ABSTRACT

The management of Training and Placement is supported by paper-based systems, databases, spreadsheets and E-mail communications. Training and Placement is the crucial part of any educational institute in which most of the work till now is being done manually. The project will include minimum manual work and maximum optimization, abstraction and security.

The PLACEMENT ENQUIRY SYSTEM is an Android-based application developed in windows platform for the placement department of the college in order to provide the details of its students in a database for the companies to their process of recruitment provided with a proper login. throughout the organization and outside as well with proper login provided. The system can used for college to manage the student information with regards to placement details. This project contains all the details of the students that can be viewed by all the users (read only), but can be modified only with an authorized service.

So, our project provides a facility of maintaining the details of the students, and gets the requested list of candidates for the companies who would like to recruit the people. Our project provides the facility of maintaining the details of the students. It reduces the manual work and consumes less paper work to reduce the time. This project is developed with KOTLIN.

TABLE OF CONTENTS

CHAPTER -1

1.INTRODUCTION

The purpose of the project “PLACEMENT MANAGEMENT SYSTEM”, is the manual work makes the process slow and other problems such as inconsistency and ambiguity on operations. In order to avoid this Android-based placement managed system is proposed, where the student information in the college with regard to placement is managed efficiently. It intends to help fast in fast access procedures in placement related activities and ensures to maintain the details of the student. The key feature of this project is that it is one time registration enabled. The placement cell calls the companies to select their students for jobs via the campus interview. The job details of the placed students will be provided by the administrator. Our project provides the facility of maintaining the details of the students and gets the requested list of candidates for the company who would like to recruit the students.

CHAPTER – 2

2.CONCEPTS AND METHODS

2.1 PROBLEM DESCRIPTION

All the process in existing system are handled manually. All the work that is done in existing system is done by the human intervention. As all the work is done manually, there will a lot of burden on placement officer and it also increases the maximum chance of errors. This is slow and time consuming.

Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information is difficult and lengthy. The records were never used to be in systematic order. If any information had to be found it is required to go through the different registers, documents. Once the records were entered it will be difficult to update the records.

The reason behind it is that there is a lot of information to be maintained and have to be kept in mind while doing the work. So, we have to use the computerized version to solve the difficulties of existing system.

2.2 PROPOSED SOLUTION

The aim of the proposed system is to automate the existing manual system by the help of computerized equipment and full-fledged software, full filling all the requirements of the user, so that the data/information can be stored for longer period with easy accessing and manipulation of same.

The Android based Training and Placement Application gives more easiness to TPO, Placement coordinators and Students that they can modify and access information so quickly. The system provides a better way to maintain students information in the database, ensures data correctness and data integrity as well. The system also reduces the paperwork time and provides an efficient information flow between different system modules. Our project provides good performance and services to the users.

2.3 SYSTEM REQUIREMENTS

2.3.1SoftwareUsed*:*

This section describes the software requirements which are used for developing the project.

S.No	Software	Requirement
1.	Operating system	Windows/Mac/Linux/Chrome Os
2.	Android Studio	Latest version
3.	Database	Firebase

2.3.2HardwareUsed*:*

This section describes the hardware requirements which are used for developing the project.

S.No	Hardware	Requirement
1.	Processor	2.4 GHz Processor speed
2.	RAM	minimum 4GB
3.	disk space	8 GB of available disk space minimum

CHAPTER - 3

3. IMPLEMENTATION:

3.1 Components Used:-

1. Activity
2. Services
3. Intents
4. Widgets
5. Views
6. Notifications
7. Layout XML Files
8. App APK Files
9. JSON Data
10. Firebase Storage
11. Firebase Real time Database
12. Firebase Authentication

3.1.1 Activity:-

An activity is a class that is considered as an entry point for users that represents a single screen. A messenger application might have an activity that shows a new notification, another activity which reads messages and another which composes a new message. Each activity is independent of one another. Forexample – camera application can be started in an email application to compose an email that shares an image. The picture below depicts how each new activity adds an item to back stack and how the current activity is destroyed and previous activity is resumed.

3.1.2 Services:-

A service is a component that runs in the back ground, it acts as an invisible worker of our application. It keeps updating data sources and activities. It also broadcasts intents and performs tasks when applications are not active. An example of service is we can surf the internet or use any other application while listening to music.

3.1.3 Intents:-

It is an inter-application message passing framework for communication between android components. It is also used for transferring data between different Activities as well as to start a new service and display a list of contacts in List View. Example – the camera application sends an intent to the operating system when the user decides to share a picture.

3.1.4 Widgets:-

Widgets are variations of Broadcast Receivers and essential aspects of homescreencustomization. They display data and allow users to perform actions on them. There are various types of widgets:

- ***Information widget*:** These widgets display crucial information and track how the information changes over time. Example – Clock widgets and widgets that display weather and time information.

- ***Collection widget*:** As the name depicts, collection widgets are a collection of information of the same type. Its use is for browsing information and opening any one of the elements to view details. Example – music widgets, as we can skip pause and play music outside the music application.

- ***Control widget*:** These widgets display functionalities and by using them, the user can trigger from home screen without opening the application. Example – pause and play the video outside the application.

- ***Hybrid widget*:** These widgets combine features of all the other three widgets. Example – music player widget is a control widget but it also informs the user about which track is playing currently, which means it is a combination of control and information thus it is termed as hybrid widget.

3.1.5 Views:-

View is responsible for drawing and event handling. They are rectangular elements on the screen. Some of the views are Edit Text, Image View Button, Check Box and Image Button.

3.1.5.1 ImageView*:-*

In Android, Image View class is used to display an image file in application. Image file is easy to use but hard to master in Android, because of the various screen sizes in Android devices. Scale type options are used for scaling the bounds of an image to the bounds of the image view.

3.1.5.2 Text View*:-*

Text View in Android is one of the basic and important UI elements. This plays a very important role in the UI experience and depends on how the information is displayed to the user. A Text View displays text to the user and optionally allows them to edit it. A Text View is a complete text editor, however the basic class is configured to not allow editing.

3.1.5.3 Button*:-*

In Android applications, a **Button** is a user interface that is used to perform some action when clicked or tapped. It is a very common widget in Android and developers often use it. A button consists of text or an icon (or both text and an icon) that communicates what action occurs when the user touches it.

3.1.5.4 Edit Text*:-*

The Edit Text is a user interface which is used for entering and changing the text. The Edit Text is the standard text entry widget in Android apps. If the user needs to enter text into an app, this is the primary way for them to do that. There are many important properties that can be set to customize the behavior of an Edit Text.

3.1.5.5 Card View*:-*

Card View is a new widget in Android that can be used to display any sort of data by providing a rounded corner layout along with a specific elevation. Card View is the view that can display views on top of each other. The main usage of Card View is that it helps to give a rich feel and look to the UI design. This widget

can be easily seen in many different Android Apps. Card View can be used for creating items in list view or inside Recycler View. The best part about Card View is that it extends Frame layout and it can be displayed on all platforms of Android. Now we will see the simple example of Card View implementation.

3.1.5.6 Recycler View*:-*

Recycler View is a View Group added to the android studio as a successor of the Grid View and List View. It is an improvement on both of them and can be found in the latest v-7 support packages. It has been created to make possible construction of any lists with **XML** layouts as an item which can be customized vastly while improving on the efficiency of List Views and Grid Views.

This improvement is achieved by recycling the views which are out of the visibility of the user. For example, if a user scrolled down to a position where items 4 and 5 are visible; items 1, 2, and 3 would be cleared from the memory to reduce memory consumption.

It is a container for displaying large datasets which can be scrolled efficiently by maintaining limited number of views. You can use Recycler View widget when you have data collections whose elements change at runtime depend on network event or user action

3.1.6 Notifications*:-*

A notification is a message that Android displays outside your app's UI to provide the user with reminders, communication from other people, or other timely information from your app. Users can tap the notification to open your app or take an action directly from the notification.

It alerts users when the application is not visible or is inactive. This alert flashes on the screen and then disappears. Example – Notification of the new incoming message popped on the screen.

3.1.6.1 Toast Messages*:-*

An Android Toast is a small message displayed on the screen, similar to a tool tip or other similar popup notification. A Toast is displayed on top of the main content of an activity, and only remains visible for a short time period.

A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout.

3.1.7 Layout XML Files

Layout is the structure for the user interface in the application. XML files provide different types of layouts for the different type of screen, it also specifies which GUI component, an activity or fragment holds.

3.1.8 App APK Files

Apk file is the package file format that contains the program's code, resources, assets. The Android operating system uses them for installing mobile applications and middleware.

3.1.9 JSON Data:

JSON, or JavaScript Object Notation, is a format used to represent data. It was introduced in the early 2000s as part of JavaScript and gradually expanded to become the most common medium for describing and exchanging text-based data.

3.1.10 Firebase Storage :

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality.

3.1.11 Firebase Real Time Database :

The Firebase Real time Database is a cloud-hosted database. Data is stored as JSON and synchronized in real time to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Real time Database instance and automatically receive updates with the newest data.

3.1.12 Firebase Authentication :

Authentication is the process of determining whether someone or something is, in fact, who or what it says it is. During authentication, credentials provided by the user are compared to those on file in a database of authorized users' information either on the local operating system server or through an authentication server. If the credentials entered match those on file and the authenticated entity is authorized to use the resource, the user is granted access.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Face book and Twitter, and more.

Authenticate users with their email addresses and passwords. The Firebase Authentication SDK provides methods to create and manage users that use their email addresses and passwords to sign in. Firebase Authentication also handles sending password reset emails.

3.2 PSEUDO CODE/ALGORITHMS

Code

3.2.1***Code for Activity***

Code for sign in activity:-

```
package com.example.demoapp

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import com.example.demoapp.databinding.ActivitySignInBinding
import com.google.firebase.auth.FirebaseAuth

class SignInActivity : AppCompatActivity() {
    private lateinit var binding: ActivitySignInBinding
    private lateinit var firebaseAuth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivitySignInBinding.inflate(layoutInflater)
        setContentView(binding.root)


        firebaseAuth = FirebaseAuth.getInstance()
        binding.textView.setOnClickListener {
            val intent = Intent(this, SignUpActivity::class.java)
            startActivity(intent)
        }

        binding.button.setOnClickListener {
            val email = binding.emailEt.text.toString()
            val pass = binding.passET.text.toString()

            if (email.isNotEmpty() && pass.isNotEmpty()) {

                firebaseAuth.signInWithEmailAndPassword(email,
pass).addOnCompleteListener {
                    if (it.isSuccessful) {
                        val intent = Intent(this, MainActivity::class.java)
                        startActivity(intent)
                    } else {
                        Toast.makeText(this, it.exception.toString(),
Toast.LENGTH_SHORT).show()

                    }
                }
```

```
            } else {
                Toast.makeText(this, "Empty Fields Are not Allowed !!",
    Toast.LENGTH_SHORT).show()

            }
        }
    }
```

Code for Signup Activity:-

```
package com.example.demoapp

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.LayoutInflater
import android.widget.Toast
import com.example.demoapp.databinding.ActivitySignUpBinding
import com.google.firebase.auth.FirebaseAuth

class SignUpActivity : AppCompatActivity() {

    private lateinit var binding: ActivitySignUpBinding
    private lateinit var firebaseAuth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivitySignUpBinding.inflate(layoutInflater)
        setContentView(binding.root)

        firebaseAuth = FirebaseAuth.getInstance()

        binding.textView.setOnClickListener {
            val intent = Intent(this, SignInActivity::class.java)
            startActivity(intent)
        }
        binding.button.setOnClickListener {
            val email = binding.emailEt.text.toString()
            val pass = binding.passET.text.toString()
            val confirmPass = binding.confirmPassEt.text.toString()

            if (email.isNotEmpty() && pass.isNotEmpty() &&
    confirmPass.isNotEmpty()) {
                if (pass == confirmPass) {

                    firebaseAuth.createUserWithEmailAndPassword(email,
    pass).addOnCompleteListener {
                        if (it.isSuccessful) {
                            val intent = Intent(this, SignInActivity::class.java)
                            startActivity(intent)
```

```
                    } else {
                        Toast.makeText(this, it.exception.toString(),
Toast.LENGTH_SHORT).show()

                    }
                }
            } else {
                Toast.makeText(this, "Password is not matching",
Toast.LENGTH_SHORT).show()
            }
        } else {
            Toast.makeText(this, "Empty Fields Are not Allowed !!",
Toast.LENGTH_SHORT).show()

        }
    }
}
```

Code for Main Activity:- class MainActivity:

```
AppCompatActivity() {


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

valstudent_wise:Button=          findViewById(R.id.student_wise)
valcompany_wise:Button= findViewById(R.id.company_wise)

student_wise.setOnClickListener{
valintent = Intent(this,MainActivity2::class.java)
startActivity(intent)
}

company_wise.setOnClickListener{
valintent = Intent(this,MainActivity4::class.java)
startActivity(intent)
}
}
}
```

Code for Main Activity 2:-

```
class MainActivity2 : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
```

```
setContentView(R.layout.activity_main2)

valget_details: Button = findViewById(R.id.button)
valhome_button: Button = findViewById(R.id.home_button)
valrollnumberfield: EditText=findViewById(R.id.textfield1)

get_details.setOnClickListener{

if(rollnumberfield.text.toString().isNotEmpty()) {

valintent = Intent(this, MainActivity3::class.java)
intent.putExtra("Student_roll",
rollnumberfield.text.toString().toUpperCase())
startActivity(intent)
        }
else{
Toast.makeText(applicationContext,"Enter Roll
 Number",Toast.LENGTH_SHORT).show()
        }
}

home_button.setOnClickListener{
valintent = Intent(this, MainActivity::class.java)
startActivity(intent)
}
}
}
```

Code for Main Activity 4:-

```
class MainActivity4 :
AppCompatActivity() {     override fun
onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main4)

supportActionBar?.hide()

valgetdetailsc: Button = findViewById(R.id.get_details_c)
valhomeButton: Button = findViewById(R.id.Home_Button)
valcompanyname: EditText=findViewById(R.id.company_name_input)


homeButton.setOnClickListener{
valintent = Intent(this, MainActivity::class.java)
startActivity(intent)
}

getdetailsc.setOnClickListener{
```

```
if(companyname.text.toString().isNotEmpty()) {
valintent = Intent(this, MainActivity5::class.java)
intent.putExtra("company_name",
companyname.text.toString().toLowerCase())
startActivity(intent)
        }
else{
Toast.makeText(applicationContext,"Enter Company Name",
Toast.LENGTH_SHORT).show()
        }
}
}
}
```

Code for Main Activity 5:-

```
class MainActivity5 : AppCompatActivity() {

val database: FirebaseDatabase = FirebaseDatabase.getInstance()
valdatabaseReference = database.getReference("17-21 batch")

val array1: MutableList<String> = mutableListOf<String>()    val
array2: MutableList<String> = mutableListOf<String>()    val
array3: MutableList<Bitmap> = mutableListOf<Bitmap>()

  override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main5)

supportActionBar?.hide()

      home_button_activity5.setOnClickListener {
val intent = Intent(this, MainActivity4::class.java)
startActivity(intent)
      }

lateinitvar adapter:
RecyclerView.Adapter<RecyclerAdapter.ViewHolder>lateinitvarlayoutManager:
RecyclerView.LayoutManager

val company: String? = intent.getStringExtra("company_name")
company_name_description.text = company

layoutManager = LinearLayoutManager(this, LinearLayoutManager.VERTICAL,
false)
traineeDetailsRecyclerView.layoutManager = layoutManager
```

```
databaseReference.child("company wise").child(company.toString())
.addValueEventListener(object :              ValueEventListener {
            override  fun  onDataChange(snapshot:  DataSnapshot)  {
if (snapshot.exists()) {

for            (i            in            snapshot.children)            {
array1.add(i.key.toString())
                    array2.add(i.getValue().toString())
                }

adapter = RecyclerAdapter(array1, array2)
traineeDetailsRecyclerView.adapter = adapter
                }
            }

            override fun onCancelled(error: DatabaseError) {
                TODO("Not yet implemented")
            }
        })
    }
}
```

3.2.2***Code for layouts:-***

Code for Layout 1:-

```
<?xml version="1.0" encoding="utf-
8"?><androidx.constraintlayout.widget.ConstraintLayoutxmlns: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="@color/background"
android:orientation="vertical"
tools:context=".MainActivity">


<Button        android:id="@+id/student_wise"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Student wise details"
android:textColor="@color/black"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.487"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
```

```
app:layout_constraintVertical_bias="0.427"
android:background="@drawable/input"
app:backgroundTint="#DA8E8E"/>

<Button
android:id="@+id/company_wise"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="28dp"        android:text="Compnay
Wise Details"        app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/student_wise"
android:background="@drawable/input"
app:backgroundTint="#DA8E8E"/>


</androidx.constraintlayout.widget.ConstraintLayout>
```

Code for Layout 2:-

```
<?xml version="1.0" encoding="utf-
8"?><androidx.constraintlayout.widget.ConstraintLayoutxmlns: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:orientation="vertical"
android:background="@color/background"
tools:context=".MainActivity2">

<ImageViewandroid:id="@+id/vvit_logo_s"
android:layout_width="368dp"
android:layout_height="268dp"
android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/vvit_logo1" />

<TextView
android:id="@+id/student_wise_enquiry_text"
android:layout_width="343dp"
android:layout_height="37dp"
android:layout_below="@id/vvit_logo_s"
android:layout_centerHorizontal="true"
android:layout_marginTop="300dp"
android:text="Student Wise Details"
```

```
android:gravity="center"
android:textColor="@color/black"
android:textScaleX="1.5"
android:textSize="20sp" android:textStyle="bold"
app:layout_constraintEnd_toEndOf="@+id/textfield1"
app:layout_constraintHorizontal_bias="0.84"
app:layout_constraintStart_toStartOf="@+id/textfield1"
app:layout_constraintTop_toTopOf="@+id/vvit_logo_s" />

<EditTextandroid:id="@+id/textfield1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/student_wise_enquiry_text"
android:layout_centerHorizontal="true"
android:layout_marginTop="12dp"
android:background="@drawable/input"
android:ems="10"        android:hint="Roll
Number"
android:inputType="textPersonName"
android:paddingLeft="20dp"
app:layout_constraintEnd_toEndOf="pare
nt"
app:layout_constraintHorizontal_bias="0.
509"
app:layout_constraintStart_toStartOf="par
ent"
app:layout_constraintTop_toBottomOf="@+id/student_wise_enquiry_text" />

<Button
android:id="@+id/button"        android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/textfield1"
android:layout_centerHorizontal="true"
android:layout_marginTop="24dp"
android:background="@drawable/input"
android:text="Get Details"        app:backgroundTint="#DA8E8E"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.509"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textfield1" />

<Button        android:id="@+id/home_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/textfield1"
android:layout_centerHorizontal="true"
android:layout_marginTop="24dp"
android:background="@drawable/input"
android:text="Home"
app:backgroundTint="#DA8E8E"
```

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.509"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/button" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

Code for Layout 4:-

```
<?xml version="1.0" encoding="utf-
8"?><androidx.constraintlayout.widget.ConstraintLayoutxmlns: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="@color/background"     android:orientation="vertical"
tools:context=".MainActivity4">

<ImageViewandroid:id="@+id/vvit_logo_c"
android:layout_width="368dp"
android:layout_height="268dp"
android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/vvit_logo1" />

<TextView
android:id="@+id/compnay_wise_enquiry_text"
android:layout_width="343dp"
android:layout_height="37dp"
android:layout_below="@id/vvit_logo_c"
android:layout_centerHorizontal="true"
android:layout_marginTop="300dp"
android:text="Company Wise Details"
android:textColor="@color/black"
android:textScaleX="1.5"
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintEnd_toEndOf="@+id/company_name_input"
app:layout_constraintHorizontal_bias="0.84"
```

```
app:layout_constraintStart_toStartOf="@+id/company_name_input"
app:layout_constraintTop_toTopOf="@+id/vvit_logo_c" />

<EditTextandroid:id="@+id/company_name_input"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/compnay_wise_enquiry_text"
android:layout_centerHorizontal="true"
android:layout_marginTop="12dp"
android:background="@drawable/input"
android:ems="10"        android:hint="Company
Name"
android:inputType="textPersonName"
android:paddingLeft="20dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.509"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/compnay_wise_enquiry_text" />

<Button
android:id="@+id/get_details_c"        android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/company_name_input"
android:layout_centerHorizontal="true"
android:layout_marginTop="24dp"
android:background="@drawable/input"
android:text="Get Details"        app:backgroundTint="#DA8E8E"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.509"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/company_name_input" />

<Button        android:id="@+id/Home_Button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/company_name_input"
android:layout_centerHorizontal="true"
android:layout_marginTop="24dp"
android:background="@drawable/input"
android:text="Home"        app:backgroundTint="#DA8E8E"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.509"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/get_details_c" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Code for Layout 5:-

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollViewxmlns: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:background="@color/background"
android:layout_height="match_parent">

<androidx.constraintlayout.widget.ConstraintLayoutandroid:layout_width="match
_parent"        android:layout_height="wrap_content"
tools:context=".MainActivity5">


<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"            android:gravity="center"
android:text="Compnay Wise Details"
android:textColor="@color/black"            android:textSize="30sp"
android:textStyle="bold"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />


<TextView
android:id="@+id/company_name_description"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:gravity="center"
android:text="TextView"
android:textColor="@color/black"
android:textSize="30sp"
android:textStyle="bold"
app:layout_constraintEnd_toEndOf="@+id/textView"
app:layout_constraintStart_toStartOf="@+id/textView"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<androidx.recyclerview.widget.RecyclerViewandroid:id="@+id/traineeD
etailsRecyclerView"            android:layout_width="409dp"
android:layout_height="wrap_content"
android:layout_marginTop="24dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/company_name_description"
tools:listitem="@layout/mylayout" />
```

```
<Button
android:id="@+id/home_button_activity5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:background="@drawable/input"
android:text="Back"            app:backgroundTint="#DA8E8E"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/traineeDetailsRecyclerView" />

<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text=""
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/home_button_activity5" />


</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>
```

Layout file for card view used in Layout 5:-

```
<?xml version="1.0" encoding="utf-
8"?><androidx.cardview.widget.CardViewxmlns:android="htt
p://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"
app:cardUseCompatPadding="true"
app:cardCornerRadius="25dp"     app:contentPadding="5dp"
app:cardBackgroundColor="@color/white"
android:layout_height="wrap_content">


<LinearLayoutandroid:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal">


<LinearLayout
android:layout_width="wrap_content"
android:layout_height="match_parent"           android:layout_weight="1"
```

```
android:orientation="horizontal">

<LinearLayout
android:layout_width="29dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:orientation="vertical">

<TextViewandroid:id="@+id/student_name"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Student Name:"
android:textColor="@color/black"
android:textStyle="bold"
android:textSize="20sp" />

<TextViewandroid:id="@+id/annual_package"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Package:"
android:textColor="@color/black"
android:textStyle="bold"
android:textSize="20sp" />

<TextViewandroid:id="@+id/student_branch"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Branch:"
android:textColor="@color/black"
android:textStyle="bold"
android:textSize="20sp" />
</LinearLayout>

<LinearLayoutandroid:layout_width="wrap_content"
android:layout_height="match_parent"
android:layout_weight="1"
android:orientation="vertical">

<TextViewandroid:id="@+id/trainee_name"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="19BQ1A0553"
android:textColor="@color/black"
android:textStyle="bold"
android:textSize="20sp" />

<TextViewandroid:id="@+id/trainee_package"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="12LPA"
```

```
android:textColor="@color/black"
android:textStyle="bold"
android:textSize="20sp" />

<TextViewandroid:id="@+id/trainee_branch"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="CSE"
android:textColor="@color/black"
android:textStyle="bold"
android:textSize="20sp" />
</LinearLayout>



</LinearLayout>

</LinearLayout>
</androidx.cardview.widget.CardView>
```

Layout file for sign in Activity:-

```
<?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/signin_screen"
    tools:context=".SignInActivity">


    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/emailLayout"
        style="@style/LoginTextInputOuterFieldStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="16dp"
        android:layout_marginTop="350dp"
        app:boxStrokeColor="@color/yellow"
        app:hintTextColor="@color/yellow"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/emailEt"
            style="@style/LoginTextInputInnerFieldStyle"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Type your Email"
        android:inputType="textEmailAddress" />
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/passwordLayout"
    style="@style/LoginTextInputOuterFieldStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="16dp"
    android:layout_marginTop="24dp"
    app:boxStrokeColor="@color/yellow"
    app:hintTextColor="@color/yellow"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/emailLayout"
    app:passwordToggleEnabled="true">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/passET"
        style="@style/LoginTextInputInnerFieldStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Type Your Password"
        android:inputType="textPassword" />
</com.google.android.material.textfield.TextInputLayout>

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="16dp"
    android:background="@drawable/input"
    app:backgroundTint="@color/yellow"
    android:text="Sign In"
    android:textColor="@color/black"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/passwordLayout" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:text="Not Registered Yet , Sign Up !"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button" />

    </androidx.constraintlayout.widget.ConstraintLayout>
```

Layout file for Signup Activity:-

```
    <?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/signup_screen"
        tools:context=".SignUpActivity">


        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/emailLayout"
            style="@style/LoginTextInputOuterFieldStyle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginHorizontal="16dp"
            android:layout_marginTop="300dp"
            app:boxStrokeColor="@color/yellow"
            app:hintTextColor="@color/yellow"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">

            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/emailEt"
                style="@style/LoginTextInputInnerFieldStyle"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Type your Email"
                android:inputType="textEmailAddress" />
        </com.google.android.material.textfield.TextInputLayout>

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/passwordLayout"
            style="@style/LoginTextInputOuterFieldStyle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginHorizontal="16dp"
```

```
    android:layout_marginTop="24dp"
    app:boxStrokeColor="@color/yellow"
    app:hintTextColor="@color/yellow"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/emailLayout"
    app:passwordToggleEnabled="true">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/passET"
        style="@style/LoginTextInputInnerFieldStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Type Your Password"
        android:inputType="textPassword" />
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/confirmPasswordLayout"
    style="@style/LoginTextInputOuterFieldStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="16dp"
    android:layout_marginTop="24dp"
    app:boxStrokeColor="@color/yellow"
    app:hintTextColor="@color/yellow"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/passwordLayout"
    app:passwordToggleEnabled="true">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/confirmPassEt"
        style="@style/LoginTextInputInnerFieldStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Retype Your Password"
        android:inputType="textPassword" />
</com.google.android.material.textfield.TextInputLayout>

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="16dp"
    android:background="@drawable/input"
    app:backgroundTint="@color/yellow"
    android:text="Sign Up"
    android:textColor="@color/black"
    android:textStyle="bold"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/confirmPasswordLayout" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/black"
        android:text="Already Registered , Sign In !"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Android manifest file:-

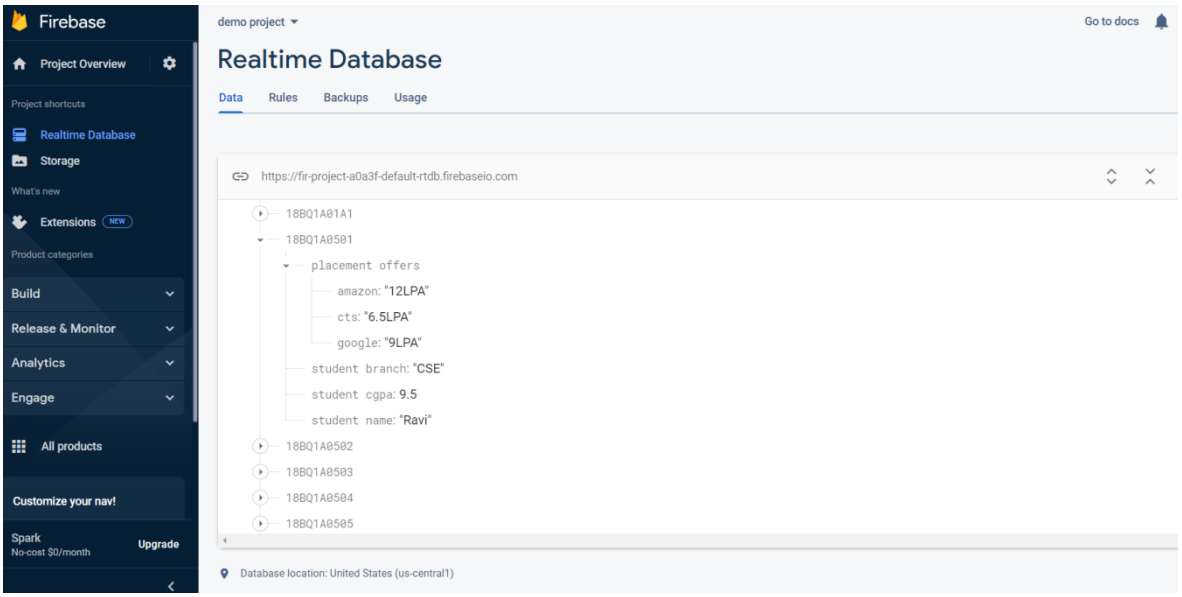
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.demoapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/vvit_icon"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/vvit_icon"
        android:supportsRtl="true"
        android:theme="@style/Theme.DemoApp">
        <activity
            android:name=".SignUpActivity"
            android:theme="@style/Theme.DemoApp.NoActionBar"
            android:exported="false"> </activity>

        <activity
            android:name=".MainActivity"
            android:theme="@style/Theme.DemoApp.NoActionBar"
            android:exported="true"> </activity>

        <activity android:name=".MainActivity5"> </activity>

        <activity
            android:name=".MainActivity4"
            android:theme="@style/Theme.AppCompat.Light.NoActionBar">
</activity>

        <activity android:name=".MainActivity3"> </activity>
```

```
        <activity
            android:name=".MainActivity2"
            android:theme="@style/Theme.DemoApp.NoActionBar"> </activity>

        <activity
            android:name=".SignInActivity"
            android:theme="@style/Theme.DemoApp.NoActionBar"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

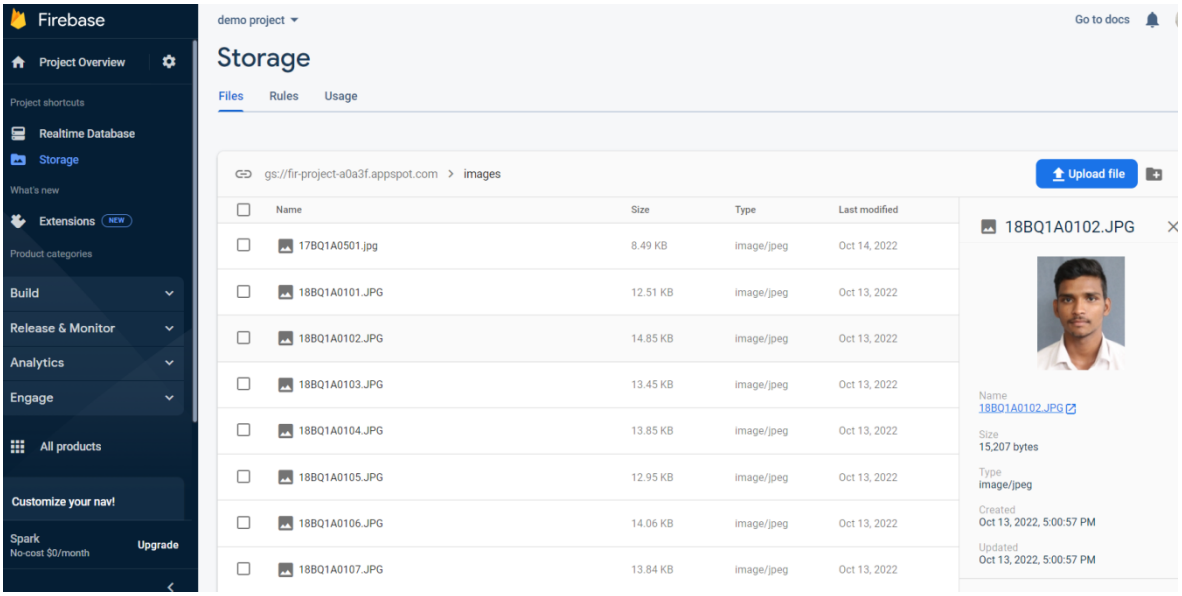
3.2 Screen Shots:-

3.2.1 *Database Structure :*

1. Real Time Database :



2. Firebase Storage :



3.2.2 Working of Application :

1. ***Login screen***: When the application is opened we get the login page. The screen will be shown as below.

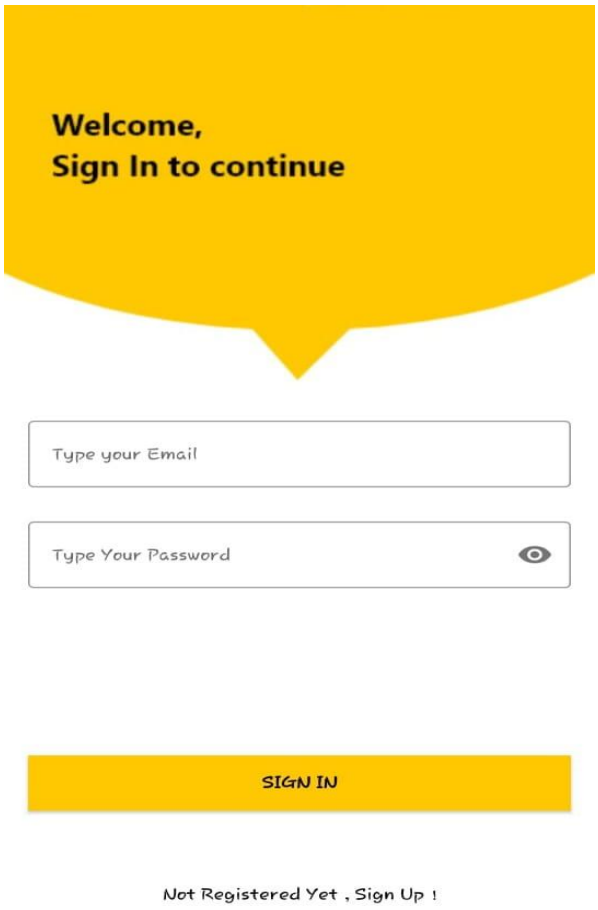


Fig 3.1 Login Screen

2. ***Sign up screen***: When the user is not a registered user then he/she can sign up in the sign up page. The screen will be shown as below.

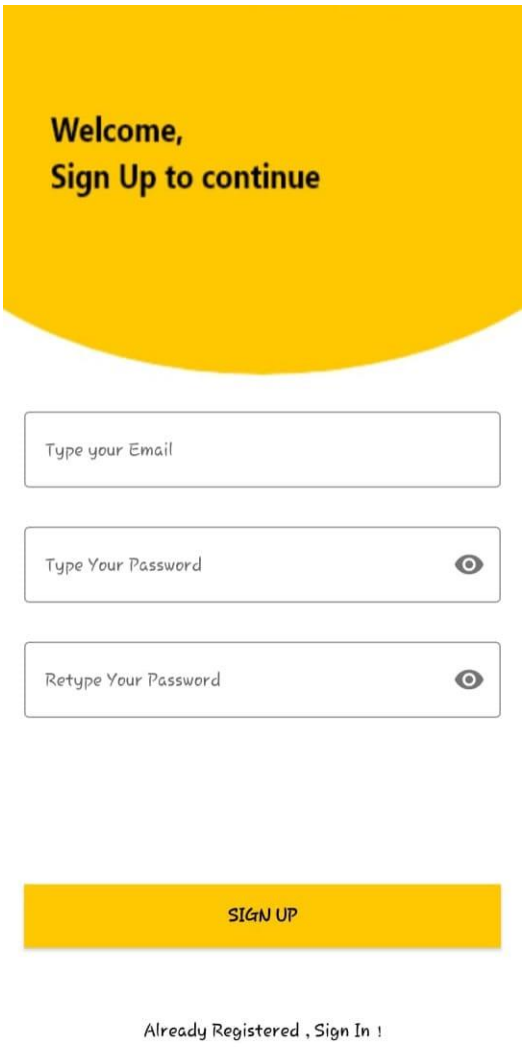


Fig 3.2 Sign up Screen

3. *Options Screen*:-After the user logged in the application, home screen will be displayed. This screen will be as shown below.

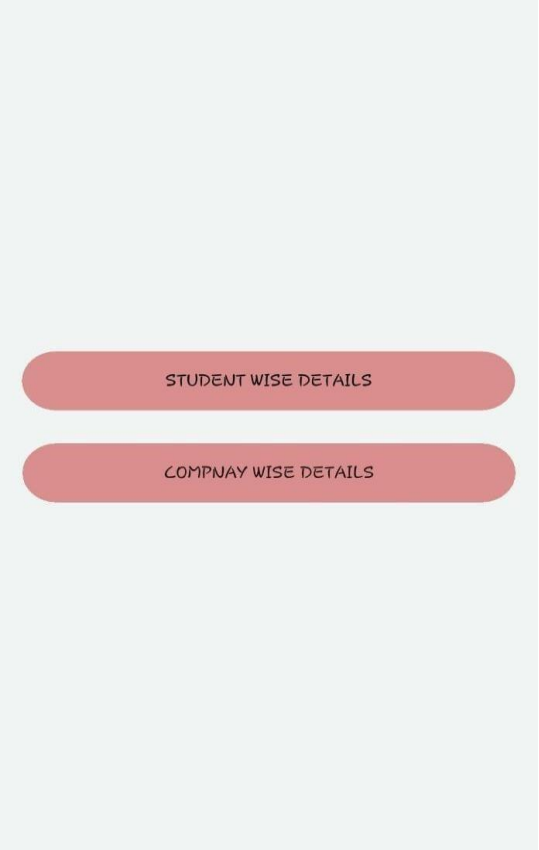


Fig 3.3 Options Screen

4. ***Entering Student roll number*:-** When Button “Student wise details” is clicked new screen will be displayed. The screen will be as shown below.

Fig 3.4 Entering Student Roll Number

5. ***Displaying student details*:-** After entering the student roll number and clicking the button “get details”, new screen will be displayed in which placement details of the student will be displayed.

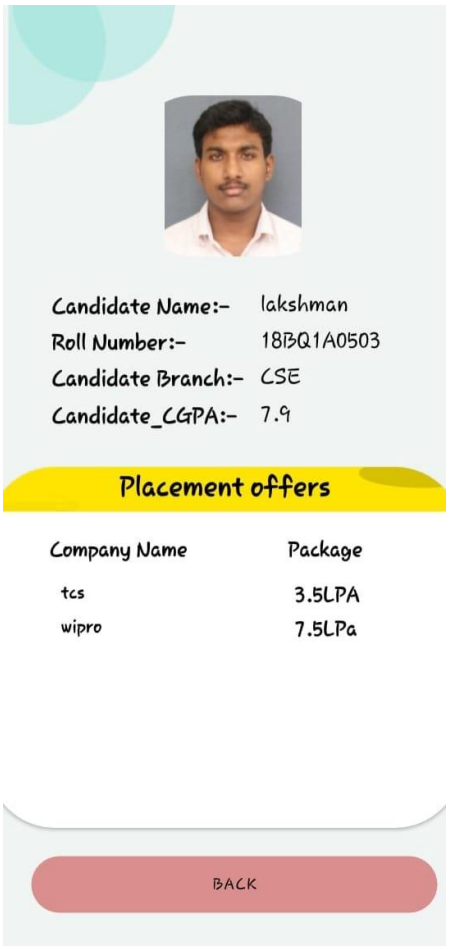


Fig 3.5 Displaying Student Details

***6. .Entering company name*:-** When Button “Company wise details” is clicked new screen will be displayed. The screen will be as shown below.

Fig 3.6 Entering Company Name

7. ***Company wise details*:-**After entering the student roll number and clicking the button “get details”, new screen will be displayed in which placement details of the student will be displayed.

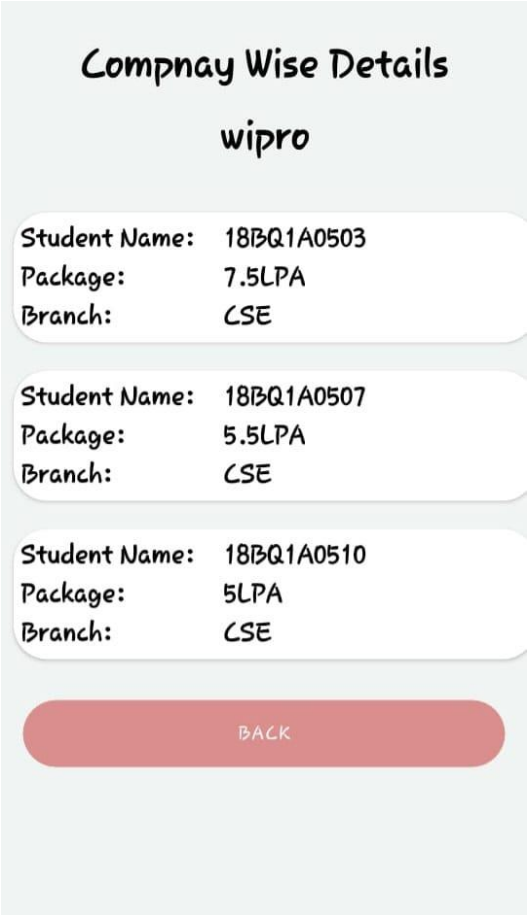


Fig 3.7 Displaying Company Wise Details

CHAPTER - 4

4.CONCLUSION

With the completion of this project, we have successfully achieved our objective of our project i.e. by maintaining the student details related to placements in an efficient manner using Android based placement management system.

Presently we designed our Training & Placement Cell to be very User Friendly. Many features are enhanced to the present Training & Placement Cell. With this Training & Placement Cell most of the time is saved. The features of the system can be further enhanced in many ways. The documentation that has enclosed can enable even a person with minimum knowledge to understand it well.

This project automates most of the activities of college placement cell. It eliminates the tedious manual repeated work done by the staff members of the placement cell. This project is aimed at developing an Android application for the Training and Placement Department of the College. This system can be used as application for the Training and Placement Officers (TPO) of the college to manage the student information with regard to placement.

BIBLIOGRAPHY:-

https://developer.android.com/courses/android-development-withkotlin/course?=badge-emailhttps://developer.android.com/reference/android/package-summaryhttps://developer.android.com/guide/topics/ui/layout/recyclerviewhttps://developer.android.com/jetpack/androidx/releases/cardviewhttps://www.youtube.com/watch?v=jDS0H9VL6bI