ST. TERESA'S COLLEGE(AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY,KOTTAYAM

Hotel Management System

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of **BACHELOR OF SCIENCE IN COMPUTER APPLICATIONS [TRIPLE MAIN]**

By

SEREENA SABU JOHN

III B.Sc. Computer Applications [Triple Main]

Register No: SB22CA031

Under the guidance of

Ms.Megha George

Assistant Professor

Department of Computer Applications 2022 -2025

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

CERTIFICATE

This is to certify that the project report entitled **"Hotel Management System App"** is a bonafide record of the work done by SEREENA SABU JOHN (SB22CA031) during the year 2022 – 2025 and submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Applications (Triple Main) under Mahatma Gandhi University, Kottayam.

Head of the Department 17/03/2025

ST. TERESA'S COLLEGE DEPT. OF COMPUTERS ERNAKULAM

Internal Examiner

Date: 17-03-2025

17/3/25

External Examiner

Dr. Reseena.mol.NA

DECLARATION

I'm, SEREENA SABU JOHN (SB22CA031) B.Sc. Computer Applications [Triple Main] student of St. Teresa's College (Autonomous), Ernakulam, hereby declare that the project submitted for Bachelor's Degree in Computer Application is my original work. We further declare that the said work has not previously been submitted to any other university or academic body.

Date: 17-03-2025

Place:Ernakulam

SEREENA SABU JOHN

ACKNOWLEDGEMENT

I would like to convey my heartfelt gratitude to **Rev. Dr. Sr. Vinitha (CSST) Manager, Director Rev. Sr. Emeline (CSST) and Principal Dr. Alphonsa Vijaya Joseph** for providing me with this wonderful opportunity to work on a project with the topic Hotel Management System App.

I would like to express my profound gratitude to the Head of the Department of Computer Applications and **my project guide Ms. Megha George** and all other faculty of the department for their contributions to the completion of my project. The completion of the project would not have been possible without their help and insights.

I would also like to Thank my project guide at LCC Computer Education Institute, Mr. Gokul Anand for training me well to develop this project.

Finally, I take this opportunity to Thank all them who has directly or indirectly helped me with my project.

SEREENA SABU JOHN

ABSTRACT

This project aims at creating on Hotel Management System which can be used by Admin and Customers. The admin to advise/publish the availability of rooms in different hotels and customers are checking the availability of room in required hotel. Customers should be able to know the availability of the rooms on a particular date to reserve in hotel. They should be able to reserve the available rooms according to their need in advance to make their stay comfortable. The Admin hands the booking information of customers. The users can register and log into the system. The administrator will know the details of reservation and daily income. The hotel department maintain the seat availability and booking details in certain database. This project provides high security to Admin and user information. The main objective of this project is to design a hotel management system for running a hotel business. The system should be as flexible as possible so that it can be used for different hotels. You have to find out which procedures hotels have used for different hotels. You have to find out which procedures hotels have and based on that information, you should create a system which makes it efficient. You need to find out how a hotel system works on the internet, use your own experience or directly talk to people in the hotel business. The more diverse the sources of your information are, the better will be the resulting system and, possibly, your grade.

Contents

INTRODUCTION

1.ABOUT PROJECT

The aim of this project is to develop an integrated Hotel Management System that both administrators and customers can use. The admin will inform customers of the availability of rooms in various hotels, and customers will verify the availability of rooms in the desired hotel. Customers should be able to find out if a room in a particular hotel is available. They should be able to book available rooms in advance based on their needs to make their stay more comfortable. To make their stay more comfortable, they should be able to book available rooms in advance based on their needs. Customers' reservation information is given to the Admin. Users should sign up for the system and log in. The administrator will know the reservation information and regular income. Seat availability and booking information are held in a database by the hotel department. Admin and user data are kept secure in this project. The administrator may manage the hotel's different departments. The specifics of the customer's check-in and check-out are updated. The authority has the power to add hotel packages, room information, online booking, room availability. The main goal of this project is to create a hotel management framework for use in a hotel. The system should be as adaptable as possible, allowing it to be used in a variety of hotels. To learn about the various techniques that hotels have used. We need to figure out what protocol hotels use, and then build a machine that follows those procedures. We need to look up how a hotel system functions on the internet, use our own knowledge, or speak with people who work in the hotel industry directly. This project aims to make record updating, maintenance, and searching more user-friendly. The entire information has been stored in the database, and anyone who wishes to retrieve it will be unable to do so; only authentication will be able to retrieve the correct information from the files.

2.SYSTEM ANALYSIS

2.1INTRODUCTION

System Analysis is a detailed study of the various operations performed by the system ant their relationship within the modules of the system. This phase involves the study of the parent system and identification of the system objectives. The main objective of this phase involves gathering necessary information and using the structured tool for analysis. This includes designing the system. In this project,the requirements are studied in detail and information are collected and documented

2.2EXISTING SYSTEM

The existing (traditional) hotel management system refers to the current methods used by hotels to handle daily operations such as reservations, billing, housekeeping, and guest management. These systems can be manual, semi-automated, or fully digital, depending on the hotel's size and technological adoption.

- Limitations of Existing Systems
- Lack of Automation – Many hotels still rely on manual processes, leading to inefficiencies.
- No Real-time Data – Traditional systems cannot provide instant updates on room availability or pricing.
- Security Concerns – Paper-based or outdated software solutions are vulnerable to data breaches and loss.
- Poor Guest Experience – Slow check-ins, billing errors, and lack of personalized service affect customer satisfaction.

2.3PROPOSED SYSTEM

The Automated system with distributed architecture can support issues like,

- The system maintains the different location that are available and registered in a central DB, which leads easy accessibility and consistency.
- Each Accommodation available units and all the unit facilities are also available at the click of a mouse.
- The registration of new guest is online house new guest can make them they convenient for registration process on the basic of 24*7*326 days.
- The Units can be booked by the Registered guest irrespective of the geographical barriers.
- The Guest are provided with up to minute information related to the unit availability and their status.From their convenient place.
- The decision process in more faster and more consistent.
- The guest have information at their demand related to any unit status of their own unit booking status.

2.4 HARDWARE & SOFTWARE SPECIFICATIONS

- Front End :

  HTML, CSS, JavaScript
- Back End :

  Database Management System : MySQL

- Operating System :

  Windows 10/11, Android

  Browser : Google Chrome
- Software Used :

  Xampp

3.SYSTEM DESIGN

3.1 INTRODUCTION

System design is the process of defining and planning the architecture, components, modules, interfaces, and interactions of a complex software or hardware system. It involves making decisions about how different parts of a system will work together to achieve the desired functionality, performance, scalability, reliability, and maintainability.

The goal of system design is to create a blueprint or roadmap for building a system that meets the requirements and objectives of a project. This includes breaking down the system into smaller subsystems, modules, or components, and determining how they will communicate and collaborate to accomplish the overall goals. System design takes into consideration various technical and non-technical aspects, such as:

Architecture: Deciding on the overall structure of the system, including high-level components, their relationships, and how data flows between them.

Components and Modules: Identifying the individual pieces that will make up the system and defining their responsibilities and interactions.

Data Management: Designing the data model and storage mechanisms that will be used to store, retrieve, and manipulate data within the system.

Communication and Interfaces: Specifying how different components will communicate with each other

Scalability and Performance: Planning for the system's ability to handle increased load and traffic over time.

Security: Addressing potential security vulnerabilities and implementing measures to protect the system from unauthorized access, data breaches, and other security threats.

Reliability and Fault Tolerance: Designing the system to be resilient in the face of failures, ensuring that it can continue to function properly even when certain components or services fail.

Deployment and Infrastructure: Defining how the system will be deployed on various environments (development, testing, production) and determining the required infrastructure, such as servers, networking, and cloud services.

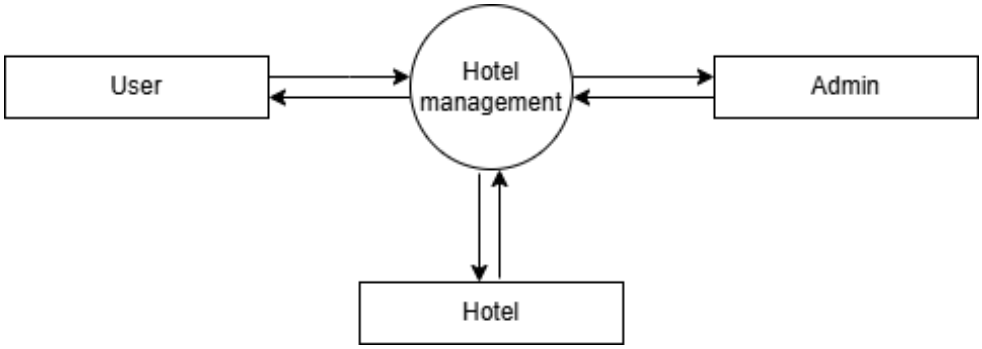
User Experience : Considering how users will interact with the system, including designing intuitive interfaces and workflows that provide a positive user experience.

Maintainability and Extensibility: Creating a design that allows for easy maintenance, updates, and future enhancements without major disruptions to the system.

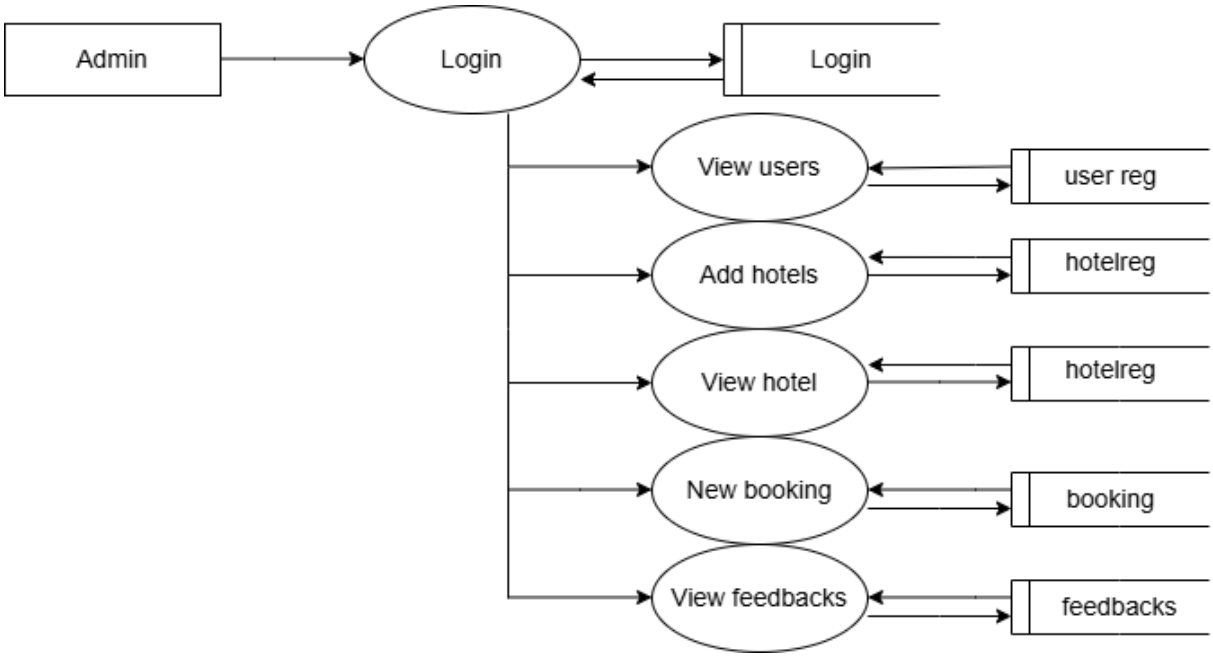
System design often involves collaboration among software architects, engineers, product managers, and other stakeholders. It's a crucial phase in the software development lifecycle, as the decisions made during this stage have a significant impact on the system's overall quality and success

3.2 DATA FLOW DIAGRAM

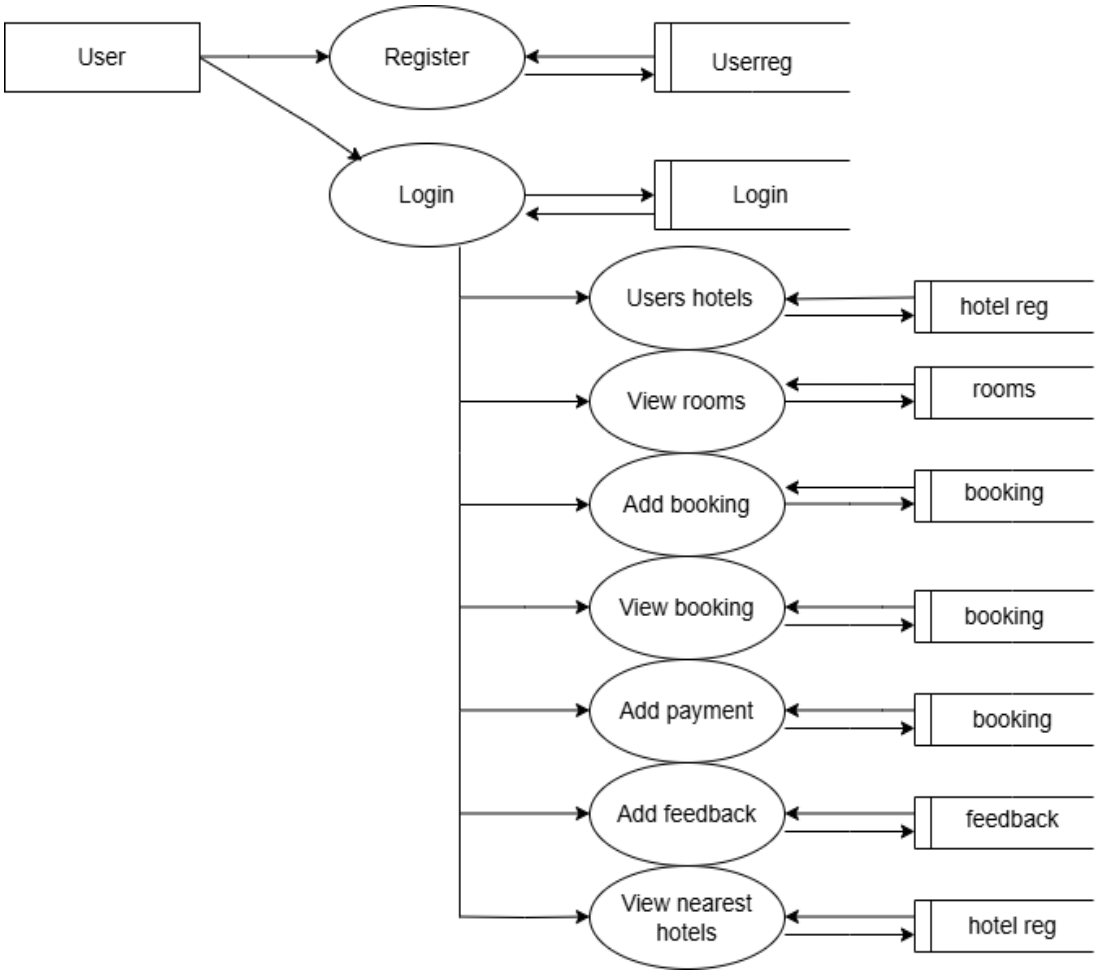
Level 0 :



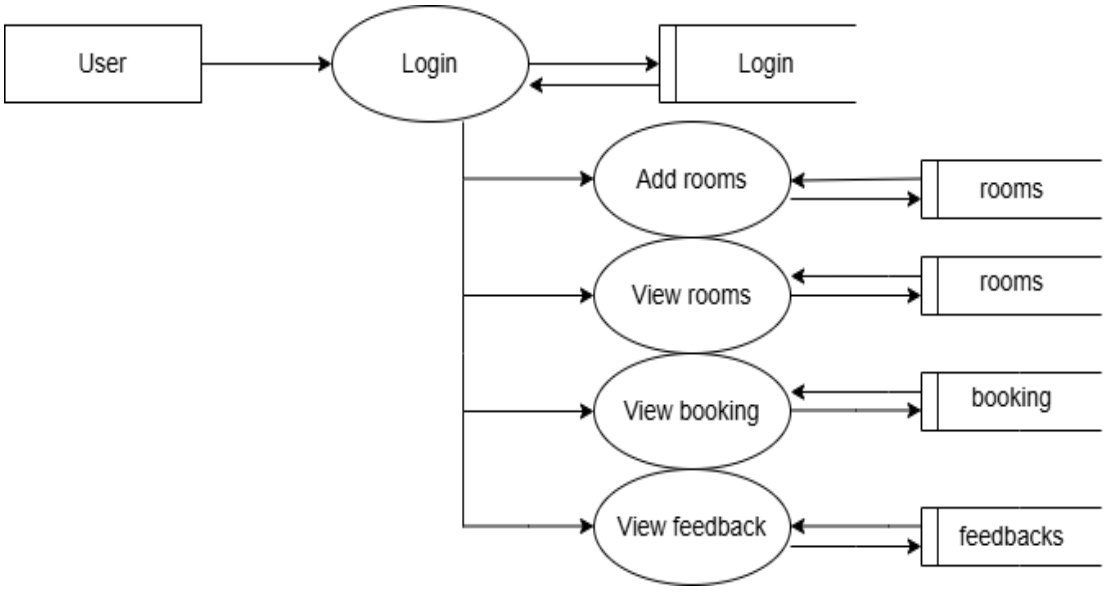
Level 1 : Admin



Level 1.1 : User



Level 1.3 : Hotel



3.3 TABLES DESIGN

Table 1: Login

Field	Type	Default	Key	Description
logid	int (20)	(NULL)	primary	Login Id
uid	int (20)	(NULL)	primary	User id
lemail	varchar(100)	(NULL)	primary	Login email
lpassword	varchar(100)	(NULL)	primary	Login Password
lstatus	varchar(100)	(NULL)	primary	Login Status

Table 2: Userreg

Field	Type	Null	Key	Description
uid	int(20)	No	primary	User id
name	varchar(100)	No	primary	Name of the user
email	varchar(100)	No	primary	Email id of user
phoneno	varchar(100)	No	primary	Phone no of user
password	varchar(100)	No	primary	Password for reg

Table 3: Booking

Field	Type	Null	Key	Description
bid	int(20)	No	PRI	Booking Id
rid	int(20)	No		Register id
hid	int(20)	No		Hotel id
uid	int(20)	No		User id
number_of_guests	varchar(100)	No		Guests details
bstatus	varchar(100)	No		Booking status
no_of_nights	varchar(100)	No		Day and night preference
b_room_type	varchar(100)	No		Booking room type
b_room_price	varchar(100)	No		Room price
check_in_date_time	varchar(100)	Yes		Checkin date and time
check_out_date_time	varchar(100)	No		Check out date and time

4.SYSTEM DEVELOPMENT

4.1 PROCESS

Admin : This module manage overall activities of the system.

- Login

  Manage Users

  Update & Delete Hotels

- Manage Hotels

  Update hotel

  Delete hotel

- Reservation
- Feedbacks
- Availability of hotels and rooms
- Checkin checkout

USER : Can book the rooms.

- Login
- Booking rooms
- Payment
- Checking the service
- Feedbacks

Hotel Mangers

- Login
- Add hotels
- Checking the availability of rooms
- Service
- Payment
- Feedbacks

5.SYSTEM TESTING AND IMPLEMENTATION

5.1 INTRODUCTION

System Testing is a type of software testing that is performed on a completely integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or the context of both. System testing tests the design and behavior of the system and also the expectations of the customer.

5.2 IMPLEMENTATION

The software implementation phase is part of the continuous improvement cycle where deliverables designed are adopted and made operational by an organization or end-users. It is an executing phase of the system development lifecycle that includes the steps of concept to function by coding, testing, and deployment. Implementation encompasses a range of steps that include installation, configuration, customization, and integration to make sure it runs as expected and fulfills the expected objectives and requirements.

- Testing the development system with the sample data.
- Detection and correction of errors.
- Making necessary changes in the system.
- Training and involvement of user personnel.
- Installation of software utilities.

6.CONCLUSION

The developed system effectively streamlines hotel operations by automating key tasks like room reservations, check-in/check-out processes, billing, and staff management, leading to improved efficiency, reduced errors, enhanced guest satisfaction, and valuable data insights for better decision-making. Ultimately contributing to increased profitability for the hotel business. While highlighting the user-friendly interface and security features implemented within the system.

7. SAMPLE CODE

7.1Admin :

```
public class AddHotels extends AppCompatActivity {

    EditText hname, hemail, hpassword, hphoneno, hdescription,   hwebsite,addressEditText;
    Button register,btn;
    Spinner htype;
    RatingBar star;

    String ADDRESS = "";
    GPSTracker gps;
    double latitude;
    double longitude;
    private static final int PICK_IMAGE = 1;
    private static final int PICK_Camera_IMAGE = 2;
    private Bitmap bitmap;
    String bal = "", MAX;
    Uri imageUri;
    EditText  address;

    FusedLocationProviderClient mFusedLocationClient;
    int PERMISSION_ID = 44;
    TextView LAT, LONG, mapaddress;
    ImageView locationMarker,img, camera, gallery;
    String na, em, pa, ph, ty, im, des, webb, ra,addresss,latitudee,longitudee;
    TextView latitudeTextView, longitudeTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_hotels);

        hname = findViewById(R.id.hotel_name);
        hemail = findViewById(R.id.hotel_email);
        hphoneno = findViewById(R.id.hotel_phoneno);
        hpassword = findViewById(R.id.hotel_pass);
        hdescription = findViewById(R.id.hotel_descri);
        htype = findViewById(R.id.hotel_type);
        addressEditText = findViewById(R.id.hotel_address);
        locationMarker=findViewById(R.id.locationIcon);
        latitudeTextView = findViewById(R.id.Lat);
        longitudeTextView = findViewById(R.id.Long);
```

```
        LAT = findViewById(R.id.Lat);
        LONG = findViewById(R.id.Long);
        hwebsite = findViewById(R.id.websiteurl);
        star = (RatingBar) findViewById(R.id.starid);
        img = findViewById(R.id.add_image);
        camera = findViewById(R.id.cam);
        gallery = findViewById(R.id.gal);
        register = findViewById(R.id.register_btn);


        camera.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // TODO Auto-generated method stub
                String fileName = "new-photo-name.jpg";
                //create parameters for Intent with filename
                ContentValues values = new ContentValues();
                values.put(MediaStore.Images.Media.TITLE, fileName);
                values.put(MediaStore.Images.Media.DESCRIPTION, "Image captured by
camera");
                //imageUri is the current activity attribute, define and save it for later usage (also in
onSaveInstanceState)\Context.
                imageUri =
getApplicationContext().getContentResolver().insert(MediaStore.Images.Media.EXTERNAL
_CONTENT_URI, values);
                //create new Intent
                Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
                intent.putExtra(MediaStore.EXTRA_VIDEO_QUALITY, 1);
                startActivityForResult(intent, PICK_Camera_IMAGE);
            }
        });
        gallery.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                // TODO Auto-generated method stub
                try {
                    Intent gintent = new Intent();
                    gintent.setType("image/*");
                    gintent.setAction(Intent.ACTION_GET_CONTENT);
                    startActivityForResult(Intent.createChooser(gintent, "Select Picture"),
```

```
PICK_IMAGE);

            } catch (Exception e) {
                Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
                Log.e(e.getClass().getName(), e.getMessage(), e);
            }
        }
    });
```

7.2User :

```
public class BookRoom extends AppCompatActivity {

    EditText guest, checkin,checkout,nonights,specialreq,no_room,total_price;
    TextView roomtype, price;
    Button book;

    String rid,rty,rpr,rni,req,gues,hid,chein,cheout,bid,total,hotelna;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_book_room);

        roomtype=findViewById(R.id.b_room_type);
        price=findViewById(R.id.b_room_price);
        total_price = findViewById(R.id.b_total_price);
        guest = findViewById(R.id.number_of_guests);
        checkin = findViewById(R.id.check_in_date_time);
        checkout = findViewById(R.id.check_out_date_time);
        nonights = findViewById(R.id.no_of_nights);
        specialreq = findViewById(R.id.special_requests);
//        no_room = findViewById(R.id.number_of_rooms);
        book = findViewById(R.id.submit_booking);


        // Calculate Total Price Dynamically
        nonights.setOnFocusChangeListener((v, hasFocus) -> {
            if (!hasFocus) {
                calculateTotalPrice();
            }
        });

        // Check-in Date & Time
        checkin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Calendar calendar = Calendar.getInstance();
                int year = calendar.get(Calendar.YEAR);
                int month = calendar.get(Calendar.MONTH);
                int day = calendar.get(Calendar.DAY_OF_MONTH);

                DatePickerDialog datePickerDialog = new DatePickerDialog(
```

```
                v.getContext(),
                (view, selectedYear, selectedMonth, selectedDay) -> {
                    selectedMonth += 1;
                    String selectedDate = selectedDay + "/" + selectedMonth + "/" +
selectedYear;

                    // Open TimePickerDialog after date selection
                    TimePickerDialog timePickerDialog = new TimePickerDialog(
                            v.getContext(),
                            (timeView, hourOfDay, minute) -> {
                                String selectedTime = String.format("%02d:%02d", hourOfDay,
minute);
                                String checkInDateTime = selectedDate + " " + selectedTime;
                                checkin.setText(checkInDateTime);
                            },
                            calendar.get(Calendar.HOUR_OF_DAY),
                            calendar.get(Calendar.MINUTE),
                            false
                    );
                    timePickerDialog.setTitle("Select Check-in Time");
                    timePickerDialog.show();
                },
                year, month, day
            );

            datePickerDialog.setTitle("Select Check-in Date");
            datePickerDialog.show();
        }
    });

// Check-out Date & Time
    checkout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Calendar calendar = Calendar.getInstance();
            int year = calendar.get(Calendar.YEAR);
            int month = calendar.get(Calendar.MONTH);
            int day = calendar.get(Calendar.DAY_OF_MONTH);

            DatePickerDialog datePickerDialog = new DatePickerDialog(
                    v.getContext(),
                    (view, selectedYear, selectedMonth, selectedDay) -> {
                        selectedMonth += 1;
```

```
                    String selectedDate = selectedDay + "/" + selectedMonth + "/" +
selectedYear;

                    // Open TimePickerDialog after date selection
                    TimePickerDialog timePickerDialog = new TimePickerDialog(
                            v.getContext(),
                            (timeView, hourOfDay, minute) -> {
                                String selectedTime = String.format("%02d:%02d", hourOfDay,
minute);
                                String checkOutDateTime = selectedDate + " " + selectedTime;
                                checkout.setText(checkOutDateTime);
                            },
                            calendar.get(Calendar.HOUR_OF_DAY),
                            calendar.get(Calendar.MINUTE),
                            false
                    );
                    timePickerDialog.setTitle("Select Check-out Time");
                    timePickerDialog.show();
                },
                year, month, day
            );

            datePickerDialog.setTitle("Select Check-out Date");
            datePickerDialog.show();
        }
    });


    Intent i = getIntent();
    rid = i.getStringExtra("rid");
    hid = i.getStringExtra("hid");
    rty = i.getStringExtra("roomtype");
    rpr = i.getStringExtra("roomprice");

    roomtype.setText(rty != null ? rty : "N/A");
    price.setText(rpr != null ? rpr : "N/A");


    book.setOnClickListener(v -> {
        rty=roomtype.getText().toString();
        rpr=price.getText().toString();
        chein=checkin.getText().toString();
        cheout=checkout.getText().toString();
```

```
        gues=guest.getText().toString();
        req=specialreq.getText().toString();
        rni=nonights.getText().toString();
        total=total_price.getText().toString();

        bookRoom();
    });
}
```

7.3Hotels :

```
public class AddRooms extends AppCompatActivity {

    EditText noroom, maxoccu, price, roomdesc;
    Button btnroom;

    TextView facilityText;
    Spinner roomtype, ava_status;
    ImageView img, camera, gallery;
    private static final int PICK_IMAGE = 1;
    private static final int PICK_Camera_IMAGE = 2;
    private Bitmap bitmap;
    String bal = "", MAX;
    Uri imageUri;
    String ty, no, max, im, des, pri, faci, sta;

    private ArrayList<String> selectedFacilities = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_rooms);

        roomtype = findViewById(R.id.room_type);
        noroom = findViewById(R.id.number_of_rooms);
        maxoccu = findViewById(R.id.max_occupancy);
        price = findViewById(R.id.price_per_night);
        roomdesc = findViewById(R.id.room_description);
        facilityText = findViewById(R.id.faci_text);
        ava_status = findViewById(R.id.availability_status);
        img = findViewById(R.id.add_image);
        camera = findViewById(R.id.cam);
        gallery = findViewById(R.id.gal);
        btnroom = findViewById(R.id.add_room_button);

        camera.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // TODO Auto-generated method stub
                String fileName = "new-photo-name.jpg";
                //create parameters for Intent with filename
                ContentValues values = new ContentValues();
                values.put(MediaStore.Images.Media.TITLE, fileName);
```

```
            values.put(MediaStore.Images.Media.DESCRIPTION, "Image captured by
camera");
            //imageUri is the current activity attribute, define and save it for later usage (also in
onSaveInstanceState)\Context.
            imageUri =
getApplicationContext().getContentResolver().insert(MediaStore.Images.Media.EXTERNAL
_CONTENT_URI, values);
            //create new Intent
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
            intent.putExtra(MediaStore.EXTRA_VIDEO_QUALITY, 1);
            startActivityForResult(intent, PICK_Camera_IMAGE);
         }
      });
      gallery.setOnClickListener(new View.OnClickListener() {

         @Override
         public void onClick(View arg0) {
            // TODO Auto-generated method stub
            // TODO Auto-generated method stub
            try {
               Intent gintent = new Intent();
               gintent.setType("image/*");
               gintent.setAction(Intent.ACTION_GET_CONTENT);
               startActivityForResult(Intent.createChooser(gintent, "Select Picture"),
PICK_IMAGE);

            } catch (Exception e) {
               Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
               Log.e(e.getClass().getName(), e.getMessage(), e);
            }
         }
      });

      String room_type[] = {"Choose Room Type", "Standard Room", "Deluxe Room", "Suite
Room", "Executive Room", "Family Room", "Twin Room"};

      ArrayAdapter arrayAdapter = new ArrayAdapter(getApplicationContext(),
android.R.layout.simple_spinner_dropdown_item, room_type) {
         @Override
         public boolean isEnabled(int position) {
            return position != 0;
```

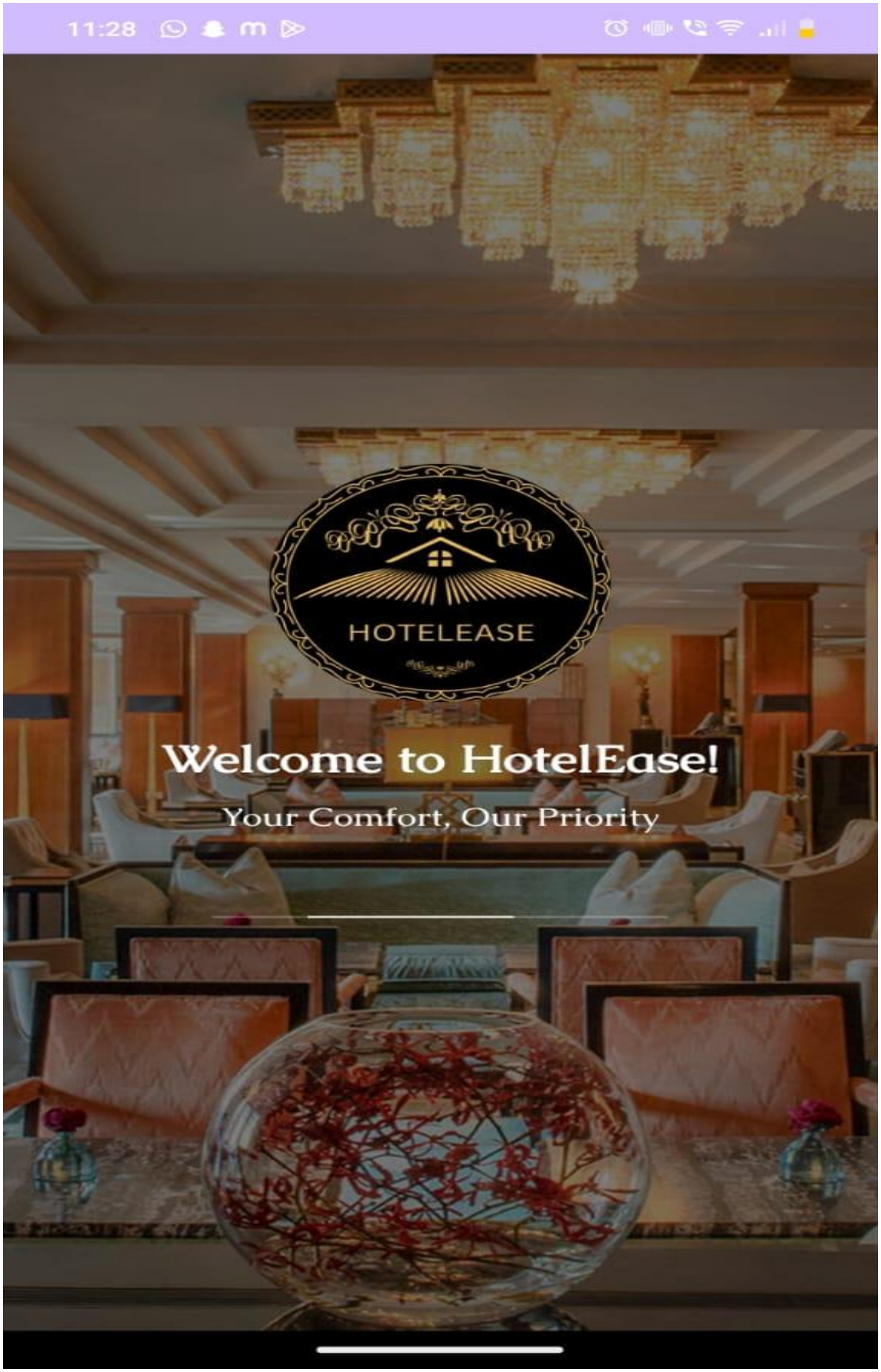
```
        }

        @RequiresApi(api = Build.VERSION_CODES.O)
        @Override
        public View getDropDownView(int position, @Nullable View convertView,
@NonNull ViewGroup parent) {
            View view = super.getDropDownView(position, convertView, parent);
            TextView textView = (TextView) view;
            if (position == 0) {
                Typeface typeface = ResourcesCompat.getFont(getApplicationContext(),
R.font.artifika);
                textView.setTypeface(typeface);
                textView.setTextColor(Color.BLUE);
                textView.setBackgroundColor(Color.LTGRAY);
            } else {
                textView.setTextColor(Color.BLUE);
                Typeface typeface = ResourcesCompat.getFont(getApplicationContext(),
R.font.artifika);
                textView.setTypeface(typeface);
            }
            return view;
        }

        @Override
        public View getView(int position, View convertView, ViewGroup parent) {
            View view = super.getView(position, convertView, parent);
            TextView textView = (TextView) view;
            Typeface typeface = ResourcesCompat.getFont(getApplicationContext(),
R.font.artifika);
            textView.setTypeface(typeface);
            return view;
        }
    };

arrayAdapter.setDropDownViewResource(androidx.appcompat.R.layout.support_simple_spi
nner_dropdown_item);
    roomtype.setAdapter(arrayAdapter);

    String av_status[] = {"Choose Availability Status", "Available", "Not Available"};

    ArrayAdapter arrayAdapter1 = new ArrayAdapter(getApplicationContext(),
android.R.layout.simple_spinner_dropdown_item, av_status) {
        @Override
```

```
        public boolean isEnabled(int position) {
            return position != 0;
        }

        @RequiresApi(api = Build.VERSION_CODES.O)
        @Override
        public View getDropDownView(int position, @Nullable View convertView,
@NonNull ViewGroup parent) {
            View view = super.getDropDownView(position, convertView, parent);
            TextView textView = (TextView) view;
            if (position == 0) {
                Typeface typeface = ResourcesCompat.getFont(getApplicationContext(),
R.font.artifika);
                textView.setTypeface(typeface);
                textView.setTextColor(Color.BLUE);
                textView.setBackgroundColor(Color.LTGRAY);
            } else {
                textView.setTextColor(Color.BLUE);
                Typeface typeface = ResourcesCompat.getFont(getApplicationContext(),
R.font.artifika);
                textView.setTypeface(typeface);
            }
            return view;
        }

        @Override
        public View getView(int position, View convertView, ViewGroup parent) {
            View view = super.getView(position, convertView, parent);
            TextView textView = (TextView) view;
            Typeface typeface = ResourcesCompat.getFont(getApplicationContext(),
R.font.artifika);
            textView.setTypeface(typeface);
            return view;
        }
    };

arrayAdapter.setDropDownViewResource(androidx.appcompat.R.layout.support_simple_spi
nner_dropdown_item);
    ava_status.setAdapter(arrayAdapter1);

    String Facilities[] = {"Air Conditioning (AC)", "Free Wi-Fi", "Smart TV", "Mini-Bar",
"Private Bathroom", "Balcony Access", "Heating System", "Coffee/Tea Maker",
"Refrigerator", "Microwave", "Bathtub", "Mirror with Lighting","In-Room Dining (24/7
```

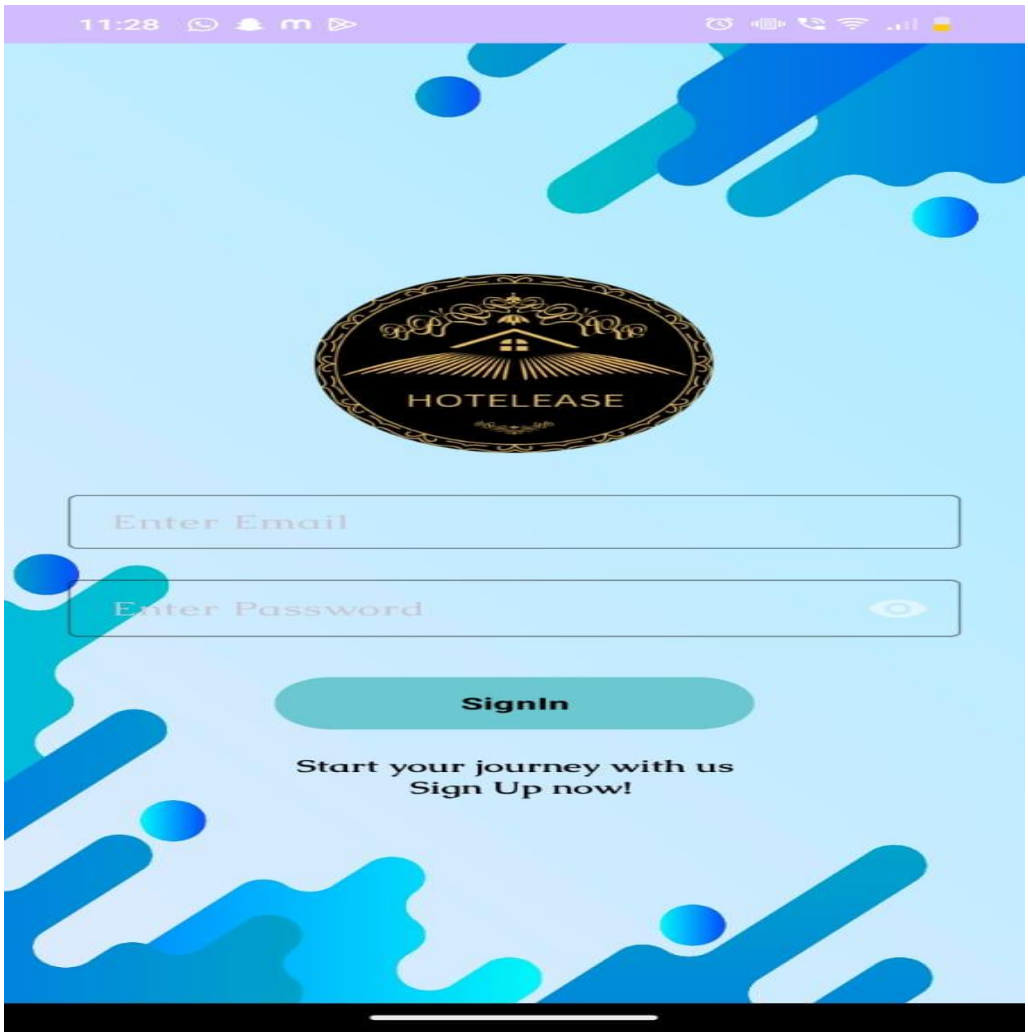
```
Room Service)","Mini Kitchenette","USB Charging Ports"};

        //facilityText Multi-Selection
        facilityText.setOnClickListener(v -> showMultiSelectDialog("Select Facilities",
Facilities, selectedFacilities, facilityText));

        btnroom.setOnClickListener(v -> {
            no = noroom.getText().toString();
            max = maxoccu.getText().toString();
            pri = price.getText().toString();
            des = roomdesc.getText().toString();
            sta = ava_status.getSelectedItem().toString();
            ty = roomtype.getSelectedItem().toString();

            if (no.isEmpty()) {
                Snackbar.make(noroom, "Please Enter No of Rooms", Snackbar.LENGTH_LONG)
                        .setAction("dismiss", new View.OnClickListener() {
                            @Override
                            public void onClick(View view) {

                            }
                        })

.setActionTextColor(getResources().getColor(android.R.color.holo_red_light)).show();

            } else if (max.isEmpty()) {
                Snackbar.make(maxoccu, "Please Enter Maximum Ocuupancy",
Snackbar.LENGTH_LONG)
                        .setAction("dismiss", new View.OnClickListener() {
                            @Override
                            public void onClick(View view) {

                            }
                        })

.setActionTextColor(getResources().getColor(android.R.color.holo_red_light)).show();

            } else if (pri.isEmpty()) {
                Snackbar.make(price, "Please Enter Price", Snackbar.LENGTH_LONG)
                        .setAction("dismiss", new View.OnClickListener() {
                            @Override
                            public void onClick(View view) {
```

```
                    }
                })

.setActionTextColor(getResources().getColor(android.R.color.holo_red_light)).show();
        } else if (des.isEmpty()) {
            Snackbar.make(roomdesc, "Please Enter Description", Snackbar.LENGTH_LONG)
                    .setAction("dismiss", new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {

                        }
                    })

.setActionTextColor(getResources().getColor(android.R.color.holo_red_light)).show();

        } else {
            addRooms();
        }

    });

}
```

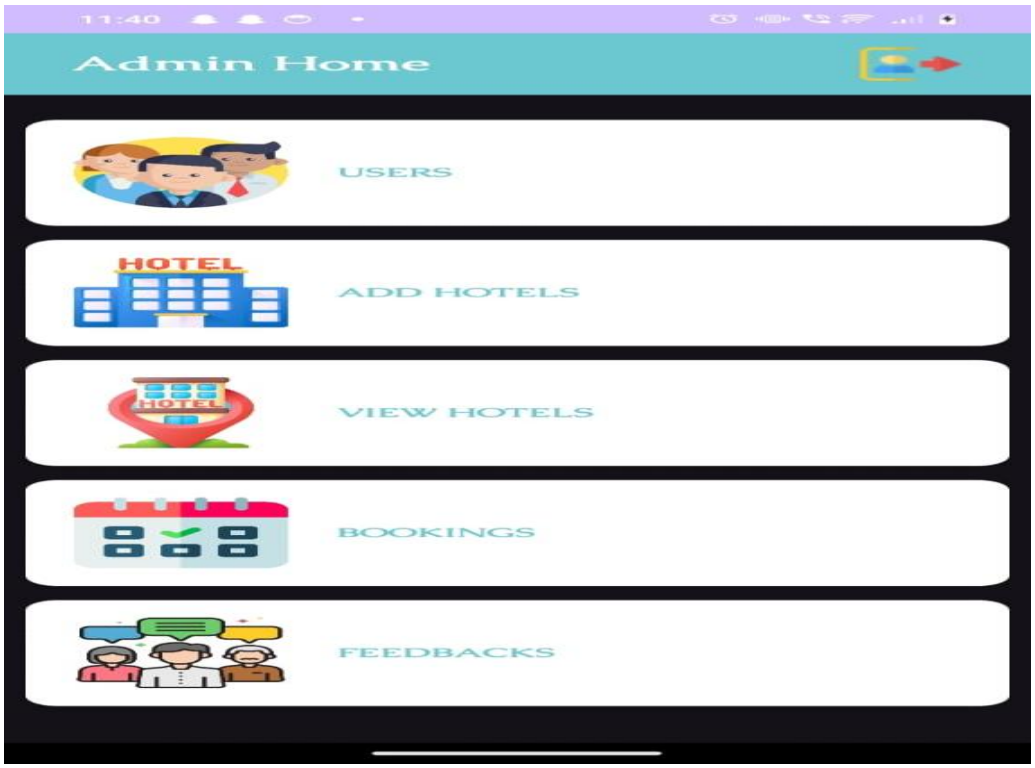
8.APPENDIX

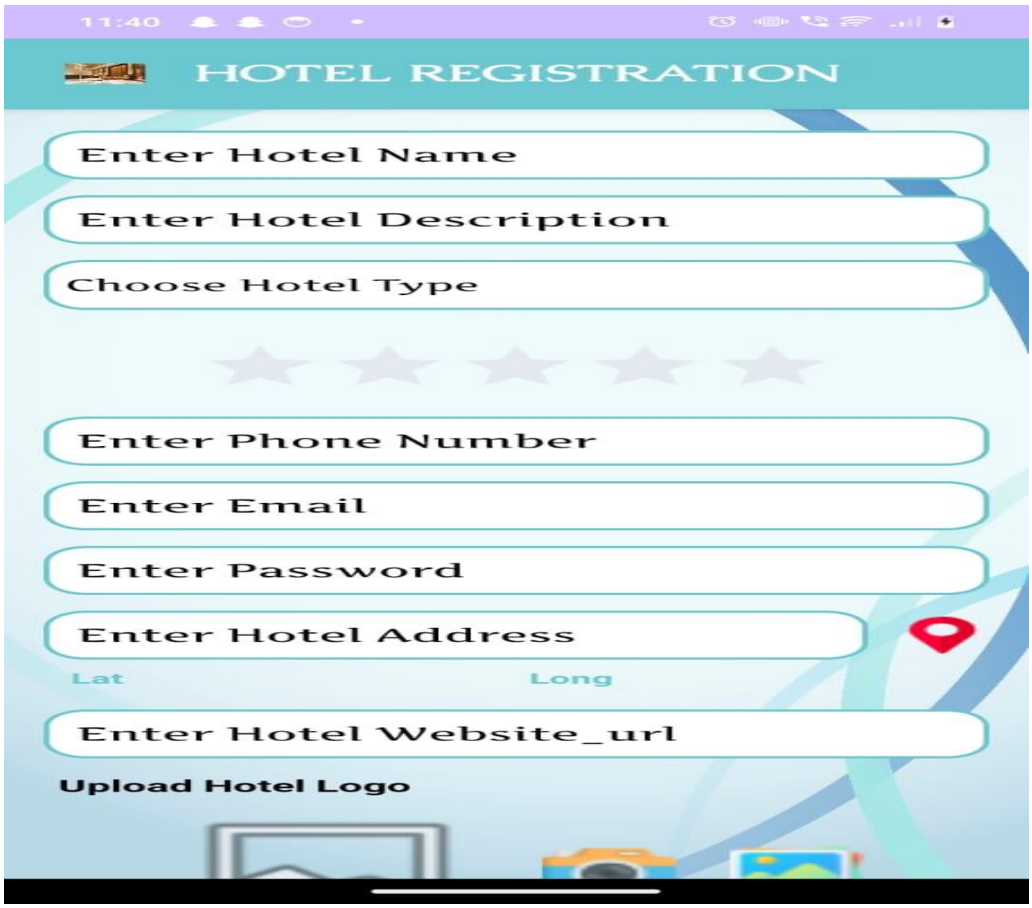

11:28
HOTELEASE
Welcome to HotelEase!
Your Comfort, Our Priority

8.1Login page

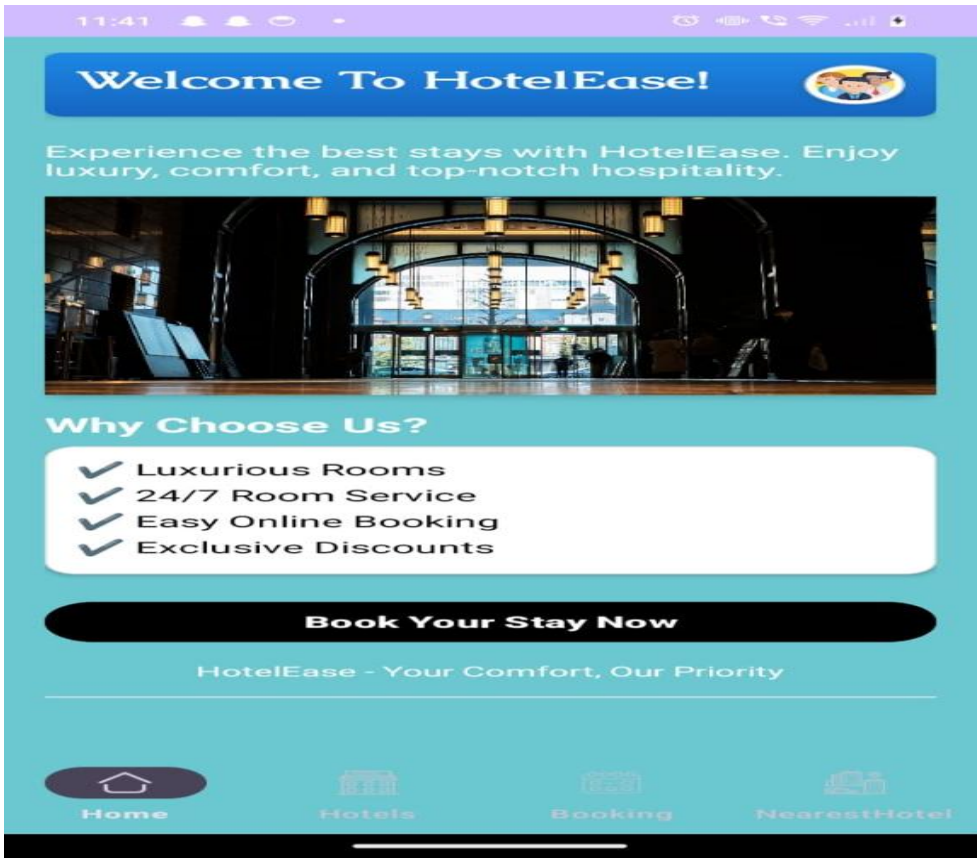


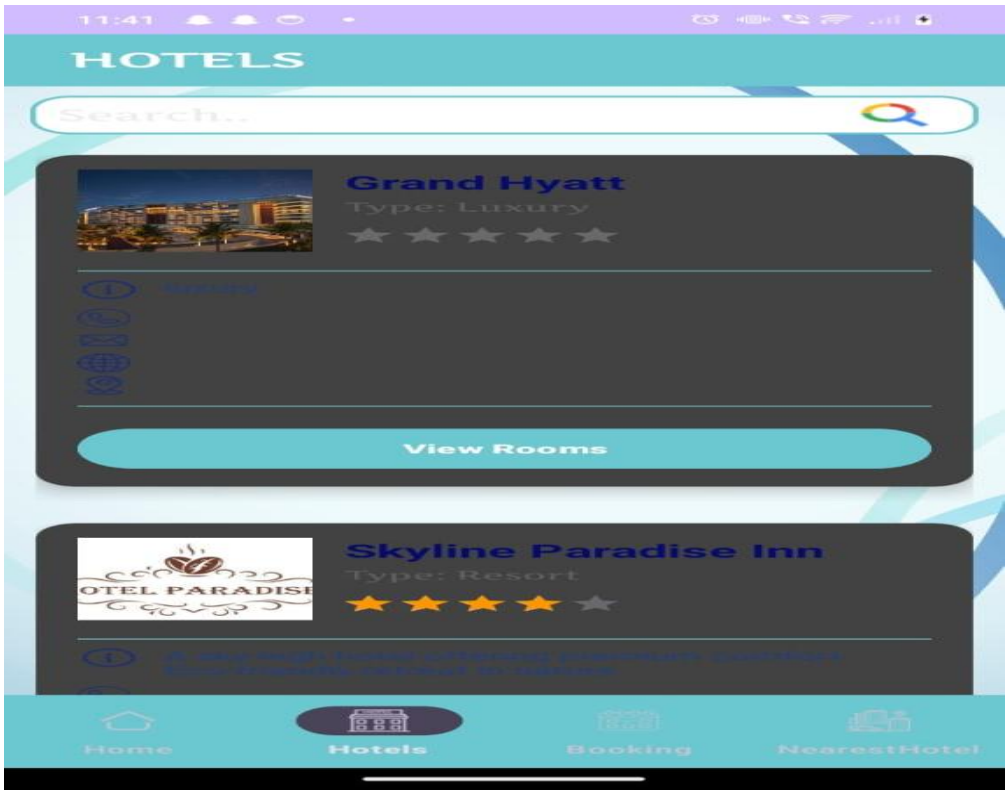
8.2Admin


11:40
Admin Home
USERS
HOTEL
ADD HOTELS
HOTEL
VIEW HOTELS
BOOKINGS
FEEDBACKS

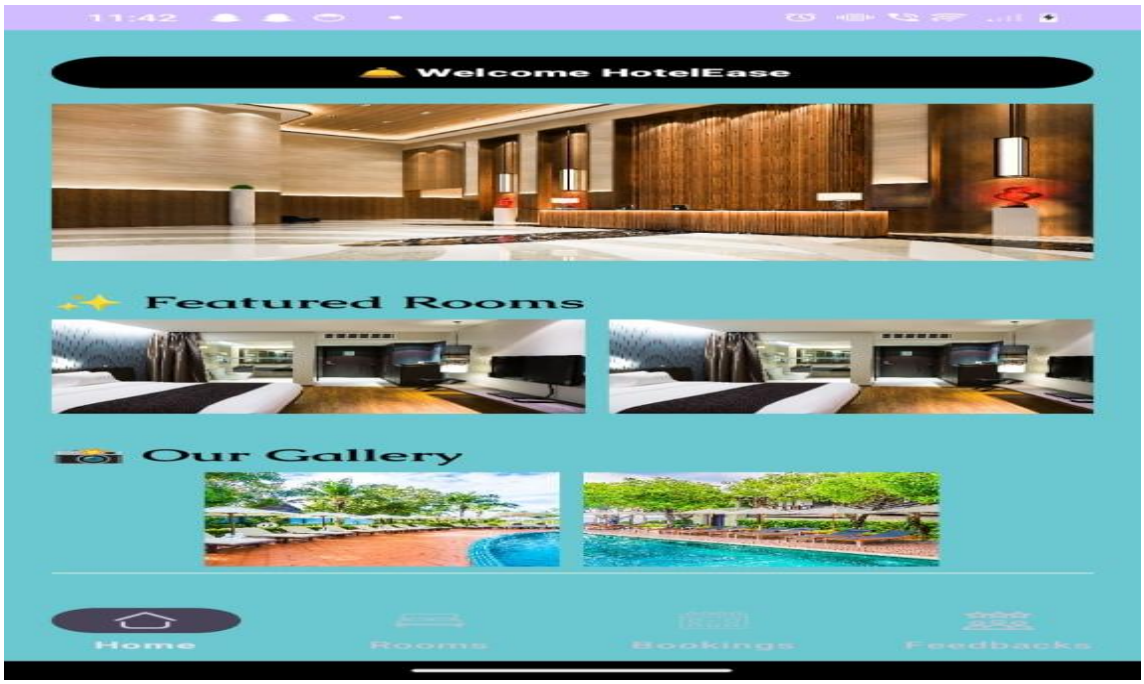

11:40
HOTEL REGISTRATION
Enter Hotel Name
Enter Hotel Description
Choose Hotel Type
Enter Phone Number
Enter Email
Enter Password
Enter Hotel Address
Lat
Long
Enter Hotel Website_url
Upload Hotel Logo

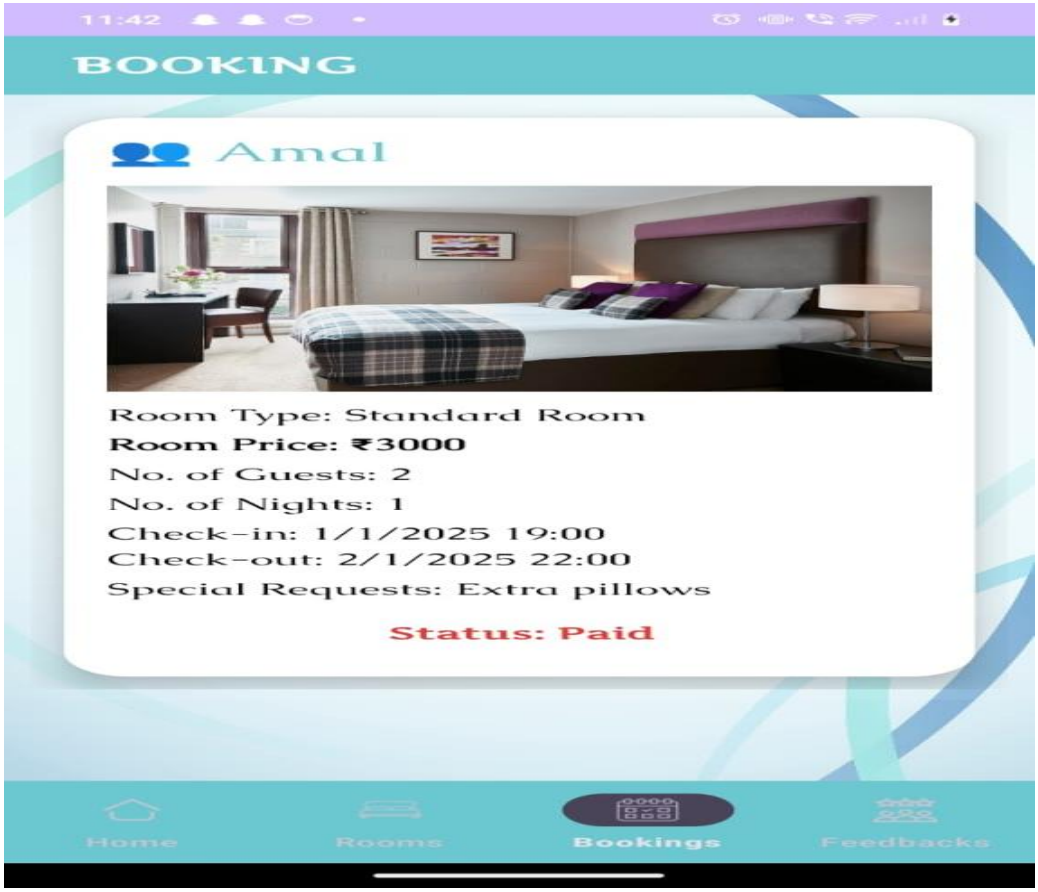
8.3User


11:41
Welcome To HotelEase!
Experience the best stays with HotelEase. Enjoy luxury, comfort, and top-notch hospitality.
Why Choose Us?
Luxurious Rooms
24/7 Room Service
Easy Online Booking
Exclusive Discounts
Book Your Stay Now
HotelEase - Your Comfort, Our Priority
Home
Hotels
Booking
NearestHotel


11:41
HOTELS
Search...
Grand Hyatt
Type: Luxury
View Rooms
Skyline Paradise Inn
Type: Resort
Home
Hotels
Booking
NearestHotel

8.4Hotels


11:42
Welcome HotelEase
Featured Rooms
Our Gallery
Home
Rooms
Bookings
Feedbacks


11:42
BOOKING
Amal
Room Type: Standard Room
Room Price: ₹3000
No. of Guests: 2
No. of Nights: 1
Check-in: 1/1/2025 19:00
Check-out: 2/1/2025 22:00
Special Requests: Extra pillows
Status: Paid
Home
Rooms
Bookings
Feedbacks

8.REFERENCES

- https://www.geeksforgeeks.org/hotel-management-system/
- https://www.jetir.org/papers/JETIR2005452
- https://www.weareplanet.com/blog/hotel-management-system-guide
- https://www.slideshare.net.com
- https://vertabelo.com/blog/data-model-for-hotel-management-system
- https://github.com
- https://www.altexsoft.com/blog/hotel-property-management-systems-