VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Santhibastwad road, Machhe Belagavi-590018,
Karnataka, India

MOBILE APPLICATION DEVELOPMENT MINI PROJECT (18CSMP68) REPORT

ON

"FLIGHT BOOKING APP"

Submitted in partial fulfillment of the requirements
for the Degree of

BACHELOR OF ENGINEERING

In

INFORMATION SCIENCE AND ENGINEERING

For the Academic Year 2022-2023

Submitted By

ANUSHKA ROY	**1JS20IS023**
G SANJANA REDDY	**1JS20IS036**

Under the guidance of

Mrs. Nagashree S, Ms. Sukrutha C B
Assistant Professor, ISE, JSSATE

DEPARTMENT OF INFORMATION SCIENCE OF ENGINEERING
JSS ACADEMY OF TECHNICAL EDUCATION
JSSATEB Campus, Dr Vishnuvardhan Road, Uttarahalli Kengeri Main Road,
Bangalore-560060.

JSS MAHAVIDYAPEETHA, MYSURU

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr. Vishnuvardhan Road, Bengaluru - 560060

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that MOBILE APPLICATION DEVELOPMENT MINI PROJECT Report entitled **"FLIFHT BOOKING APP"** is a bona fide work carried out by **ANUSHKA ROY (1JS20IS023)** and **G SANJANA REDDY (1JS20IS036)** in partial fulfillment for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University Belagavi during the year 2022-23.

Signature of the Guides

Mrs. Nagashree S,
Assistant Professor
Dept. of ISE
JSSATE, Bengaluru

Ms. Sukrutha C B,
Assistant Professor
Dept. of ISE
JSSATE, Bengaluru

Signature of the HOD

Dr. Rekha P.M.,
Professor
Dept. of ISE
JSSATE, Bengaluru

Name of the Examiners **Signature with date**

1.

2.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So, with gratitude, we acknowledge all those whose guidance and encouragement crowned our effort with success.

First and foremost, we would like to thank the **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Bhimsen Soragoan**, Principal, JSSATE, Bangalore for providing an opportunity to carry out MOBILE APPLICATION DEVELOPMENT LAB WITH MINI PROJECT WORK as a part of our curriculum in the partial fulfillment of the degree course.

We express our sincere gratitude for our Head of the department, **Dr. Rekha PM**, for co-operation and encouragement.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guide **Mrs.Nagashree S,** Assistant Professor and **Ms.Sukrutha C B,** Assistant Professor for the constant encouragement, valuable help and assistance in every possible way.

We would like to thank all **ISE department teachers** and **non-teaching staff** for providing us with their valuable guidance and for being there at all stages of our work.

Anushka Roy (1JS20IS023)

G Sanjana Reddy (1JS20IS036)

ABSTRACT

This Android flight booking app is designed to offer users a simplified and intuitive way to book flights. With a user-friendly interface, the app streamlines the flight booking process, providing a seamless experience for travelers.

Using the app, users can search for flights based on their preferred criteria such as departure and destination cities, travel dates, and the number of passengers. The app retrieves real-time flight data from reliable sources, ensuring that users have access to the most up-to-date information.

The app features various functionalities to enhance the user experience. Users can compare flight prices from different airlines, view detailed flight itineraries, and select their preferred seats. Additionally, the app provides information on baggage allowances, flight amenities, and any additional services offered by the airlines.

To simplify the payment process, the app integrates secure payment gateways, allowing users to make online payments conveniently and securely. Upon successful booking, users receive instant notifications and electronic tickets directly on their Android devices.

The app also offers features to assist travelers throughout their journey. Users can save their travel preferences, frequent flyer details, and passport information for quick and easy future bookings. Additionally, the app provides real-time flight status updates, gate information, and alerts for any flight delays or cancellations.

Furthermore, the app incorporates social sharing functionalities, enabling users to share their travel plans with friends and family through popular social media platforms.

In conclusion, this Android flight booking app simplifies the process of booking flights, providing users with a seamless and efficient experience. By leveraging real-time data, an intuitive design, and convenient features, this app empowers travelers to effortlessly plan their flights, making it an essential tool for anyone looking to book flights using their Android devices.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Overview

Welcome to our innovative flight booking app! With a user-centric design and powerful features, our app aims to revolutionize the way you book flights. Whether you're a frequent traveler or planning a spontaneous trip, our app offers a seamless and hassle-free experience, right at your fingertips.

Our intuitive and user-friendly interface makes it effortless to search for flights based on your preferences. Simply enter your departure city, destination, travel dates, and the number of passengers, and let our app do the rest. We fetch real-time flight data from reliable sources, ensuring that you have access to the most up-to-date information available.

One of the standout features of our app is the ability to compare flight prices from multiple airlines. We understand the importance of finding the best deals, and our app provides you with the flexibility to select the most affordable options without compromising on quality. In just a few taps, you can explore various flight itineraries, compare fares, and choose the one that suits your needs and budget.

When it comes to the booking process, we've made it secure and convenient. Our app seamlessly integrates with secure payment gateways, allowing you to make online payments with peace of mind. Once your booking is confirmed, you'll receive instant notifications and electronic tickets, eliminating the need for paper documents and ensuring that your travel plans are readily accessibleon your Android device.

We understand that traveling can sometimes come with uncertainties, which is why we've includedfeatures to assist you throughout your journey. Stay updated with real-time flight status information, gate details, and receive timely alerts for any changes or delays. You can also save your travel preferences, frequent flyer details, and passport information within the app, making future bookings a breeze

Additionally, our app embraces the power of social connectivity. Share your travel plans with your loved ones effortlessly through popular social media platforms, keeping them informed and involved in your exciting adventures.

We're committed to providing you with an exceptional flight booking experience. Our app combines the convenience of modern technology with a personalized touch, allowing you to take control of your travel plans and explore the world with ease. So, why wait? Download our app now and embark on a journey of seamless flight bookings, endless possibilities, and unforgettable experiences!

1.2 Problem Statement

The existing flight booking process poses several challenges and inconveniences for travelers, highlighting the need for a more efficient and user-friendly solution. The current problem lies in the lack of a streamlined and comprehensive platform that caters to the diverse needs of travelers when it comes to booking flights through mobile applications.

The primary issues faced by users include:

1. Complex and Time-Consuming Search Process: Many flight booking apps present a cumbersome search process, requiring users to navigate through multiple screens and input various details repeatedly. This complexity results in a time-consuming and frustrating experience for users who seek a quick and efficient flight search.

2. Limited Comparison Capabilities: Existing flight booking apps often lack robust comparison features, making it challenging for users to compare flight prices, schedules, and amenities across multiple airlines. This limitation hinders users from making informed decisions and finding the best options that meet their preferences and budget.

3. Lack of Real-Time Updates: Users frequently encounter difficulties in obtaining real-time flight information, including flight status updates, gate changes, and delays. The absence of timely notifications leads to confusion and inconveniences, disrupting travel plans and causing unnecessary stress for travelers.

4. Incomplete Travel Support: Many existing flight booking apps fail to provide comprehensive travel support beyond the booking stage. Users often lack access to important information such as baggage allowances, airport facilities, and additional services offered by airlines, making it challenging to plan their journey effectively.

In light of these challenges, there is a pressing need for a flight booking app that addresses these issues and provides a seamless, user-friendly, and secure platform for travelers to search, compare, book, and manage their flights efficiently. By overcoming these hurdles, the ideal flight booking app would empower users with a streamlined and hassle-free experience, revolutionizing the way flights are booked and enhancing the overall travel experience.

1.3 Motivation

The motivation behind developing a new and improved flight booking app stems from the need to address the existing challenges and limitations faced by travelers in the current mobile app landscape. By recognizing the shortcomings of existing flight booking apps, we are driven to create a solution that revolutionizes the way flights are booked and enhances the overall travel experience.

The motivation to develop this app arises from the desire to provide users with a seamless, efficient, and user-friendly platform that simplifies the flight booking process. We understand that planning a trip, whether for business or leisure, can often be time-consuming and stressful. By streamlining the search process, simplifying the comparison of flight options, and providing real- time updates, we aim to alleviate the frustrations encountered by users when booking flights.

The motivation also comes from the recognition that travelers expect a secure and reliable payment process. We understand the importance of instilling trust and confidence in our users, ensuring that their financial information is protected while making online payments. By integrating secure payment gateways, we strive to offer a seamless and worry-free payment experience.

Furthermore, our motivation lies in the aspiration to provide comprehensive travel support to users throughout their journey. We understand that a successful travel experience extends beyond the mere booking of flights. By offering information on baggage allowances, airport facilities, and additional services, we aim to empower users to plan and manage their trips effectively, ensuring a smooth and enjoyable travel experience.

Additionally, the motivation to develop this app stems from our commitment to leveraging the power of technology to enhance the travel industry. By embracing the potential of mobile applications, we seek to provide users with a convenient and personalized platform that caters to their individual travel preferences and needs.

Ultimately, our motivation is to create an exceptional flight booking app that puts the user first. We aspire to transform the way travelers book flights, providing them with a user-centric, efficient, and reliable platform that simplifies the entire process and ensures a seamless travel experience from start to finish.

1.4 MAD Technologies

In the mini project, the following mobile application development technologies are used:

- Java: Java is the primary programming language used for developing the micro-banking

  Android application. Java has been a widely adopted language for Android development due to its platform compatibility and extensive libraries and frameworks available. It provides a strong foundation for building robust and scalable applications, including banking applications. With its mature ecosystem and extensive community support, Java remains a popular choice for developing Android applications, ensuring stability and reliability for our FLIGHT-BOOKING APP.

- Android Studio: Android Studio serves as the Integrated Development Environment (IDE)

  for building the FLIGHT-BOOKING APP. It provides a comprehensive set of tools, including code editing, debugging, testing, and performance profiling, specifically tailored for Android app development.

- Data Persistence: For storing and managing banking data, the micro-banking application may

  utilize technologies such as SQLite, Room, or other database frameworks supported by Android. These technologies ensure efficient and secure data storage on the device.

- Authentication and Authorization: To ensure secure access to user accounts, the FLIGHT-

  BOOKING APP may implement authentication mechanisms such as username/password login, biometric authentication, or integration with third-party authentication providers.

- Transaction History and Reporting: To provide users with a comprehensive overview of their

  financial activities, the FLIGHT-BOOKING APP may incorporate features like transaction history tracking, generating account statements, and generating reports.

By utilizing these technologies, the FLIGHT-BOOKING Appication can offer a reliable, secure, and user-friendly platform for users to perform banking transactions, manage their finances, and access essential banking services conveniently through their mobile devices.

1.5 Applications of MAD Technologies

Mobile application development technologies have a wide range of applications across various industries and use cases. In the flight booking app, these technologies can be applied in
the following ways:

1. Flight Search and Booking: Implement a user-friendly interface that allows users to search forflights, view available options, compare prices, and make bookings directly through the app. Integration with flight APIs or GDS (Global Distribution Systems) enables real-time access to flight data and availability.

2. Mobile Payment Integration: Integrate secure mobile payment gateways, such as credit card processors or digital wallets, to facilitate seamless and secure transactions for flight bookings. Thisallows users to make payments directly within the app, providing a convenient and hassle-free experience.

3. Real-Time Flight Updates: Utilize technologies like flight tracking APIs or data feeds to provideusers with real-time updates on flight status, including departure and arrival times, delays, gate changes, and cancellations. This ensures that users have the latest information about their booked flights.

4. Personalized Flight Recommendations: Utilize machine learning algorithms to analyze user preferences, historical data, and travel patterns to offer personalized flight recommendations. Thiscan include suggestions based on preferred airlines, optimal routes, or discounted fares, enhancingthe user experience and increasing customer satisfaction.

5. Travel Itinerary Management: Enable users to manage their travel itineraries within the app, including storing flight details, hotel reservations, and car rental information. Integration with travel APIs or services allows users to access their itineraries offline and receive notifications forimportant updates or reminders.

6. Loyalty Program Integration: Integrate loyalty programs offered by airlines or travel agencies,allowing users to earn and redeem rewards or frequent flyer miles through the app. This incentivizes user engagement and encourages customer loyalty.

7. In-App Support and Communication: Implement in-app messaging or chatbot features to provide customer support and assistance throughout the flight booking process. This can includeanswering queries, resolving issues, and providing personalized recommendations based on userinquiries.

8. Social Sharing and Trip Planning: Enable users to share their flight itineraries and travel plans with friends and family through social media integration. Additionally, provide features for collaborative trip planning, allowing users to invite others to join their trips and coordinate travelarrangements.

By leveraging mobile application development technologies, a flight booking app can offer users aseamless and efficient way to search, book, and manage their flights, enhancing the overall travel experience.

CHAPTER 2
SYSTEM REQUIREMENTS

1. Hardware and Software Requirements

Hardware Requirements:

- Android Device: The application will be developed for Android devices. Any Android smartphone or tablet with a camera and sufficient processing power should be capable of running the application smoothly. The specific Android version supported by the application should be determined during the development process.
- Camera: The Android device should have a functioning camera for capturing photos. The quality and resolution of the camera may vary across devices, but a reasonably capable camera should be sufficient for text extraction purposes.

Software Requirements:

- Android SDK: The Android Software Development Kit (SDK) is necessary for developing Android applications. It includes tools, libraries, and resources required to build, test, and debug Android apps. The SDK provides APIs for interacting with device features like the camera and internet connectivity.
- Java Programming Language: The application will be developed using the Java programming language. Java has long been the standard language for Android development and is fully compatible with the Android platform. It provides a robust and reliable foundation for building secure and scalable applications. With its extensive libraries, frameworks, and toolsets, Java offers developers a wide range of resources to efficiently develop and maintain Android applications.
- Android Studio: Android Studio provides a comprehensive set of tools and features, including code editor, emulator, debugging tools, and project management capabilities. Android Studio simplifies the development process by offering a user-friendly interface and seamless integration with the Android SDK.

CHAPTER 3
SYSTEM DESIGN

3.1 Proposed System

The proposed Flight booking Android application developed in Java focuses on providing users with convenient and secure access to booking through their mobile devices. The key features of the proposed system are as follows:

- User-Friendly Interface: The flight booking app will have a user-friendly interface designed to simplify the flight search and booking process. It will provide clear navigation options for searching flights, selecting seats, and completing the booking process. The interface will prioritize ease of use and intuitive design to ensure a
  - seamless flight booking experience.

- Secure Authentication: The app will implement robust security measures to protect user information and transactions. It will include secure login credentials and may incorporate additional security measures such as two-factor authentication to ensure only authorized users can access the app and make flight bookings.

- Flight Search and Booking: Users will be able to search for flights based on their desired destinations, travel dates, and other preferences. The app will provide a comprehensive list of available flights with details on airlines, prices, and schedules. Users can select their preferred flights and complete the booking process securely within the app.

- Mobile Payments: The flight booking app will integrate with mobile payment gateways to facilitate secure and convenient transactions. Users can make payments for their flight bookings using various payment methods, such as credit cards, digital wallets, or mobile payment apps.

- Notifications and Alerts: The app will send real-time notifications and alerts to users for important updates regarding their flight bookings.

Customer Support: The app will provide customer support features, such as in-app messaging or chatbot assistance, to address any queries or concerns users may have regarding their flight bookings. Users can seek help, ask questions, and receive prompt responses to ensure a positive customer experience.

- Trip Management: The app will allow users to manage their flight itineraries within the app. Users can view and access their booked flights, make changes or cancellations if applicable, and store travel details such as e-tickets or boarding passes. The app may also provide features for adding hotel or car rental information to create a comprehensive trip itinerary.

- By incorporating these key features, the flight booking app will provide users with a convenient and user-friendly platform to search, book, and manage their flights, ensuring a seamless and enjoyable travel experience.

.

3.2 Flow of Activity

After opening the application (refer fig 3.1), there will be a pop up of a splash screen that will display the logo of our application for a while, and a page will appear. It will have a login form that requires an email address and a password to be filled (refer fig 3.2). If the credentials are correct, you will be able to access the next page, and if you are a new user, you may have to sign up for the app. You will need to include the necessary information in the signup page such as email address, password. After filling up the form, you'll have to submit it, and by clicking on login button it will redirect you to a login page. If you feed the wrong information on the login page, it will provide you a toast message of invalid login credentials. Otherwise, it will display a success toast message and display the next activity which is the main page. The second screen consists of a book a ticket or check the booking or you can cancel the booking (refer fig 3.3).

When you click on the book button you will be redirected to another page which would display the menu to search for your desired from and to location followed by the date with a calendar menu.(refer fig 3.4) After selecting the locations and date the page would be redirected to a new page where you would see the available flights(refer fig 3.5)

After completing the booking you can go back and check your current bookings and it would display ypur current booking. You also have an option tp cancel your existing booking (refer fig 3.6)

Fig 3.1 Sign-up page

Fig 3.2 Log-in page

Fig 3.3 Home page

Fig 3.4 ticket booking

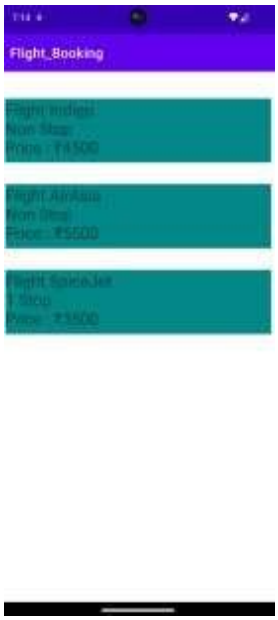

Fig 3.5 available flights

Fig 3.6 final ticket

CHAPTER 4
IMPLEMENTATION

4.1 Module Description

LoginActivity Module

The "LoginActivity" module is a key component of an Android application that facilitates user authentication and login functionality. From a technical standpoint, the module imports necessary dependencies and extends the "AppCompatActivity" class. It declares variables to reference user interface elements like EditText, Button, and TextView.

In terms of functionality, the module allows users to enter their username and password. When the login button is clicked, the module retrieves the entered credentials and compares them to valid credentials stored as constants. If the credentials match, the user is considered authenticated and is redirected to the MainActivity. In case of invalid credentials, an error message is displayed, and the input fields are cleared.

The LoginActivity module also handles the signup process. When the signup TextView is clicked, it creates an intent to launch the SignupActivity, providing users with the option to create a new account. The module includes a private method, validateCredentials(), which performs the validation of the entered credentials against the predefined valid credentials.

From a functional perspective, the LoginActivity module ensures that only authorized users can access the application's features and resources. It enhances security by validating the provided credentials and prevents unauthorized access. The module also provides a smooth user experience by displaying error messages for invalid credentials and clearing input fields for retry.

Overall, the LoginActivity module combines technical implementation, such as variable declarations and event listeners, with functional aspects like authentication and navigation, to create a seamless and secure login process within the Android application.

SignupActivity Module

The "SignupActivity" module is an essential part of the Android application and is responsible for handling user signup functionality. The module consists of variable declarations for UI elements like EditText and Button. It allows users to enter their desired username and password for creating a new account.

From a technical standpoint, the SignupActivity imports necessary dependencies and extends the "AppCompatActivity" class. It initializes the UI elements in the onCreate() method and sets up click listeners for the signup button and back button.

When the signup button is clicked, the module retrieves the entered username and password from the EditText fields. It creates an intent to launch the LoginActivity and passes the username and password as extras. This allows the LoginActivity to receive and process the new user's credentials for further authentication. After launching the LoginActivity, the SignupActivity finishes its own instance.

The back button click listener creates an intent to navigate back to the LoginActivity, providing users with an option to return to the login screen without signing up.

Functionally, the SignupActivity module enables users to create a new account by entering a username and password. It enhances user experience by providing a smooth transition to the LoginActivity after successful signup. From a security perspective, the module facilitates the creation of new user accounts, allowing for personalized access to the application's features and resources.

Overall, the SignupActivity combines technical implementation, such as variable initialization and intent creation, with functional aspects like user account creation andnavigation, to deliver a seamless and intuitive signup process within the Android application.

.

4.2 Source code

LoginActivity.java

```
package com.example.flight_booking;

import androidx.appcompat.app.AppCompatActivity;import

android.content.Intent;
import android.os.Bundle; import
android.view.View; import
android.widget.Button; import
android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class login extends AppCompatActivity {

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

        TextView register = findViewById(R.id.register_link);
        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(login.this,registration.class);
                startActivity(intent);
            }
        });

        myDatabaseHelper dbHelper = new myDatabaseHelper(this);EditText

        username1 = findViewById(R.id.username);
        EditText password1 = findViewById(R.id.password);

        Button login_btn = findViewById(R.id.login_btn);

        login_btn.setOnClickListener(new View.OnClickListener() {
            @Override

       public void onClick(View view) {

                    String username = username1.getText().toString(); String
                    password = password1.getText().toString(); Boolean user_b =
                    dbHelper.checkUsername(username);Boolean pass_b =
                    dbHelper.checkPassword(password);
```

```
                if(username.equals("") || password.equals(""))
                {
                    Toast.makeText(login.this, "All fields are required!",
    Toast.LENGTH_SHORT).show();
                }
                else
                {
                    if(user_b == true)
                    {
                        if(pass_b == true)
                        {
                            Toast.makeText(login.this, "Login Successfull!",
    Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent(login.this, mainscreen.class);
                            intent.putExtra("username",username);
                            intent.putExtra("password",password); startActivity(intent);
                        }
                        else
                        {
                            Toast.makeText(login.this, "Password incorrect!",
    Toast.LENGTH_SHORT).show();
                        }
                    }
                    else
                    {
                        Toast.makeText(login.this, "User not Exist! Register Now!",
    Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });

    }
}
```

MainActivity.java

```
package com.example.flight_booking;

import androidx.appcompat.app.AppCompatActivity;import

android.content.Intent;
import android.os.Bundle; import
android.view.View; import
android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);Button
        go = findViewById(R.id.go);
        go.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this,login.class);
                startActivity(intent);
            }
        });
    }
}
```

List_Of_Flights.java

```
package com.example.flight_booking;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class list_of_flight extends AppCompatActivity {

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

        TextView indigo = findViewById(R.id.indigo);
        TextView airasia = findViewById(R.id.airasia);
        TextView spicejet = findViewById(R.id.spicejet);

        String username = getIntent().getStringExtra("username");
        String password = getIntent().getStringExtra("password");
        String from = getIntent().getStringExtra("from");
        String to = getIntent().getStringExtra("to");
        String date = getIntent().getStringExtra("date");

        indigo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { String
                s = indigo.getText().toString();
                s += "\nFrom : "+from+"\nTo : "+to+"\nDate : "+date;
                Intent intent = new Intent(list_of_flight.this,book_final.class);
                intent.putExtra("value",s);
                intent.putExtra("username",username);
                intent.putExtra("password",password);
                startActivity(intent);
            }
        });
```

```
        airasia.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { String
                s = airasia.getText().toString();
                s += "\nFrom : "+from+"\nTo : "+to+"\nDate : "+date;
                Intent intent = new Intent(list_of_flight.this,book_final.class);
 intent.putExtra("value",s);
                intent.putExtra("username",username);
                intent.putExtra("password",password);
                startActivity(intent);
            }
        });
        spicejet.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String s = spicejet.getText().toString();
                s += "\nFrom : "+from+"\nTo : "+to+"\nDate : "+date;
                Intent intent = new Intent(list_of_flight.this,book_final.class);
                intent.putExtra("value",s);
                intent.putExtra("username",username);
                intent.putExtra("password",password);
                startActivity(intent);
            }
        });
    }
}
```

Book_Final.java

```
package com.example.flight_booking;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class book_final extends AppCompatActivity {

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

        myDatabaseHelper dbHelper = new myDatabaseHelper(this);

        String s = getIntent().getStringExtra("value");
        String username = getIntent().getStringExtra("username");
        String password = getIntent().getStringExtra("password");
```

```
        TextView t = findViewById(R.id.value);
        t.setText(s);


 Button book = findViewById(R.id.book);

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

                int b =
dbHelper.insert_history(username,password,s);
                if(b == 0) {
                    Intent intent = new
Intent(book_final.this,payment.class);
                    startActivity(intent);
                }
                else
                    Toast.makeText(book_final.this, "BookingFailed",
Toast.LENGTH_SHORT).show();

            }
        });


    }
}
```

Payment.java

```
package com.example.flight_booking;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View; import
android.widget.Button; import
android.widget.TextView;import
android.widget.Toast;

public class payment extends AppCompatActivity {
 @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_payment);

       TextView gpay = findViewById(R.id.gpay);
       TextView phonepe = findViewById(R.id.phonepe);
       TextView paytm = findViewById(R.id.paytm);
       TextView credpay = findViewById(R.id.credpay);
       Button pay = findViewById(R.id.pay);
```

```
gpay.setOnClickListener(new
    View.OnClickListener() {@Override
    public void onClick(View
      v) {gpay.setText("Pay
      Using");
      gpay.setBackground(Drawable.createFromPath("@drawable/grey"));
    }
  });
  phonepe.setOnClickListener(new
    View.OnClickListener() {@Override
    public void onClick(View v) {
      phonepe.setText("Pay Using");
      phonepe.setBackground(Drawable.createFromPath("@drawable/gre
      y"));
    }
  });
  paytm.setOnClickListener(new
    View.OnClickListener() {@Override
    public void onClick(View
      v) {paytm.setText("Pay
      Using");
      paytm.setBackground(Drawable.createFromPath("@drawable/grey"));
    }
  });
  credpay.setOnClickListener(new
    View.OnClickListener() {@Override
    public void onClick(View
      v) { credpay.setText("Pay
      Using");
      credpay.setBackground(Drawable.createFromPath("@drawable/grey"));
    }
  });
  pay.setOnClickListener(new
    View.OnClickListener() {@Override
```

```
        public void onClick(View v) {
           Toast.makeText(payment.this, "Paid Successfully!!!", Toast.LENGTH_SHORT).show();

//            Intent intent = new Intent(payment.this,mainscreen.class);
//            startActivity(intent);
        }
      });
  }}
```

SpaceTokenizer.java

```
package com.example.flight_booking;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View; import
android.widget.Button; import
android.widget.TextView;import
android.widget.Toast;
public class payment extends AppCompatActivity {

  @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_payment); TextView
       gpay = findViewById(R.id.gpay); TextView phonepe =
       findViewById(R.id.phonepe);TextView paytm =
       findViewById(R.id.paytm); TextView credpay =
       findViewById(R.id.credpay); Button pay =
       findViewById(R.id.pay);

      package com.example.flight_booking;

import android.widget.MultiAutoCompleteTextView;

public class SpaceTokenizer implements MultiAutoCompleteTextView.Tokenizer {
   @Override
   public int findTokenStart(CharSequence text, int cursor) {
```

```
 int i = cursor;
        while (i > 0 && text.charAt(i - 1) != ' ') {i-
            -;
        }
        return i;
    }

    @Override
    public int findTokenEnd(CharSequence text, int cursor) {
        int i = cursor;
        int len = text.length();

        while (i < len) {
            if (text.charAt(i) == ' ') {
                return i;
            } else {
                i++;
            }
        }
        return len;
    }

    @Override
    public CharSequence terminateToken(CharSequence text) {
        return text;
    }
}
```

CHAPTER 5
RESULTS

SignupActivity

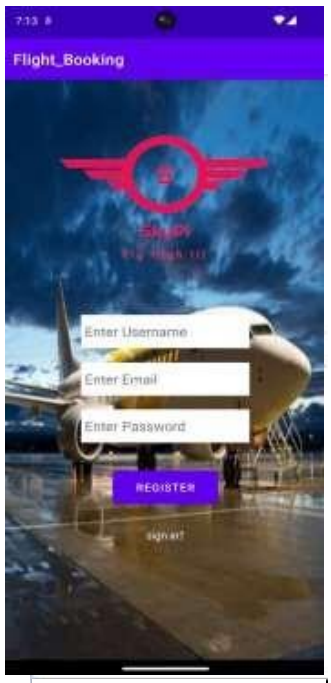

Fig 5.1 Sign-up

The sign-up page offers a simple user interface as seen in the fig 5.1 and has 3 plaintext boxes, one for entering username and the other for password.

The user has to fill in the username, email password fields with valid strings and click on signup button, upon clicking the signup button the user's credentials are now saved and the user can proceed to LoginActivity page by clicking on the 'register' button.

LoginActivity

Fig 5.2 Login page

The login page also contains 2 plaintext boxes, 1 button and a text view as shown in fig 5.2 The user must enter his/her credentials as entered in the signup page for a successful login. Upon entering valid details and clicking on the login button the user gets to see a toast message saying "login successful" and is directed to second page which is the MainActivity module or else the user will see a toast message saying "invalid credentials" and the application remains in the same page.

There is also a text view that says "Register", upon clicking on this textview the user is directed to the signup activity where the user can create an account by specifying the necessary credentials.

MainActivity

Fig 5.3 Main page

This page consists of 1 textview and 3 buttons as shown infig 5.3.

Buttons: These are used to select one of the three options. That is to book a flight, or to cancel abooking or one can see the existing booking TextViews:Text view displays the header of the app as “Flight Booking App’

Fig 5.4 Ticket Page

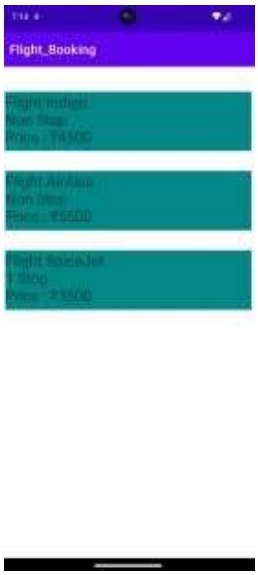

Fig 5.5 Available Flights

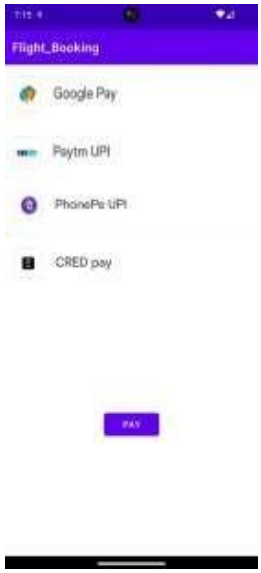

Figure 5.6 Payment Page

After the user selects the option to book a flight he is redirect to a page where he should choosehi to and from with a specific date as shown in (fig 5.4)

Once the user selects his destination, he is redirected to a new window where the availableflights would be displayed as shown in (fig 5.5)

After selecting one of the available flights the user can choose one of the payment options and complete the transaction as shown in (fig 5.6)

Fig 5.7 Booked Ticket

On the Check booking page, the details of the existing booking will be displayed as shown in the figure 5.7.

CONCLUSIONS

In conclusion, a well-designed and feature-rich flight booking app can greatly enhance the travel experience for users. By incorporating various functionalities and improvements, such as a user- friendly interface, faster search and booking processes, and personalized recommendations, the app can simplify the flight booking process and provide convenience at every step. The integration of technologies like virtual reality, artificial intelligence, and biometric authentication can further elevate the app's capabilities, offering immersive experiences, personalized assistance, and enhanced security measures. These advancements have the potential to revolutionize how users search for flights, make bookings, and manage their travel itineraries.

Additionally, by focusing on simplicity and ease of use, flight booking apps can cater to a wider audience, including users with varying levels of technical expertise. Clear and intuitive interfaces, along with offline access and multi-language support, ensure that users can effortlessly access their travel information and make bookings even in diverse and unpredictable situations. Furthermore, by embracing social integration and sustainability considerations, flight booking apps can foster a sense of community among travelers and encourage more environmentally conscious travel choices.

Ultimately, a successful flight booking app combines technological innovation, user-centric design, and seamless functionality to create a comprehensive platform that meets the evolving needs of travelers. By continually adapting to emerging trends and leveraging advancements in technology, flight booking apps can shape the future of travel, providing users with a more efficient, personalized, and enjoyable booking experience.

FUTURE ENHANCEMENTS

1. User-Friendly Interface: Streamline the app's user interface to make it more intuitive, visually appealing, and easy to navigate, ensuring a seamless booking experience for users.

2. Faster Search and Booking: Optimize the search and booking process to minimize loading times and provide real-time availability and pricing information, allowing users to book flights quickly and efficiently.

3. Saved Travel Preferences: Enable users to save their travel preferences, such as seat preferences, meal choices, and frequent flyer information, making it faster and more convenient to book future flights.

4. Push Notifications: Implement push notifications to provide users with timely updates on flight status, gate changes, delays, and other relevant information, ensuring they stay informed throughout their journey.

5. One-Click Payments: Integrate secure and convenient one-click payment options, such as digital wallets or saved credit card information, to expedite the payment process and reduce the need for manual entry.

6. Multi-Language Support: Offer support for multiple languages to cater to a diverse user base, allowing users to interact with the app in their preferred language.

7. Offline Access: Enable users to access their flight details, itineraries, and e-tickets even when they are offline, ensuring they can retrieve important travel information without an internet connection.

8. In-App Customer Support: Provide in-app customer support options, such as live chat or direct messaging, allowing users to easily reach out for assistance or resolve any issues they may encounter during the booking process.

REFERENCES

- https://developer.android.com
- https://stackoverflow.com
- https://www.youtube.com
- https://www.udemy.com