAuthCredential);
                    } @Override
                    public void onVerificationFailed(FirebaseException e) {
                        Toast.makeText(getApplicationContext(),e.getMessage(),Toast.LENGTH_LONG).show();
                    }});        // OnVerificationStateChangedCallbacks
    }
    private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
        mAuth.signInWithCredential(credential) .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful())
                        {startActivity(new Intent(ValidationActivity.this,MainActivity.class));
                            finish();
                        } else {
```

```
                    Toast.makeText(getApplicationContext(),"Signin Code
Error",Toast.LENGTH_LONG).show();
                }}});}}
```

- ***Validation Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/grdnt"
    tools:context=".ValidationActivity">
    <ImageView
        android:id="@+id/imageView11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/headerlogo" />
    <ImageView
        android:id="@+id/imageView12"
        android:layout_width="194dp"
        android:layout_height="189dp"
        android:layout_marginTop="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView11"
        app:srcCompat="@drawable/bnkl" />
    <EditText
        android:id="@+id/Otp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="200dp"
        android:drawableLeft="@drawable/password"
        android:ems="06"
        android:hint="OTP(XXX-XXX)*"
        android:inputType="numberPassword"
        app:layout_constraintBottom_toTopOf="@+id/textView2"
        app:layout_constraintTop_toBottomOf="@+id/imageView12"
        tools:layout_editor_absoluteX="0dp" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:fontFamily="@font/sbold"
        android:text="Developed BY\nAkash Pandey"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/sbold"
        android:text="Enter OTP"
        android:textSize="34sp"
        app:layout_constraintBottom_toTopOf="@+id/Otp"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView12"
        app:layout_constraintVertical_bias="0.244" />
    <Button
        android:id="@+id/back3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_marginBottom="10dp"
        android:background="#D565C1E8"
        android:drawableLeft="@drawable/back"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Back"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/next"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />
    <Button
        android:id="@+id/next"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_marginBottom="12dp"
        android:background="#D565C1E8"
        android:drawableRight="@drawable/next"
        android:fontFamily="@font/regular"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:padding="10dp"
        android:text="Next"
        android:textColor="#000000"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/textView2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ***Singlerow Activity (XML)***

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
  android:layout_height="wrap_content">
    <TextView
        android:id="@+id/Trans_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:fontFamily="@font/sbold"
        android:text="Text 1" />
    <TextView
        android:id="@+id/Trans_time"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Text 2"
```

```
        android:fontFamily="@font/sbold"
        android:layout_below="@+id/Trans_id"
        android:layout_centerHorizontal="true" />
    <TextView
        android:id="@+id/Trans_to"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Text 3"
        android:fontFamily="@font/sbold"
        android:layout_below="@+id/Trans_time"
        android:layout_centerHorizontal="true" />
    <TextView
        android:id="@+id/Trans_from"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Text 4"
        android:fontFamily="@font/sbold"
        android:layout_below="@+id/Trans_to"
        android:layout_centerHorizontal="true" />
    <TextView
        android:id="@+id/Trans_ammount"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Text 5"
        android:fontFamily="@font/sbold"
        android:layout_below="@+id/Trans_from"
        android:layout_centerHorizontal="true" />
    <TextView
        android:id="@+id/Trans_idd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Text 6"
        android:fontFamily="@font/sbold"
        android:layout_below="@+id/Trans_ammount"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

9.CONCLUSION

The development of this Gramin E bank application using Android, Java, and Firebase has been a challenging yet rewarding experience. The app aims to provide a secure and convenient platform for users to perform various banking transactions, such as fund transfers, bill payments, and account management, from the comfort of their mobile devices.
Throughout the development process, several key features were successfully implemented. The app integrates Firebase Authentication, allowing users to securely sign in with their credentials or use popular social media accounts for seamless authentication. Firebase Realtime Database was utilized to store and retrieve user data, ensuring real-time synchronization and efficient data management.

The user interface was designed with a focus on intuitive navigation and a modern, visually appealing layout. The app adheres to Material Design guidelines, providing a consistent and familiar experience across different Android devices and versions.

Security was a top priority during the development phase. Sensitive data, such as user credentials and financial information, is encrypted and securely stored in the Firebase Realtime Database. Furthermore, industry-standard security practices, such as input validation and secure communication channels, were implemented to protect against potential vulnerabilities and threats.

While the app has achieved its core objectives, there is always room for improvement and future enhancements. Potential areas for future development include integrating additional payment gateways, adding support for biometric authentication, and implementing advanced features like budgeting tools and investment tracking.

10.FUTURE SCOPE

- **Payment Section**
  We can add multiple payment option like bill payment, mobile recharge and smart payment just like a QR code scanner.

- **Security and Encryption**
  We can add Two step verification for making secure payment.

- **Service and Support**
  24*7 customer support using AI and machine learning.

11.REFERENCE

- ***Resource***

1 www.javatpoint.com/software-engineering-software-development-life-cycle
2 www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
3 https://javatpoint.com/android-tutorial
4 https://www.blackbox.ai/
5 https://claude.ai/chats
6 https://firebase.google.com/docs/database/android/start

- ***Platform***

  Android Studio
  Smart phone with android operating system