MAHATMA GANDHI MEMORIAL COLLEGE

UDUPI – 576102

PROJECT REPORT 2022-2023

"TRAVEL BEES APP"

DEVELOPED BY

Miss. Sahana	Reg. No: 201161522182
Miss. Shravya	Reg. No: 201161522184
Miss. Shreya	Reg.No:201161522186

Under the guidance of

Prof. Rekha N. Chandra

Department of Computer Science

Submitted to the Mangalore University in partial fulfillment of the Award of

Bachelor in Computer Application

MANGALORE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

MAHATMA GANDHI MEMORIAL COLLEGE UDUPI-576102

ACKNOWLEDGEMENT

Behind every achievement, there is a sea of gratitude to those who have activatedthis project.
The magnitude of this project demanded the co-operation, guidance and help ofmany people.
We have been fortunate enough to have this in the entire task of completion ofour project on "**Travel Bees App"**

We would like to thank our principal **Prof. Laxminarayana Karanth** for giving us opportunity to carry out our project.

We thank **Dr. M. Vishwanath Pai**, a source of inspiration and encouragement, head of the Department of Computer Science, Mahatma Gandhi MemorialCollege, Udupi for having permitted us to carry out our project work.

We are extremely grateful to express our overwhelming gratitude to our guide **Prof. Rekha N. Chandra** Lecturer of Computer Science Department, Mahatma Gandhi Memorial College Udupi for giving us valuable guidance to undertake this project.

We are indebted to all teaching and non-teaching staff members. Mahatma Gandhi Memorial College Udupi for Making this project successful.

Thanking You

Sahana

Shravya

Shreya

INDEX

CHAPTERS


TravelBees

1.INTRODUCTION

The Tours and travel Package Management System is an on page remittance web, mobile application. This application will help people from different location to hassle free payment, booking or search a package event for travelling to the registered places within very less time period. Facilities like adding packages, customer enquiry customer registration, payment and online invoice alert functionalities are available. This application is designed for the customers to get a good journey and get good memories

SYNOPSIS

2.Title of the Project:

Online Tours and Travels Package Management System – "TRAVEL BEES"

3.Introduction, Objectives, Scope of the Project:

3.1 Introduction:

The Tours and travel Package Management System is an on page remittance web, mobile application. This application will help people from different location to hassle free payment, booking or search a package event for travelling to the registered places within very less time period. Facilities like adding packages, customer enquiry customer registration, payment and online invoice alert functionalities are available. This application is designed for the customers to get a good journey and get good memories.

3.2Objectives:

- To develop a system that automates the processes and activities of travel agency.
- To develop the processes and activities of customer details.
- To provide best user interactive interface.
- To develop a system using which one can perform all operations related to travelling.

3.3Project Category:

Client-Server based application

3.4Tools/Platform

3.4.1Software Requirement Specification:

A software Requirement Specification(SRS) is the complete description about what the software will do and how the software is expected to perform. It also describes the functionality the software needs to fulfill all the user needs. It is a formal report that enables the users to review whether SRS is according to their requirements.

3.5 Modules:

3.5.1Modules used:

- **Admin**
- **User**
- **Package Module**

3.5.2Module Description:

- **Admin Authority:** This module is mainly based on admin. System will check for the User name and password for authentication. After the verification for authentication the admin can be able to precede the process. All the works are done under admins control.
- **User Authority:** This module covers the details about the registration of the user which they can be register by itself by adding data like name, password, email Id and further details. After the registration they can sign in by using User name and password.
- **Package description**: The admin can create packages by creating package page which the type, price, details, place details all the travel tour package details can be added here. Which it will be showed in user homepage.
- **Payment**: Implements payment gateway integration of an application. It provides onetime payment and recurring payment options. In this module the admin must provide the virtual invoice to the user of confirmation of their package and payment.
- **Booking confirmation/manage:** It is the process of confirming the booked packages by the admin that is booked by the user with date and comment.

4.Tools and Environment:

- Android Studio
- Google Firebase

4.1**Front End:**

Android Studio

Android Studio is the official integrated development environment (IDE) for Android application development. Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug. A project in Android Studio contains everything that defines your workspace for an app, from source code and assets to test code and build configurations.

4.2 Back End:

Google Firebase

Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, Realtime events continue to fire, giving the end user a responsive experience.

Authentication

Firebase Authentication aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users .It provides an end-to-end solution supporting email and password accounts ,phone auth and Google ,Twitter login and more .Firebase Authentication provides backend services ,easy-to-use SDKs, and readymade UI libraries to authenticate users to the app .Firebase Authentication uses Firebase Dynamic Links to send the email link to the mobile devices.

Realtime Database

The Firebase Realtime Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real-time to every connected client. It is a NoSQL database from which we can store and sync the data between our users in real-time. It uses data synchronization instead of using HTTP requests, due to this any connected device receives the updates within milliseconds. It is capable of providing all offline and online services. It ships with mobile SDKs, which allow us to build our app without the need for servers. Real-

time syncing makes it easy for our users to access their data from any device ,be it a web or mobile**.**

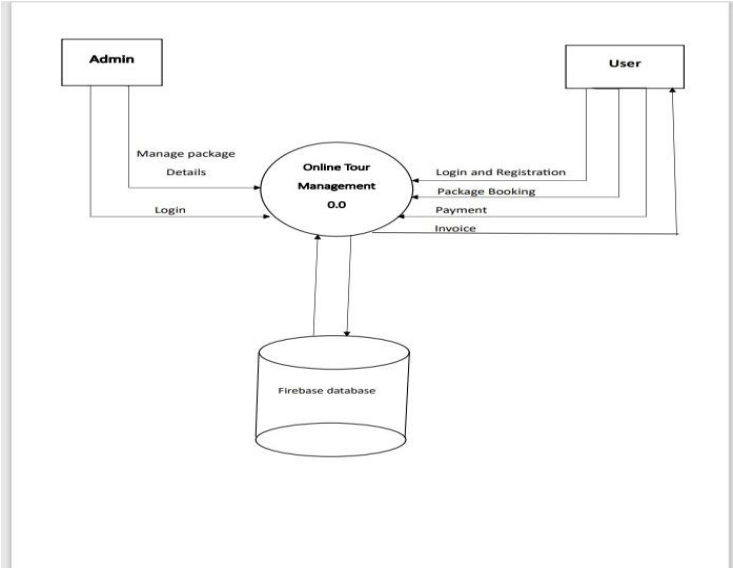
5. Hardware Requirement Specification:

RAM: Minimum 4GB or more
Hard Disk: Minimum 256GB or more
Processor: 2.0 GHz or above

6.Analysis

6.1.1Context Flow Diagram (CFD)

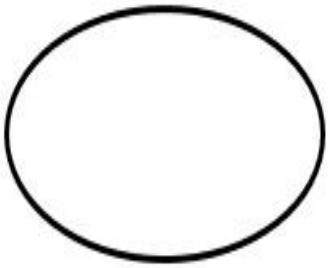
It is also known as context diagram. It's designed to be anabstraction view, showing a system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input & output data indicated by incoming or outgoing arrows.



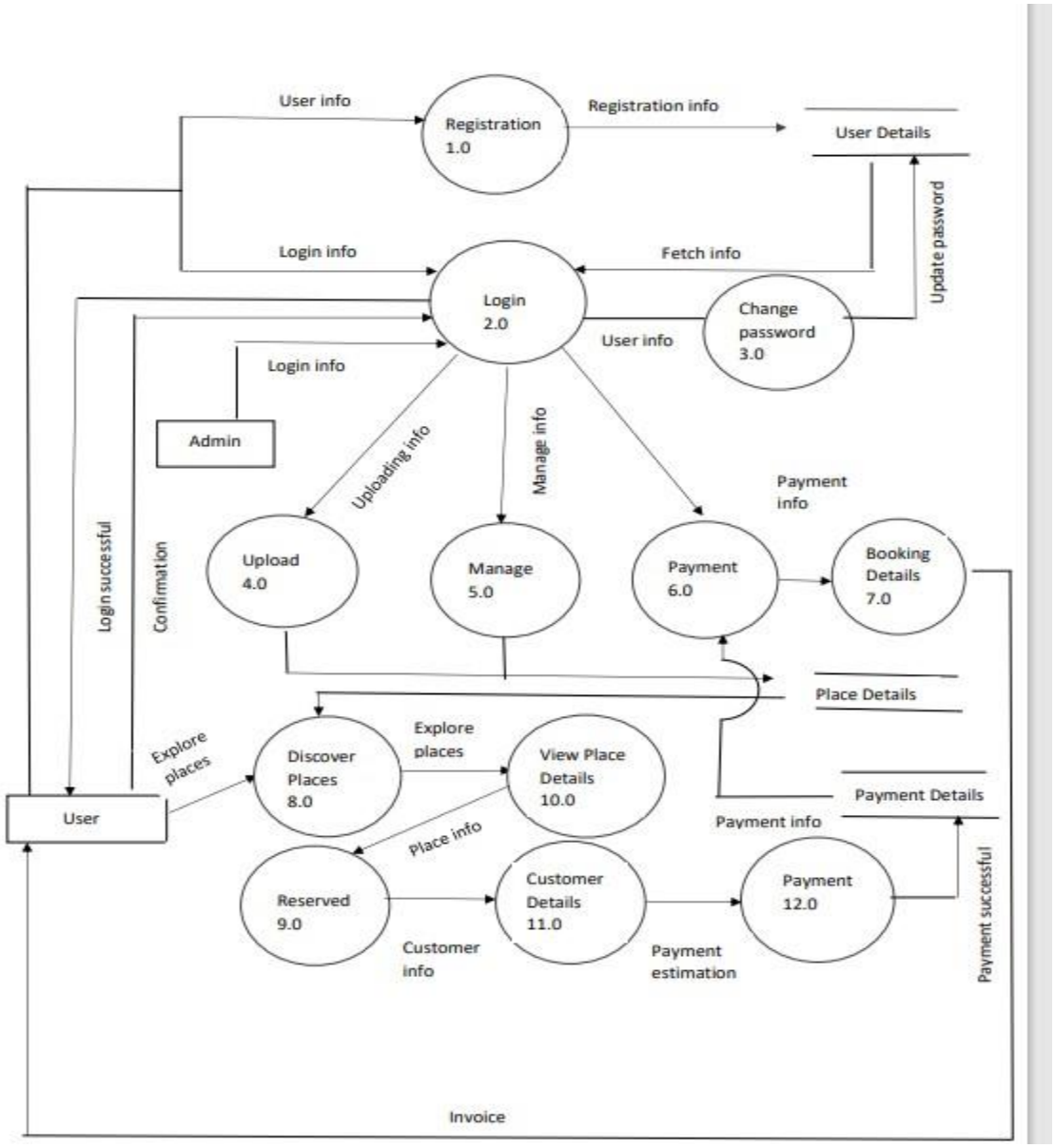
6.1.2Data Flow Diagram (DFD)

A data-flow diagram is a way of representing a flow through a process or a system(usually an information system).The DFD also provides information about the outputs and inputs of each entity and the process itself. A dataflow diagram has no control flow there are no decision rules and no loops. Levels in DFD are numbered 0,1,2 and beyond.

DFD NOTATIONS

 EXTERNAL ENTITY	Entity include source & destination of the data. Entities are represented by rectangle with their corresponding names.
 PROCESS	The task performed on data is known as process. Process is represented by circle. Somewhere round edge rectangle are also used to represent process.
 DATA FLOW	The movement of the data in the system is known as dataflow. It is represented with the help of arrow. The tail of the arrow is source and the head of the arrow is destination.
 DATA STORE	Data store includes the database of the system. It is represented by 2 parallel lines.

DATA FLOW DIAGRAM



6.1.3Data Dictionary

1. **Login @gmail + password**

- Email={0-9|A-Z|a-z}*@
- Password={0-9|A-Z|a-z|#?!@$%^&*-}.{6,}$

2 . Forgot Password @gmail + new password

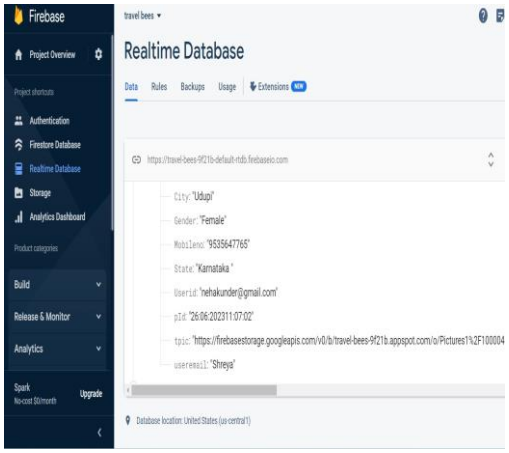
- Email={0-9|A-Z|a-z}*@
- New Password={0-9|A-Z|a-z|#?!@$%^&*-}.{6,}$

7.STRUCTURE OF SYSTEM DESIGN

Database Diagram

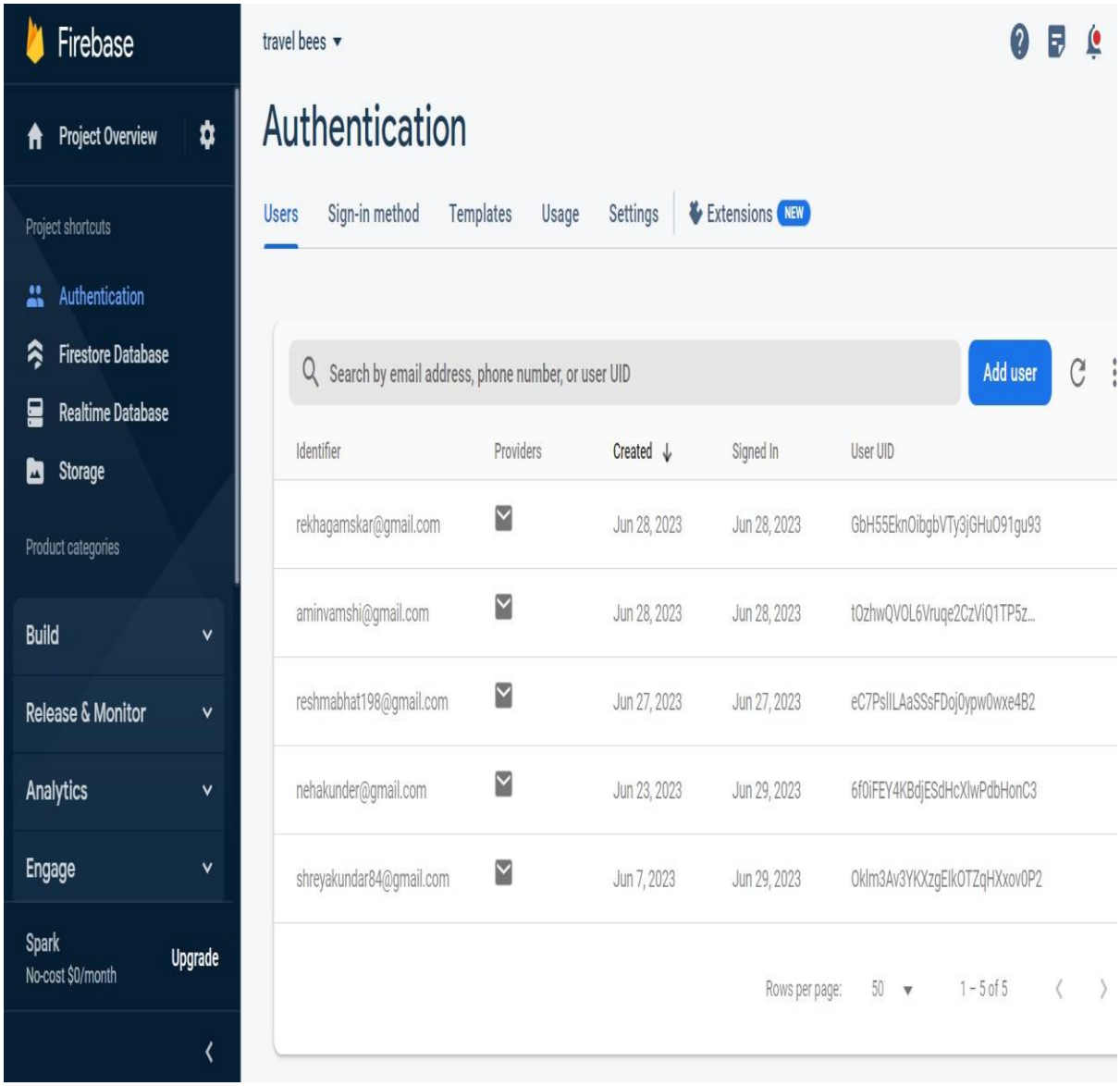
Database Design(Firebase Realtime Database)

The data is stored as JSON objects, which is not a relational database and it doesn't store data in the form of tables, rows and columns. The database is in the form of a cloud hosted json tree. The node "users" containing ID's stores the data in form of key value pairs.



Authentication

The authentication of users takes place with email and password.



8.PROGRAM CODE LISTING

JAVA CODE

REGISTER ACTIVITY

```
package com.example.projectsem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;

import android.app.ProgressDialog;
import android.content.Intent;
```

```
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Patterns;
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.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
mport com.google.firebase.auth.EmailAuthCredential;
import com.google.firebase.auth.EmailAuthProvider;
import com.google.firebase.auth.FirebaseAuth;


import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.regex.Pattern;

public class register extends AppCompatActivity {
    FirebaseAuth mAuth;
    EditText email;
    EditText pass;
    String validation="\"^(?=.*[0-9])(?=.*[a-zA-Z])(?=.*[@#$%^&+=])(?=\\\\S+$).{6,}$\"";
EditText regpass;
TextView ac;
ProgressDialog dialog;
    String em,pa;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        ac=findViewById(R.id.have_ac);
        Button btn=findViewById(R.id.btn);
        dialog=new ProgressDialog(this);
        mAuth= FirebaseAuth.getInstance();
        btn.setOnClickListener(v->{

            createuser();
        });
        ac.setOnClickListener(v->{
          startActivity(new Intent(register.this,login.class));
        });
    }


    public void createuser() {
```

```
        email = findViewById(R.id.btnlog);
        pass = findViewById(R.id.pass1);

        em = email.getText().toString();
        pa = pass.getText().toString().trim();


        if (TextUtils.isEmpty(em)) {
            email.setError("Email cannot be empty");
            email.requestFocus();
        }
    {

        else if (TextUtils.isEmpty(pa)) {
            pass.setError("Password cannot be empty");
            pass.requestFocus();
        }
        else if(!isPasswordValid(pa)){
            Toast.makeText(register.this,"Password should contain 6 character with one uppercase
letter ,one digit and one special character",Toast.LENGTH_LONG).show();
        }
        else {

            mAuth.createUserWithEmailAndPassword(em, pa).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {

                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        Toast.makeText(register.this, "User Registered successfully",
Toast.LENGTH_LONG).show();
                        dialog.setTitle("Sign Up");
                         dialog.setMessage("Signing up");
                        dialog.setCanceledOnTouchOutside(false);
                        dialog.show();
                        startActivity(new Intent(register.this, onboarding1.class));
                    } else {
                        String e = String.valueOf(task.getException());
                        Toast.makeText(register.this, "Failed" + e, Toast.LENGTH_SHORT).show();
                    }
                }
            });
        }
        }
        boolean isPasswordValid (String pa){
              Pattern pattern = Pattern.compile("^(?=.*[A-Z])(?=.*[0-
9])(?=.*[@#$%^&+=])(?=\\S+$).{6,}$");
            return pattern.matcher(pa).matches();
        }
    }
```

LOGIN ACTIVITY

```
package com.example.projectsem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Patterns;
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.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.ktx.Firebase;

public class login extends AppCompatActivity {
FirebaseAuth mAuth;
ProgressDialog p;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
p=new ProgressDialog(this);
Button btnn = findViewById(R.id.logbtn);
Button btnn2 = findViewById(R.id.btnn2);
TextView forgot = findViewById(R.id.forgottxt);
mAuth = FirebaseAuth.getInstance();
btnn.setOnClickListener(v -> {
loginuser();
});
btnn2.setOnClickListener(v -> {
startActivity(new Intent(login.this, register.class));
});
forgot.setOnClickListener(v -> {
startActivity(new Intent(login.this, Forgot_activity.class));
});
}
private void loginuser() {
EditText email1 = findViewById(R.id.btnlog);
EditText pass1 = findViewById(R.id.pass1);
String em = email1.getText().toString();
String pa = pass1.getText().toString();
p=new ProgressDialog(this);
if (TextUtils.isEmpty(em)) {
```

```
email1.setError("Email cannot be empty");
email1.requestFocus();
} else if (TextUtils.isEmpty(pa)) {
pass1.setError("Password cannot be empty");
pass1.requestFocus();
}
else if(em.equalsIgnoreCase("shreyakundar84@gmail.com"))
{
mAuth.signInWithEmailAndPassword(em,pa).addOnCompleteListener(new
OnCompleteListener<AuthResult>()
{
@Override
public void onComplete(@NonNull Task<AuthResult> task)
{
if(task.isSuccessful()){
Intent i=new Intent(login.this,dash.class);
startActivity(i);
p.setMessage("Login");
p.show();
p.dismiss();
}
else{
Toast.makeText(login.this, "Incorrect password", Toast.LENGTH_SHORT).show();
}
}
});
}else {
mAuth.signInWithEmailAndPassword(em, pa).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
@Override
public void onComplete(@NonNull Task<AuthResult> task) {
if (task.isSuccessful()) {
p.setMessage("Login");
p.show();
Toast.makeText(login.this, "User logged in successfully", Toast.LENGTH_LONG).show();
startActivity(new Intent(login.this, onboarding1.class));
} else {
Toast.makeText(login.this, "Invalid username or password", Toast.LENGTH_LONG).show();
}
}
});
}}
}
```

FORGOT PASSWORD ACTIVITY

```
package com.example.projectsem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
```

```
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;

public class Forgot_activity extends AppCompatActivity {
FirebaseAuth mAuth;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_forgot);
EditText emailforgot=findViewById(R.id.emailforgot);
Button buttonforgot=findViewById(R.id.buttonforgot);
ProgressBar p=findViewById(R.id.progressBar);
mAuth= FirebaseAuth.getInstance();
buttonforgot.setOnClickListener(v->{
String email=emailforgot.getText().toString().trim();
if(email.isEmpty()){
emailforgot.setError("Email is required");
emailforgot.requestFocus();
return;

}

if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
emailforgot.setError("Please enter valid email");
emailforgot.requestFocus();
return;
}
p.setVisibility(View.VISIBLE);
mAuth.sendPasswordResetEmail(email).addOnCompleteListener(new
OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if(task.isSuccessful()){
Toast.makeText(Forgot_activity.this, "Check your email to reset your password",
Toast.LENGTH_SHORT).show();
}
else
{
Toast.makeText(Forgot_activity.this, "Try again ,something wrong happened!",
Toast.LENGTH_SHORT).show();
}
}
});
});
}
```

```
}
```

DASHBOARD ACTIVITY

```
package com.example.projectsem;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import com.google.firebase.auth.FirebaseAuth;

public class dash extends AppCompatActivity {
FirebaseAuth mAuth;
CardView c1,c2,c3,c4;
ProgressDialog p;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_dash);
p=new ProgressDialog(this);
c1=findViewById(R.id.card1);
c2=findViewById(R.id.card2);
c3=findViewById(R.id.card3);
c4=findViewById(R.id.card4);
c1.setOnClickListener(v->{
startActivity(new Intent(dash.this,MainPage.class));
});
c3.setOnClickListener(v->{
startActivity(new Intent(dash.this,allcustrecycler.class));
});
c2.setOnClickListener(v->{
startActivity(new Intent(dash.this,imageupload.class));
});
c4.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View view){
p.setMessage("Logging out");
p.setCanceledOnTouchOutside(false);
p.show();
mAuth.signOut();
p.dismiss();
Intent j=new Intent(dash.this,login.class);
startActivity(j);
}
});
 }}
```

PROFILE ACTVITY

```
package com.example.projectsem;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.text.Editable;
import android.text.TextUtils;
import android.text.TextWatcher;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;

import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.UserProfileChangeRequest;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import de.hdodenhof.circleimageview.CircleImageView;

public class Profile extends AppCompatActivity {
    EditText fullname,fullmobile,fullcity,fullstate,fulladdress;
    Button save;
    FirebaseAuth mAuth;
    String afullname,afullmobile,afullcity,afullstate,afulladdress,gen;
    DatabaseReference databaseReference;
    private static final int galleryPic=1;
    private Uri ImageUri;
```

```
    CircleImageView img;
    RadioGroup radioGenderGroup;
    RadioButton radioGenderButton;
    FirebaseUser firebaseUser;
     private StorageReference homepic;
     DatabaseReference post2;
     ProgressDialog pg;

     String saveCurrentDate,currentTime,randomKey,saveCurrentTime;
     String downloadUri;
     @Override
     protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.profile);
        fullname = findViewById(R.id.fullname);
        fullmobile = findViewById(R.id.fullmobile);

        fullcity = findViewById(R.id.fullcity);
        fulladdress=findViewById(R.id.fulladdress);
        img=findViewById(R.id.imageViewprofile);
        fullstate = findViewById(R.id.fullstate);
        radioGenderGroup=findViewById(R.id.radioGender1);

        save= findViewById(R.id.savebtn);
        fullmobile.addTextChangedListener(new TextWatcher() {
           @Override
           public void beforeTextChanged(CharSequence s, int start, int count, int after) {

           }

           @Override
           public void onTextChanged(CharSequence s, int start, int before, int count) {
              if(validatemobile(fullmobile.getText().toString())){
//save.setEnabled(true);

              }
              else
              {
                 fullmobile.setError("Invalid mobile number");
              }
           }

           @Override
           public void afterTextChanged(Editable s) {

           }
        });
       databaseReference= FirebaseDatabase.getInstance().getReference().child("adminupload");
        post2=FirebaseDatabase.getInstance().getReference().child("ProfilePost");
        homepic= FirebaseStorage.getInstance().getReference().child("Pictures1");
        pg=new ProgressDialog(this);
        img.setOnClickListener(v->{
           uploadgallery();
```

```
        });
        save.setOnClickListener(v->{
            collectData();
        });
    }
    boolean validatemobile(String input){
        Pattern p=Pattern.compile("[6-9][0-9]{9}");
        Matcher m=p.matcher(input);
        return m.matches();
    }

    private void uploadgallery() {
        Intent i=new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(i,galleryPic);
    }

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

if(requestCode==galleryPic&&resultCode==RESULT_OK&&data!=null&&data.getData()!=nul
l){
            ImageUri=data.getData();
            img.setImageURI(ImageUri);
        }
    }

    public void collectData(){
        int selectedId=radioGenderGroup.getCheckedRadioButtonId();
        radioGenderButton=findViewById(selectedId);
        gen=radioGenderButton.getText().toString();
        afullname=fullname.getText().toString();
        afullmobile=fullmobile.getText().toString();
        afullcity=fullcity.getText().toString();
        afullstate=fullstate.getText().toString();
        afulladdress=fulladdress.getText().toString();
        if(TextUtils.isEmpty(afullname))
        {
            fullname.setError("Enter name");
            fullname.requestFocus();
        }
        else if(TextUtils.isEmpty(afullmobile))
        {
            fullmobile.setError("Enter mobile number");
            fullmobile.requestFocus();
        }
        else if(afullcity.isEmpty()){
            fullcity.setError(" Enter city");
        }
        else if(afullstate.isEmpty()){
            fullstate.setError("Enter state");
        }
```

```
        else if(afulladdress.isEmpty())
        {
            fulladdress.setError("Enter address");
        }
        else
            storeImageData();
    }

    private void storeImageData(){
        pg.setMessage("Saving..");
        pg.show();
        Calendar calendar= Calendar.getInstance();
        SimpleDateFormat currentDate=new SimpleDateFormat("mm:dd:yyyy");
        saveCurrentDate = currentDate.format(calendar.getTime());
        SimpleDateFormat currentTime=new SimpleDateFormat("HH:MM:SS");
        saveCurrentTime = currentTime.format(calendar.getTime());
        randomKey=saveCurrentDate+saveCurrentTime;
        if(randomKey==null){
            saveCurrentDate=currentDate.format(calendar.getTime());
            saveCurrentDate=currentTime.format(calendar.getTime());
        }
        //randomKey
        //String filename=System.currentTimeMillis() +".jpg";
        final StorageReference
file=homepic.child(ImageUri.getLastPathSegment()+randomKey+".jpg");
        final UploadTask uploadtask=file.putFile(ImageUri);
        uploadtask.addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                String message=e.toString();
                Toast.makeText(Profile.this, "Failed to upload image"+message,
Toast.LENGTH_SHORT).show();
            }
        }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                Toast.makeText(Profile.this, "Image uploaded sucessfully",
Toast.LENGTH_SHORT).show();
                Task<Uri> urlTask=uploadtask.continueWithTask(new
Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {
                    @Override
                    public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws
Exception {
                        if(!task.isSuccessful()){
                            throw task.getException();
                        }
                        downloadUri=file.getDownloadUrl().toString();

                        return file.getDownloadUrl();


                        //return null;
                    }
```

```
            }).addOnCompleteListener(new OnCompleteListener<Uri>() {
                @Override
                public void onComplete(@NonNull Task<Uri> task) {
                    if(task.isSuccessful()){
                      // firebaseUser=mAuth.getCurrentUser();
                       downloadUri=task.getResult().toString();
                       Toast.makeText(Profile.this, "Saved", Toast.LENGTH_SHORT).show();
                       UpdateDatabase();

                    }
                }
            });
        }
    });
}
private void UpdateDatabase(){
    String userId=getCurrentUserId();
    HashMap<String,Object> map=new HashMap<>();
    map.put("pId",randomKey);
    map.put("useremail",afullname);
    map.put("Mobileno",afullmobile);
    map.put("Gender",gen);
    map.put("City",afullcity);
    map.put("Address",afulladdress);
    map.put("tpic",downloadUri);
    map.put("State",afullstate);
    map.put("Userid",userId);
    post2.push().setValue(map).addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()) {
                pg.dismiss();
                Toast.makeText(Profile.this, "Sucessfull", Toast.LENGTH_SHORT).show();
                startActivity(new Intent(Profile.this,SplashScreen.class));
            } else {
                pg.dismiss();
                Toast.makeText(Profile.this, "Failed" + task.getException().toString(),
Toast.LENGTH_SHORT).show();
            }
        }

    });


}
public String getCurrentUserId() {
    mAuth = FirebaseAuth.getInstance();
    firebaseUser = mAuth.getCurrentUser();
    if (firebaseUser != null) {
        return firebaseUser.getEmail();
    } else {
        return null;
    }
```

```
}
}
```

HOME ACTVITY

```
package com.example.projectsem;

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

import com.example.projectsem.homeInFeed.homesInFeed;
import com.example.projectsem.homeInFeed.homesmain2;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.squareup.picasso.Picasso;

public class Mainpage2 extends AppCompatActivity {
FirebaseAuth mAuth;
private DatabaseReference HomeRef;
private RecyclerView recyclerView;
RecyclerView.LayoutManager layoutManager;
private FirebaseAuth auth;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_mainpage2);
recyclerView =findViewById(R.id.recycler2);
recyclerView.setHasFixedSize(true);
layoutManager = new LinearLayoutManager( this);
recyclerView.setLayoutManager(layoutManager);//"layoutManager"
mAuth = FirebaseAuth.getInstance();
HomeRef = FirebaseDatabase.getInstance().getReference().child("TravelPost");
auth= FirebaseAuth.getInstance();
}
@Override
protected void onStart() {
//      Toast.makeText(this, "hello", Toast.LENGTH_SHORT).show();
super.onStart();
FirebaseRecyclerOptions<Model> option = new
FirebaseRecyclerOptions.Builder<Model>().setQuery(HomeRef,Model.class).build();
```

```
FirebaseRecyclerAdapter<Model, homesmain2> adapter=new FirebaseRecyclerAdapter<Model,
homesmain2>(option) {
@Override
protected void onBindViewHolder(@NonNull homesmain2 holder, int position, @NonNull
Model model) {
holder.txtplacemain2.setText(model.getTplace());
//          holder.txtfees.setText("hello");
holder.txtfeesmain2.setText(model.getTamnt());

// holder.txtfees.setText(model.getAmountt());
//          String imageUri=null;
//          imageUri=model.getTpic();
//          Picasso.get().load(imageUri).into(holder.HIFpic);

//Picasso.get().load("https://firebasestorage.googleapis.com/v0/b/travel-bees-
9f21b.appspot.com/o/Pictures%2F100004549527%3A22%3A202315%3A06%3A65.jpg?alt=me
dia&token=36a8ee44-0051-4f7d-9fd5-b7e39f8a58c0").into(holder.HIFpic);
Picasso.get().load(model.getTpic()).into(holder.HIFpicmain2);
holder.itemView.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent=new Intent(Mainpage2.this,placedetails.class);
intent.putExtra("pId", model.getpId());
startActivity(intent);
}
});
}
@NonNull
@Override
public travelsmain2 onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_main2view, parent,
false);
travelsmain2 holder = new travelsmain2(view);
return holder;

}
};
recyclerView.setAdapter(adapter);
adapter.startListening();
}}
```

HOMEADAPTER ACTVITY

```
package com.example.projectsem.homeInFeed;

import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
```

```
import com.example.projectsem.R;
import com.example.projectsem.interfaces.homesInFeedInterface;

public class travelInFeed extends RecyclerView.ViewHolder implements View.OnClickListener
{
   public TextView txtplace,txtfees;
   public homesInFeedInterface listener;
   public ImageView HIFpic,update,delete;


   public travelInFeed(@NonNull View itemView) {
      super(itemView);
      txtplace=itemView.findViewById(R.id.txtplace);
      txtfees= itemView.findViewById(R.id.txtfees);
      HIFpic=itemView.findViewById(R.id.imageView);
      update=itemView.findViewById(R.id.update);
      delete=itemView.findViewById(R.id.delete);
   }

   public void setItemClickListener(homesInFeedInterface listener)
{
   this.listener=listener;
}

   @Override
   public void onClick(View view) {
      listener.onClick(view,getAdapterPosition(),false);
   }
}
```

PLACEDETAILS ACTVITY

```
package com.example.projectsem;

import android.annotation.SuppressLint;
import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.text.TextUtils;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;
import com.google.android.gms.tasks.Continuation;
```

```
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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 com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Objects;

public class placedetails extends AppCompatActivity {
private TextView Aplace,Ades,Aactivities,Aamount,Ameals;
private ImageView HDplacePic;
String placeID="",imageurl;
AppCompatButton book;
public static String key;
EditText Auname,Amobile,Anumperson,Acheckin;
String realpic;
FirebaseAuth mAuth;
FirebaseUser firebaseUser;
String adplace,addesc,adam,adacti,adMeals,aduname,admobile,adnumperson,adcheckin;
DatabaseReference databaseReference;
private static final int galleryPic=1;
private Uri ImageUri;
private StorageReference homepic;
DatabaseReference post1;
ProgressDialog pg;
String saveCurrentDate,currentTime,randomKey,saveCurrentTime;
String downloadUri;

@SuppressLint("MissingInflatedId")
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_placedetails);
Aplace=findViewById(R.id.addname);
Ades=findViewById(R.id.adddes);
Aactivities=findViewById(R.id.addactivities);
Aamount=findViewById(R.id.addamount);
Ameals=findViewById(R.id.addmeals);
HDplacePic=findViewById(R.id.placeimg);
Auname=findViewById(R.id.adduname);
Amobile=findViewById(R.id.addcmble);
```

```
Anumperson=findViewById(R.id.addperson);
Acheckin=findViewById(R.id.cdateformat);
book=findViewById(R.id.book);
post1=FirebaseDatabase.getInstance().getReference().child("BookPost");
FirebaseStorage.getInstance().getReference().child("BookPictures");
pg=new ProgressDialog(this);
placeID=getIntent().getStringExtra("pId");
getPlaceDetails(placeID);
book.setOnClickListener(v-> {
aduname=Auname.getText().toString();
admobile=Amobile.getText().toString();
adnumperson=Anumperson.getText().toString();
adcheckin=Acheckin.getText().toString();
if(TextUtils.isEmpty(aduname))
{
Auname.setError("Fill your name to proceed");
}
else if(TextUtils.isEmpty(admobile))
{
Amobile.setError("Fill your mobile to proceed");
}
else if(TextUtils.isEmpty(adnumperson))
{
Anumperson.setError("Fill number of persons to proceed");
}
else if(TextUtils.isEmpty(adcheckin))
{
Acheckin.setError("Fill your checkin date to proceed");
}
else {
Model model = new Model();
Intent intent = new Intent(placedetails.this, razro_pay.class);
intent.putExtra("pId", model.getpId());
startActivity(intent);
}
public String getCurrentUserId(){
mAuth=FirebaseAuth.getInstance();
firebaseUser=mAuth.getCurrentUser();
if(firebaseUser!=null){
return firebaseUser.getEmail();
}
else
{
return null;
}
}
private void getPlaceDetails(String placeID)
{
Toast.makeText(this, "toast msg", Toast.LENGTH_SHORT).show();
DatabaseReference homeRef =
FirebaseDatabase.getInstance().getReference().child("TravelPost");
homeRef.addValueEventListener(new ValueEventListener() {
@Override
```

```
public void onDataChange(@NonNull DataSnapshot snapshot) {
for(DataSnapshot dataSnapshot:snapshot.getChildren()) {
if (dataSnapshot.child("pId").getValue() != null) {
if (Objects.requireNonNull(dataSnapshot.child("pId").getValue()).toString().equals(placeID)) {
Model tour = dataSnapshot.getValue(Model.class);
Aplace.setText(tour.getTplace());
Ades.setText(tour.getTdesc());
Aactivities.setText(tour.getTacti());
Aamount.append(tour.getTamnt());
Picasso.get().load(tour.getTpic()).into(HDplacePic);
break;
}
}
Toast.makeText(placedetails.this, "" + dataSnapshot.child("pId").getValue(),
Toast.LENGTH_SHORT).show();
}}
@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});
}
```

UPLOAD ACTIVITY

```
package com.example.projectsem;

import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
```

```
import com.google.firebase.storage.UploadTask;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;

public class imageupload extends AppCompatActivity {
EditText placename,description,amount,activities;
Button upload;
FirebaseAuth mAuth;
String place,desc,am,acti;
DatabaseReference databaseReference;
private static final int galleryPic=1;
private Uri ImageUri;
ImageView img;
private StorageReference homepic;
DatabaseReference post;
ProgressDialog pg;
String saveCurrentDate,currentTime,randomKey,saveCurrentTime;
String downloadUri;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_imageupload);
placename = findViewById(R.id.enterplace);
description = findViewById(R.id.enterdes);
amount = findViewById(R.id.enteramount);
img=findViewById(R.id.imageView);
activities = findViewById(R.id.enteractivities);
upload = findViewById(R.id.upload_btn);
databaseReference= FirebaseDatabase.getInstance().getReference().child("adminupload");
post=FirebaseDatabase.getInstance().getReference().child("TravelPost");
homepic= FirebaseStorage.getInstance().getReference().child("Pictures");
pg=new ProgressDialog(this);
img.setOnClickListener(v->{
uploadgallery();
});
upload.setOnClickListener(v->{
collectData();
});
}

private void uploadgallery() {
Intent i=new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
startActivityForResult(i,galleryPic);
}

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

```
if(requestCode==galleryPic&&resultCode==RESULT_OK&&data!=null&&data.getData()!=nul
l){
ImageUri=data.getData();
img.setImageURI(ImageUri);
}
}

public void collectData(){
place=placename.getText().toString();
desc=description.getText().toString();
am=amount.getText().toString();
acti=activities.getText().toString();

if(place.isEmpty())
{
placename.setError("Place cannot be empty");
}
else if(desc.isEmpty())
{
description.setError("Description cannot be empty");
}
else if(am.isEmpty())
{
amount.setError("Amount cannot be empty");
}
else if(acti.isEmpty()){
activities.setError("Activities cannot be empty");
}
else
storeImageData();
}

private void storeImageData(){
pg.setMessage("Uploading");
pg.show();
Calendar calendar= Calendar.getInstance();
SimpleDateFormat currentDate=new SimpleDateFormat("mm:dd:yyyy");
saveCurrentDate = currentDate.format(calendar.getTime());
SimpleDateFormat currentTime=new SimpleDateFormat("HH:MM:SS");
saveCurrentTime = currentTime.format(calendar.getTime());
randomKey=saveCurrentDate+saveCurrentTime;
if(randomKey==null){
saveCurrentDate=currentDate.format(calendar.getTime());
saveCurrentDate=currentTime.format(calendar.getTime());
}

final StorageReference
file=homepic.child(ImageUri.getLastPathSegment()+randomKey+".jpg");

final UploadTask uploadtask=file.putFile(ImageUri);
uploadtask.addOnFailureListener(new OnFailureListener() {
@Override
public void onFailure(@NonNull Exception e) {
```

```
String message=e.toString();
Toast.makeText(imageupload.this, "Failed to upload image"+message,
Toast.LENGTH_SHORT).show();
}
}).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
@Override
public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
Toast.makeText(imageupload.this, "Image uploaded sucessfully",
Toast.LENGTH_SHORT).show();
Task<Uri> urlTask=uploadtask.continueWithTask(new
Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {
@Override
public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {
if(!task.isSuccessful()){
throw task.getException();
}
downloadUri=file.getDownloadUrl().toString();
return file.getDownloadUrl();

}
}).addOnCompleteListener(new OnCompleteListener<Uri>() {
@Override
public void onComplete(@NonNull Task<Uri> task) {
if(task.isSuccessful()){
downloadUri=task.getResult().toString();
Toast.makeText(imageupload.this, "Done", Toast.LENGTH_SHORT).show();
UpdateDatabase();
}
}
});
}
});
}
private void UpdateDatabase(){
HashMap<String,Object>map=new HashMap<>();
map.put("pId",randomKey);
map.put("tplace",place);
map.put("tdesc",desc);
map.put("tamnt",am);
map.put("tacti",acti);
map.put("tpic",downloadUri);
post.push().setValue(map).addOnCompleteListener(new OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if (task.isSuccessful()) {
pg.dismiss();
Toast.makeText(imageupload.this, "Sucessfull", Toast.LENGTH_SHORT).show();
startActivity(new Intent(imageupload.this,MainPage.class));
} else {
pg.dismiss();
Toast.makeText(imageupload.this, "Failed" + task.getException().toString(),
Toast.LENGTH_SHORT).show();
}
```

```
}

});
}
```

MANAGE ACTIVITY

```
package com.example.projectsem;

import android.content.DialogInterface;
import android.content.Intent;
```

```
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.projectsem.homeInFeed.homesInFeed;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.orhanobut.dialogplus.DialogPlus;
import com.orhanobut.dialogplus.ViewHolder;
import com.squareup.picasso.Picasso;

import java.util.HashMap;
import java.util.Map;

public class MainPage extends AppCompatActivity {
FirebaseAuth mAuth;
private DatabaseReference HomeRef;
private RecyclerView recyclerView;
RecyclerView.LayoutManager layoutManager;
private FirebaseAuth auth;
private DatabaseReference databaseReference;
FloatingActionButton fb;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main_page);
recyclerView =findViewById(R.id.recycler);
recyclerView.setHasFixedSize(true);
layoutManager = new LinearLayoutManager( this);
recyclerView.setLayoutManager(layoutManager);//"layoutManager"
mAuth = FirebaseAuth.getInstance();
```

```
HomeRef = FirebaseDatabase.getInstance().getReference().child("TravelPost");
auth= FirebaseAuth.getInstance();

@Override
protected void onStart() {

super.onStart();
FirebaseRecyclerOptions<Model> option = new
FirebaseRecyclerOptions.Builder<Model>().setQuery(HomeRef,Model.class).build();
FirebaseRecyclerAdapter<Model, homesInFeed>adapter=new FirebaseRecyclerAdapter<Model,
homesInFeed>(option) {
@Override
protected void onBindViewHolder(@NonNull homesInFeed holder, int position, @NonNull
Model model) {
holder.txtplace.setText(model.getTplace());
holder.txtfees.setText(model.getTamnt());

Picasso.get().load(model.getTpic()).into(holder.HIFpic);
holder.itemView.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent=new Intent(MainPage.this,placedetails.class);
intent.putExtra("pId", model.getpId());
startActivity(intent);
}
});
holder.update.setOnClickListener((view)->{
final DialogPlus dialogPlus=
DialogPlus.newDialog(holder.HIFpic.getContext()).setContentHolder(new
ViewHolder(R.layout.dialog_content)).setExpanded(true,1000).create();

View myview=dialogPlus.getHolderView();
final EditText enterplaceedit=myview.findViewById(R.id.enterplaceedit);
final EditText enteramountedit=myview.findViewById(R.id.enteramountedit);
final EditText enterdesedit=myview.findViewById(R.id.enterdesedit);
final EditText enteractivitiesedit=myview.findViewById(R.id.enteractivitiesedit);
Button updateedit=myview.findViewById(R.id.upload_btnedit);

enterplaceedit.setText(model.getTplace());
enterdesedit.setText(model.getTdesc());
enteractivitiesedit.setText(model.getTacti());
enteramountedit.setText(model.getTamnt());
dialogPlus.show();

updateedit.setOnClickListener((v)->{
Map<String,Object>map=new HashMap<>();
map.put("tplace",enterplaceedit.getText().toString());
```

```
map.put("tdesc",enterdesedit.getText().toString());
map.put("tamnt",enteramountedit.getText().toString());
map.put("tacti",enteractivitiesedit.getText().toString());

FirebaseDatabase.getInstance().getReference().child("TravelPost").child(getRef(position).getKey
()).updateChildren(map).addOnSuccessListener(new OnSuccessListener<Void>() {
@Override
public void onSuccess(Void unused) {
dialogPlus.dismiss();
}
}).addOnFailureListener((e)->{
dialogPlus.dismiss();
});
});

});
holder.delete.setOnClickListener((view)->{
AlertDialog.Builder builder=new AlertDialog.Builder(holder.HIFpic.getContext());
builder.setTitle("Delete package");
builder.setMessage("Delete...?");
builder.setPositiveButton("Yes",(dialogInterface,i)->{
FirebaseDatabase.getInstance().getReference().child("TravelPost").child(getRef(position).getKey
()).removeValue();
});
builder.setNegativeButton("No", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {

}
});
builder.show();
});

}

@NonNull
@Override
public homesInFeed onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_mainview, parent,
false);
homesInFeed holder = new homesInFeed(view);
return holder;

}
};
recyclerView.setAdapter(adapter);
```

```
adapter.startListening();
}
}
```

PAYMENT ACTIVITY

```
package com.example.projectsem;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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 com.razorpay.Checkout;
import com.razorpay.PaymentResultListener;
import com.squareup.picasso.Picasso;
import org.json.JSONObject;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Objects;
import kotlinx.coroutines.channels.ChannelResult;

public class razro_pay extends AppCompatActivity  implements PaymentResultListener {
Button paybn;
TextView paystat;
TextView textView3;
String placeId = "", imageurl;
FirebaseAuth mAuth;
FirebaseUser firebaseUser;
DatabaseReference post1;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.razro_pay);
Checkout.preload(getApplicationContext());
paybn = findViewById(R.id.paybtn);
```

```
textView3=findViewById(R.id.paydefault);
post1=FirebaseDatabase.getInstance().getReference().child("BookPost");
placeId = getIntent().getStringExtra("pId");
getPlaceDetails(placeId);
paybn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
PaymentNow(" 100");
}
});
}

private void getPlaceDetails(String placeID)
{
Toast.makeText(this, "toast msg", Toast.LENGTH_SHORT).show();
DatabaseReference homeRef =
FirebaseDatabase.getInstance().getReference().child("TravelPost");
homeRef.addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot snapshot) {
for(DataSnapshot dataSnapshot:snapshot.getChildren()) {
if (dataSnapshot.child("pId").getValue() != null) {
if (Objects.requireNonNull(dataSnapshot.child("pId").getValue()).toString().equals(placeID)) {
Model tour = dataSnapshot.getValue(Model.class);
Toast.makeText(razro_pay.this, ""+tour.getTplace(), Toast.LENGTH_SHORT).show();
textView3.append(tour.getTamnt());
break;
}
}
Toast.makeText(razro_pay.this, "" + dataSnapshot.child("pId").getValue(),
Toast.LENGTH_SHORT).show();
}}
@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});
}

public String getCurrentUserId() {
mAuth = FirebaseAuth.getInstance();
firebaseUser = mAuth.getCurrentUser();
if (firebaseUser != null) {
return firebaseUser.getEmail();
} else {
return null;
}
}
private void UpdateDatabase() {
String userId=getCurrentUserId();
Model model=new Model();
HashMap<String, Object> map = new HashMap<>();
map.put("pIid", model.getpId());
map.put("tplace",model.getTplace());
```

```
map.put("tamnt", model.getTamnt());
map.put("tuname",model.getTuname());
map.put("tmobile",model.getTmobile());
map.put("tnumperson",model.getTnumperson());
map.put("tcheckin",model.getTcheckin());
map.put("user",userId);
map.put("tpic", model.getTpic());
post1.push().setValue(map).addOnCompleteListener(new OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if (task.isSuccessful()) {
Toast.makeText(razro_pay.this, "Stored", Toast.LENGTH_SHORT).show();

} else {
Toast.makeText(razro_pay.this, "Failed" + task.getException().toString(),
Toast.LENGTH_SHORT).show();
}
}
});
}
private void PaymentNow(String amount) {
final Activity activity = this;
Checkout checkout = new Checkout();
checkout.setKeyID("rzp_test_M5lpO1nTJfHAxF");
checkout.setImage(R.drawable.ic_launcher_background);
double finalAmount = Float.parseFloat(amount) * 100;
try {
JSONObject options = new JSONObject();
options.put("name", "Travel bees");
options.put("description", "pay Charges");
options.put("image", "https://s3.amazonaws.com/rzp-mobile/images/rzp.jpg");
options.put("theme.color", "#3399cc");
options.put("currency", "INR");
options.put("amount", finalAmount + "");
options.put("prefill.email", "shravyabhandary124@gmail.com ");
options.put("prefill.contact", "6363976103");
checkout.open(activity, options);
} catch (Exception e) {
Log.e("TAG", "Error in the starting Razropay Checkout", e);

}


}

@Override
public void onPaymentSuccess(String s) {

Toast.makeText(getApplicationContext(), "payment Successfull!",
Toast.LENGTH_SHORT).show();
UpdateDatabase();
}
```

```
@Override
public void onPaymentError(int i, String s) {
Toast.makeText(getApplicationContext(), "payment Failed!", Toast.LENGTH_SHORT).show();
}
}
```

INVOICE ACTIVITY

```
package com.example.projectsem;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

import com.example.mail.databinding.ActivityMainBinding;

public class MainActivity extends AppCompatActivity {
ActivityMainBinding binding;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
binding=ActivityMainBinding.inflate(getLayoutInflater());
setContentView(binding.getRoot());
binding.sendbtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v){
String email=binding.emailAddress.getText().toString();
String subject=binding.subject.getText().toString();
String message=binding.travelbee.getText().toString();



String[] addresses=email.split(",");
Intent intent =new Intent(Intent.ACTION_SEND);
intent.setData(Uri.parse("mailto:"));
intent.putExtra(Intent.EXTRA_EMAIL,addresses);
intent.putExtra(Intent.EXTRA_SUBJECT,subject);
intent.putExtra(Intent.EXTRA_TEXT,message);

intent.setType("message/rfc822");
startActivity(Intent.createChooser(intent,"choose an email client"));
```

```
}

});
}
}
```

MYBOOKING ACTIVITY

```
package com.example.projectsem;

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

import com.example.projectsem.homeInFeed.homesInFeedcart;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class particularecycler extends AppCompatActivity {
FirebaseAuth mAuth;
FirebaseUser firebaseUser;
   private DatabaseReference HomeRef;
   private RecyclerView recyclerView;
   RecyclerView.LayoutManager layoutManager;
FirebaseUser firebaseuser;
   private FirebaseAuth auth;
   private DatabaseReference databaseReference;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_recycler);

      recyclerView =findViewById(R.id.recyclerview);
      recyclerView.setHasFixedSize(true);
      layoutManager = new LinearLayoutManager( this);
      recyclerView.setLayoutManager(layoutManager);//"layoutManager"
      HomeRef = FirebaseDatabase.getInstance().getReference().child("BookPost");
      auth= FirebaseAuth.getInstance();
   }
   @Override
   protected void onStart() {
```

```
        super.onStart();
String userId=getCurrentUserId();
        FirebaseRecyclerOptions<Model> option = new
FirebaseRecyclerOptions.Builder<Model>().setQuery(FirebaseDatabase.getInstance().getReferen
ce().child("BookPost").orderByChild("user").equalTo(userId),Model.class).build();
       FirebaseRecyclerAdapter<Model, homesInFeedcart> adapter=new
FirebaseRecyclerAdapter<Model, homesInFeedcart>(option) {
          @Override
          protected void onBindViewHolder(@NonNull homesInFeedcart holder, int position,
@NonNull Model model) {
              holder.txtcartplace.setText(model.getTplace());
              holder.txtcartfees.setText(model.getTamnt());
              holder.txtmail.setText(userId);
              holder.txtcheckin.setText(model.getTcheckin());
              holder.txtmobile.setText(model.getTmobile());
              holder.itemView.setOnClickListener(new View.OnClickListener() {
                 @Override
                 public void onClick(View v) {
//                    DatabaseReference userBookingref=HomeRef.child("BookPost").child(userId)
                    Intent intent = new Intent(particularecycler.this, dummy.class);
                    intent.putExtra("pIid", model.getpIid());
                    startActivity(intent);
                 }
              });

          }


          @NonNull
          @Override
          public homesInFeedcart onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
              View view= LayoutInflater.from(parent.getContext()).inflate(R.layout.item_viewcart,
parent, false);
              homesInFeedcart holder=new homesInFeedcart(view);
              return holder;
          }
       };
       recyclerView.setAdapter(adapter);
       adapter.startListening();
    }
    public String getCurrentUserId(){
       mAuth=FirebaseAuth.getInstance();
       firebaseUser=mAuth.getCurrentUser();
       if(firebaseUser!=null){
          return  firebaseUser.getEmail();
       }
       else
       {
          return  null;

       }
    }}
```

ALLCUSTOMER ACTIVITY

```
package com.example.projectsem;

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

import com.example.projectsem.homeInFeed.homesInallcart;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class allcustrecycler extends AppCompatActivity {
    FirebaseAuth mAuth;
    private DatabaseReference HomeRef;
    private RecyclerView recyclerView;
    RecyclerView.LayoutManager layoutManager;
    private FirebaseAuth auth;

    private DatabaseReference databaseReference;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_allcustrecycler);

        recyclerView =findViewById(R.id.allcust);
        recyclerView.setHasFixedSize(true);
        layoutManager = new LinearLayoutManager( this);
        recyclerView.setLayoutManager(layoutManager);//"layoutManager"
        mAuth = FirebaseAuth.getInstance();
        HomeRef =
FirebaseDatabase.getInstance().getReference().child("BookPost");
        auth= FirebaseAuth.getInstance();
    }
    @Override
    protected void onStart() {
```

```
//        Toast.makeText(this, "hello", Toast.LENGTH_SHORT).show();
        super.onStart();
        FirebaseRecyclerOptions<Model> option = new
FirebaseRecyclerOptions.Builder<Model>().setQuery(HomeRef,Model.class).buil
d();
        FirebaseRecyclerAdapter<Model, homesInallcart> adapter=new
FirebaseRecyclerAdapter<Model, homesInallcart>(option) {
            @Override
            protected void onBindViewHolder(@NonNull homesInallcart holder, int
position, @NonNull Model model) {
                holder.txtallplace.setText(model.getTplace());
//                holder.txtfees.setText("hello");
                holder.txtallfees.setText(model.getTamnt());
                holder.txtallcheckin.setText(model.getTcheckin());
holder.txtallmobile.setText(model.getTmobile());
holder.txtalluseremail.setText(model.getTuser());
                // holder.txtfees.setText(model.getAmountt());
//                String imageUri=null;
//                imageUri=model.getTpic();
//                Picasso.get().load(imageUri).into(holder.HIFpic);

//Picasso.get().load("https://firebasestorage.googleapis.com/v0/b/travel-bees-
9f21b.appspot.com/o/Pictures%2F100004549527%3A22%3A202315%3A06%3A
65.jpg?alt=media&token=36a8ee44-0051-4f7d-9fd5-
b7e39f8a58c0").into(holder.HIFpic);
            //   Picasso.get().load(model.getTpic()).into(holder.HIFpic);
                holder.itemView.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Intent intent=new Intent(allcustrecycler.this,placedetails.class);
                        intent.putExtra("pId", model.getpId());
                        startActivity(intent);
                    }
                });
                holder.button2.setOnClickListener(v->{
                startActivity(new Intent(allcustrecycler.this,mail.class));
                });
            }
            @NonNull
            @Override
            public homesInallcart onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
                View view=
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_viewall, parent,
false);
```

```
            homesInallcart holder = new homesInallcart(view);
            return holder;
        }
    };
    recyclerView.setAdapter(adapter);
    adapter.startListening();
  }
}
```

ANDROIDMANIFEST.XML

```
<?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.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.INTERNET" />

  <application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Projectsem"
    tools:targetApi="31">
    <activity
      android:name=".realprofile"
      android:exported="false">
      <meta-data
        android:name="android.app.lib_name"
        android:value="" />
    </activity>
    <activity
      android:name=".Profile"
      android:exported="true">
      <meta-data
        android:name="android.app.lib_name"
        android:value="" />
    </activity>
    <activity
      android:name=".mail"
      android:exported="true">
      <meta-data
        android:name="android.app.lib_name"
        android:value="" />
    </activity>
    <activity
```

```
    android:name=".razro_pay"
    android:exported="true" />
<activity
    android:name=".mybooking"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".placedetailsadmin"
    android:exported="false">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".Mainpage2"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".allcustrecycler"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".about"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".particularecycler"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".customerdetails"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".dummy"
```

```
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".dash2"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".updation"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".onboarding3"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".onboarding2"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".onboarding1"
    android:exported="false">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".dumb"
    android:exported="false">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>
<activity
    android:name=".Forgot_activity"
    android:exported="true">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
```

```
</activity>
<activity
   android:name=".Homeactivity"
   android:exported="true">
   <meta-data
      android:name="android.app.lib_name"
      android:value="" />
</activity>
<activity
   android:name=".register"
   android:exported="true">
   <meta-data
      android:name="android.app.lib_name"
      android:value="" />
</activity>
<activity
   android:name=".login"
   android:exported="true">
   <meta-data
      android:name="android.app.lib_name"
      android:value="" />
</activity>
<activity
   android:name=".User"
   android:exported="true" />
<activity
   android:name=".newactivity"
   android:exported="true">
   <meta-data
      android:name="android.app.lib_name"
      android:value="" />
</activity>
<activity
   android:name=".SplashScreen"
   android:exported="true" />
<activity
   android:name=".MainActivity"
   android:exported="true">
   <intent-filter>
      <action android:name="android.intent.action.MAIN" />

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

   <meta-data
      android:name="android.app.lib_name"
      android:value="" />
</activity>
<activity
   android:name=".dash"
   android:exported="true" />
<activity
   android:name=".imageupload"
```

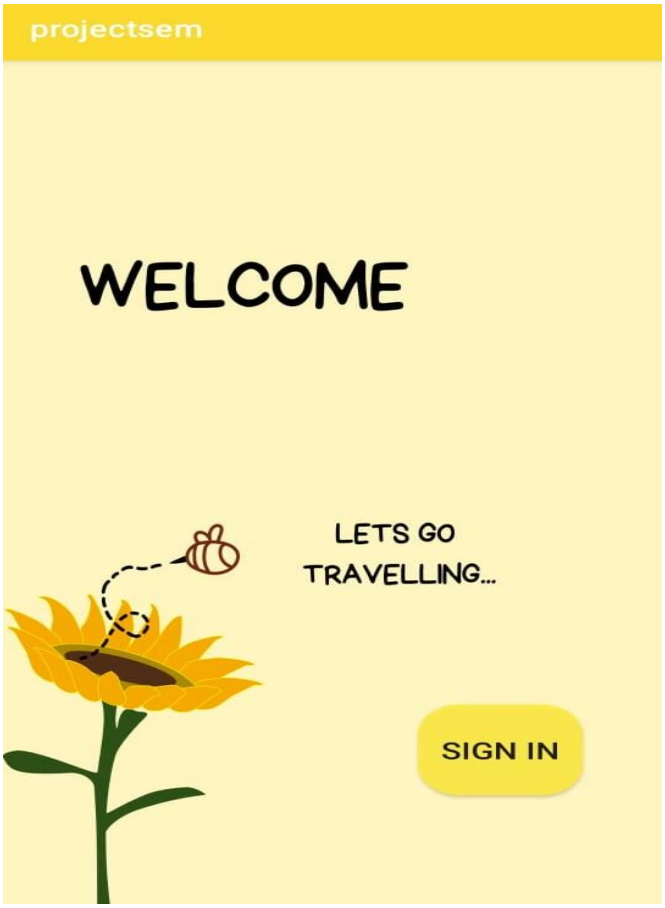
```
        android:exported="true" />
    <activity
        android:name=".MainPage"
        android:exported="true" />
    <activity
        android:name=".placedetails"
        android:exported="true" />
  </application>

</manifest>
```

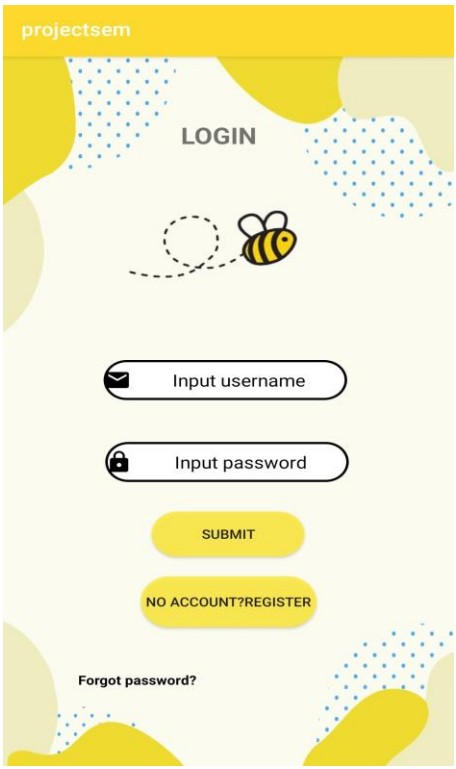
9.USER INTERFACE (SCREENS AND REPORTS)

ACTIVITY_MAIN.XML

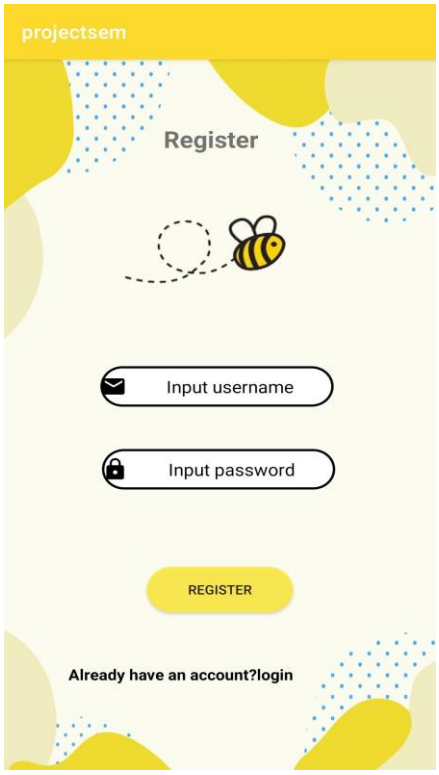
ACTIVITY_WELCOME.XML



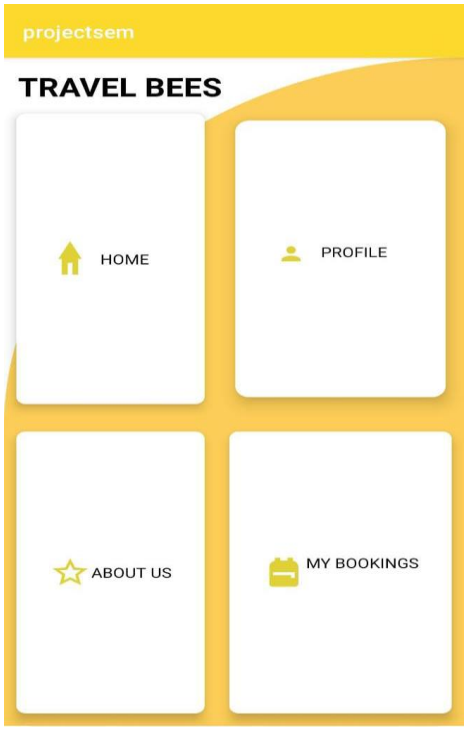
ACTIVITY_LOGIN.XML



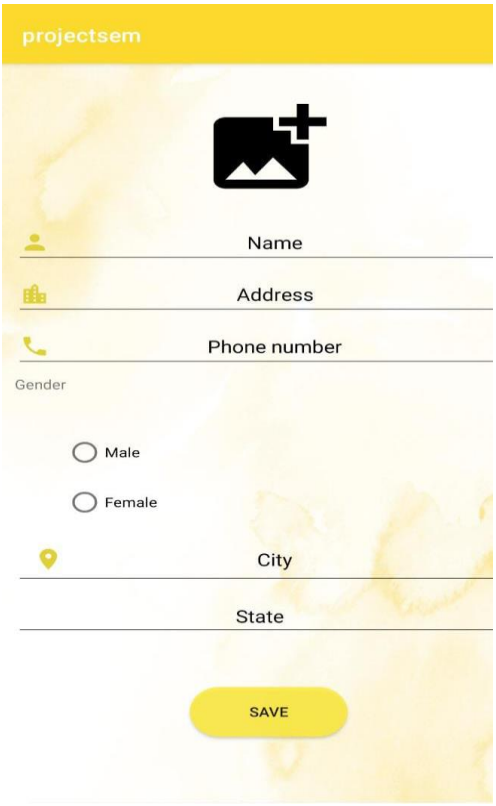
ACTIVITY_REGISTER.XML



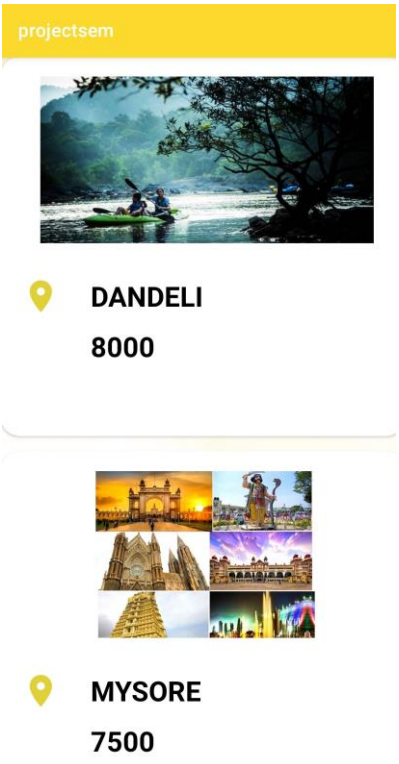
ACTIVITY_DASH2.XML



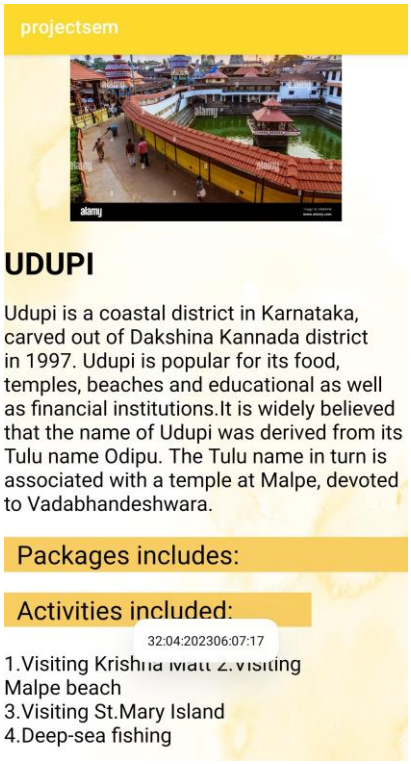
ACTIVITY_PROFILE.XML



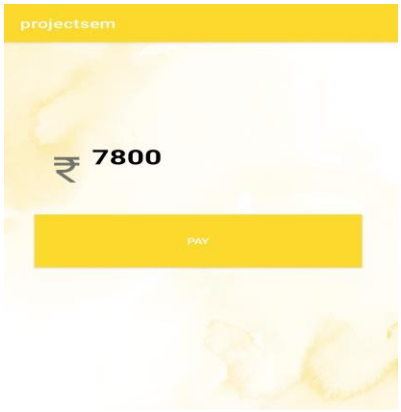
ITEM_MAIN2VIEW.XML

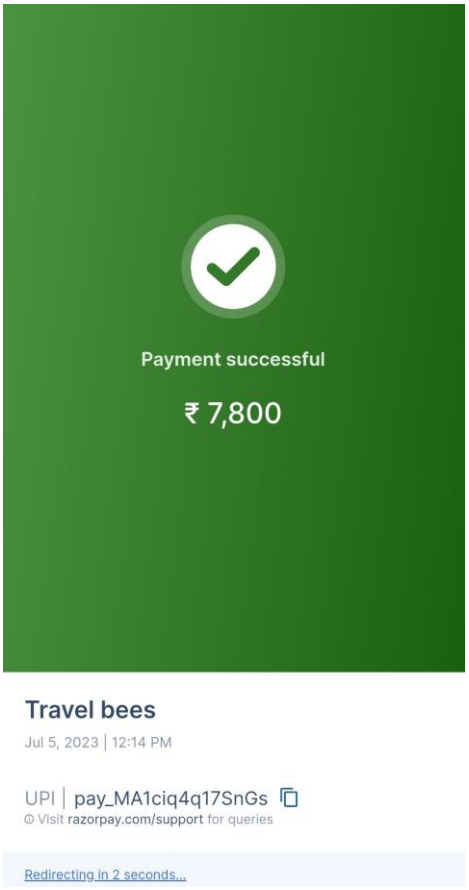


ACTIVITY_PLACEDETAILS.XML



RAZRO_PAY.XML





10.TESTING

Introduction

Testing is the process, which tells the reality efficiency and theflexibility of the system design. Reliability means how much the user is expecting from the system. Flexibility tells how much the user is comfortable and hopes additional facilities with the system.

Testing is vital to the success of the system. System testing makes a logical assumption that if all the part of the system is correct, the goal will be successively achieved. It is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing presents interesting anomaly of the software. The testing phase involves testing of system using various test data.

The first test of system to see whether it produces the correct output. When the software is tested the actual output is tested with the expected output. If there is a discrepancy the sequence of instruction must be traced to determine the problem. Breaking the program down into self-contained portions, each of which can be checked at certain key points facilities the process.

The best program is worthless if it does not meet needs. The first step n system testing is to prepare a plan that will test all aspects of the system in a way that promotes its credibility among potential users. The design phase focuses on the detail implementation of the system recommended in the feasibility study. Emphasis is on translating

performance specification into design specification. The design phase is a transition from a user-oriented document to document oriented to the programmers or database personnel. System design goes through two phases of development, logical and physical design. The logical design describes the input, output, database and procedures. Example: Dataflow Diagram. The physical design procures the working system by defining specification that tells the programmers exactly that what the candidates system must do.

The development of software system involves a series of production activities where opportunities for injunction of human error are enormous. Error may occur at the very imperfectly specified as well as later

design and development stages. Because of human inability to perform andperfection, software development is followed by a quantity assurance activity.

Quantity assurance also places a vital role in the whole development of the system. The quantity assurance whole of the testing phase is the testing phase is to assure that completeness accuracy of the system and minimize the testing process.

System Testing

Software testing is a critical element of software quantity assurance andrepresents the ultimate review of specifications, design and coding. The testing phase involves the testing system using various test data. Preparation of test dataplays a vital role in the system testing. After preparing the test data, the system under study is tested. Those test data, error where found and corrected by following testing steps and corrections were recorded for future reference. Thus a series testing is performed on the system before it is ready for implementation.The various type of testing on system is:

- **Unit testing**
- **Integrated testing**
- **Validation testing**

Unit Testing

Unit testing focuses on verification efforts on the smallest unit of software design module. Using the unit test plans, prepared in the design phase of the system as guide important control paths are tested to uncover error withinthe boundary of the module. The interfaces of the each of the module under consideration are tested. Boundary condition was checked. All independent phases were exercised to ensure error handling path was tested. Each unit was thoroughly tested to check if it might fail in any possible situation. This testing was carried out during the programming itself. At the end of the testing phase, each unit was found to be working satisfactorily, as regarded to the expected output from the module

Integration Testing

Data can have lost across an interface one module can have an adverseeffect on another's sub function, when combined may not produce the desired

major function. Global data structure can present problems. Integration testing isa symmetric testing for constructing tests to unrecovered error associated with theinterface. all modules are combined in these testing steps. Then the entireprogram was tested as a whole.

Validation Testing

At the Culmination of the integration testing the software has completely assembled as a package interfacing error have been uncovered and corrected and a final series of software validation testing begins. Here we test thesystem n a manner that can be reasonably expected by the customer, the system is tested against the system requirement specification.

TEST CASE	TEST OBJECTIVES
1	Test for username and password only.
2	Test for resetting the forgotten password.
3	Test for adding Customer's information.

Unit Testing

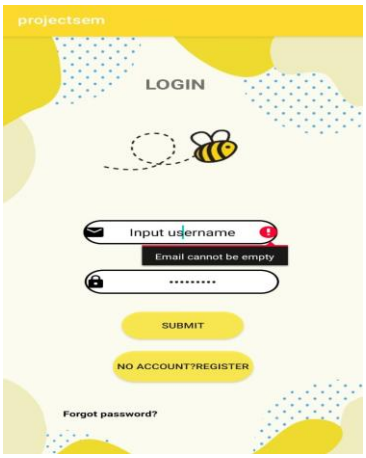
Test Case 1:

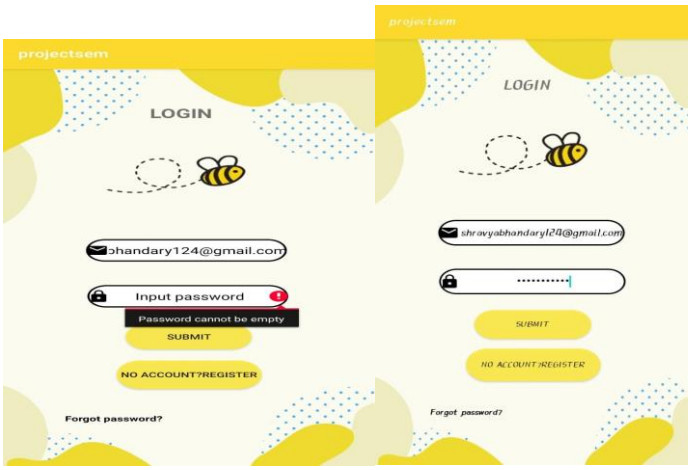
Test Objective : Test for username and password entry
Test Data : Valid-Enter valid username and password and click on LoginButton.
Output : Valid-Customer is successfully logged in and he/she can viewall the services.
Invalid-Shows the error message

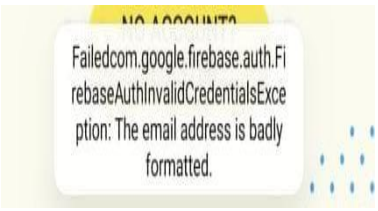
Result : Valid-The customer logins successfully and is allowed to enter the application.

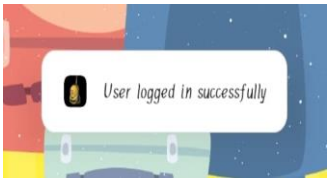
Invalid-The customer is prompt with an error message and is restricted to enter the application.

Conclusion : Both the valid and invalid result is tested. If output tally withrequired result,the test is successful.









Test Case 2:

Test Objective : Test for resetting the forgotten password.

Test Data :Valid-all required field are entered.

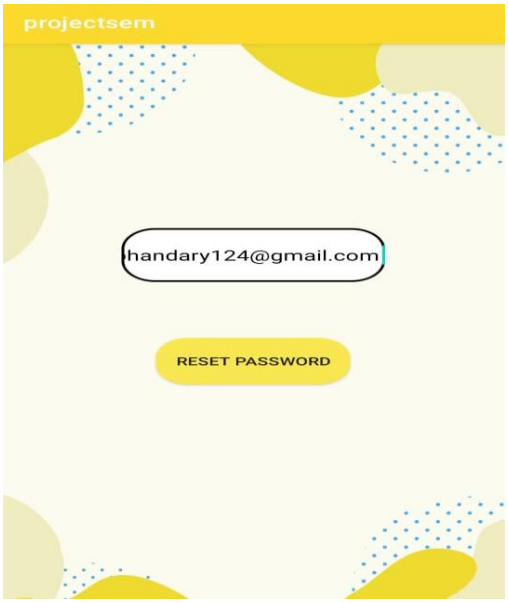
Invalid -If the textbox is blank.

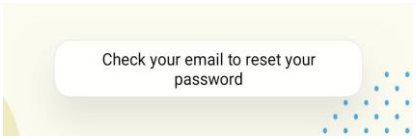
Output : Valid- Allows the customers to change password by sending an email to the registered email id.

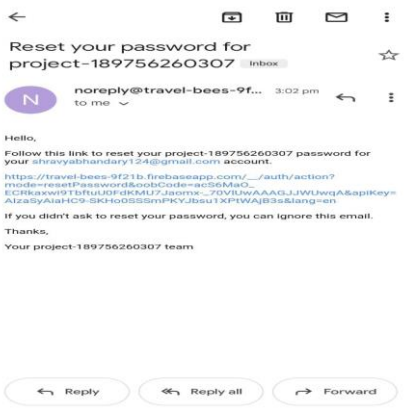
Invalid- User is prompt with an error message.

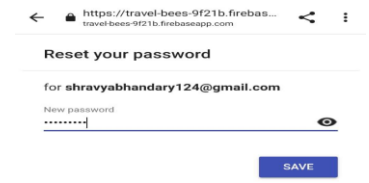
Result : Valid - Email including link to reset password will be sent.

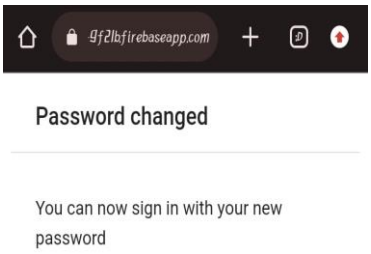
Invalid- Email including link to reset password will not be sent.











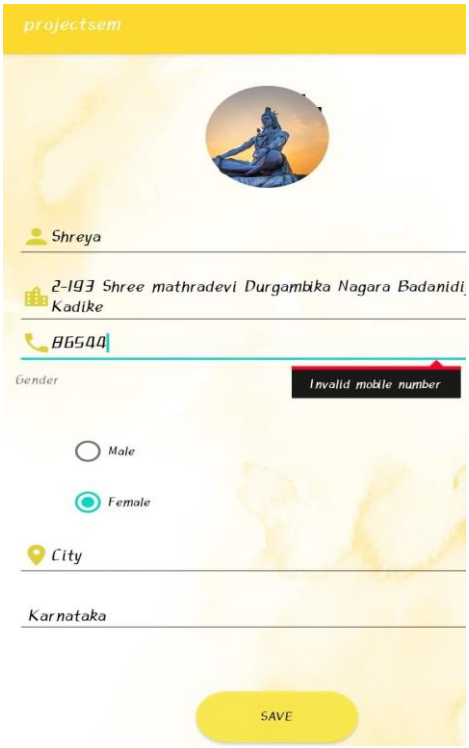
Test Case 3:

Test Objective :Test for adding Customers information.

Test Data : Valid-All required field are entered.
Invalid-Some fields are incorrect or blank.

Output : Valid -Allows all records to be added to the database.
Invalid-The Customer is prompt with an error message.

Result : Valid -All record will be saved.



11.CONCLUSION

Tourism management is a multidisciplinary field that includes all activities related to the tourism and hospitality industries. It involves the management of multitude of activities such as studying tour destination, planning the tour, making travel arrangements and providing accommodation. It also involves marketing efforts to attract tourists to travel to particular destinations.

12.LIMITATIONS

- The application requires Internet Connection.
- The application runs only on Android Based Systems.

13.SCOPE_FOR_ENHANCEMENTS

- This application can be developed to work in IOS.

14.ABBREVIATIONS AND ACRONYMS

Database Reference

A Firebase Reference represents a particular location in your database and can be used for reading or writing data to that Database location.

DataSnapshot

A Data Snapshot instance contains the data from a Firebase Database location. Any time you read data from the Database, you receive the data as a Data Snapshot.

getValue(): getValue() returns the data contained in the snapshot asnative types like Boolean,string,long,double,Map,List.

JSON : Java Script Object Notation

CFD : Context Flow Diagram

DFD : Data Flow Diagram

SDK : Software Development Kit

API : Application programming interface

JDK : Java Development Kit

IDE : Integrated Development Environment

15. BIBLIOGRAPHY

The content for this project has been taken from the following sources:

www.stackoverflow.com
www.geeksforgeeks.org

THANK YOU