PODCAST PLUS: A REDUX-INSPIRED PODCAST APP WITH DYNAMIC THEMES FOR ANDROID

A PROJECT

Submitted by

B.SURENTHAR: (20201231506239)

S.STHISHKUMAR: (20201231506233)

S.ARUNSUBRAM: (20201231506203)

V.MARIMUTHU: (20201231506223)

MENTOR
Dr.T.ARUL RAJ M.Sc.,M.Phil.,Ph.D

in partial fulfillment of the requirements for the award of degree of

BACHELOR OF SCIENCE

(Computer Science)

SRI PARAMAKALYANI COLLEGE
ALWARKURICHI – 627 412
APRIL - 2023

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW

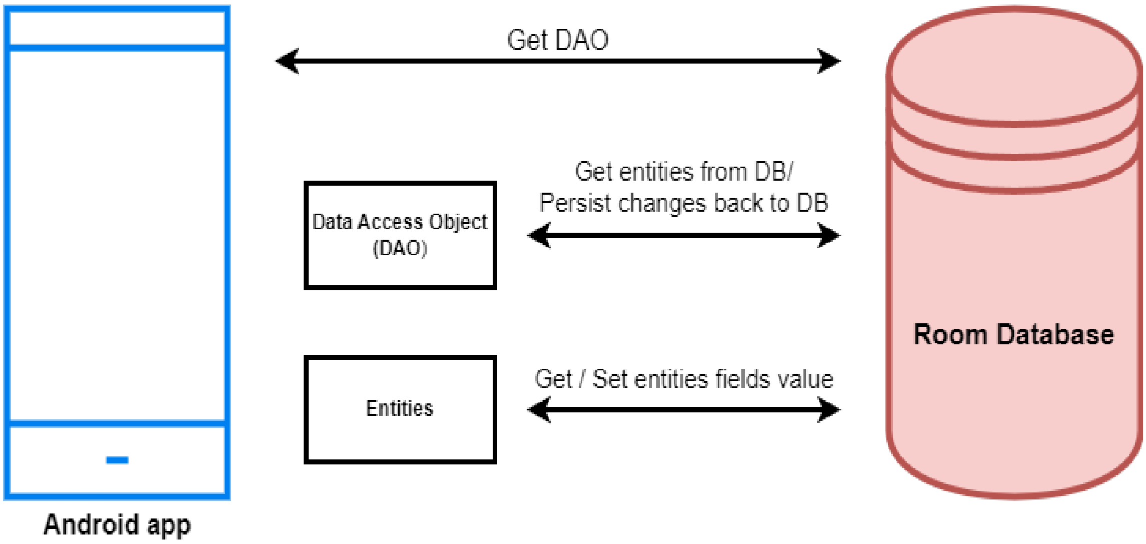
Wanderlust is a term used to describe a strong desire or impulse to travel and explore new places. It is often associated with a sense of adventure and a willingness to embrace the unknown. People who experience wanderlust may feel a deep yearning to see the world and experience different cultures, foods, and ways of life.

A project that demonstrates use of android jet pack compose to build a UI for wonderlust traveling app.The travelling app is a simple project built using the android compose UI tool kit.Wonderlust is a travel app designed to help users plan and organize.

The app provides a comprehensive platform for users to discover new destinations, browse through travel guides, book flights and accommodations, create itineraries, and connect with other travelers. With

Wonderlust, users can easily search for flights and hotels, compare prices, and make bookings directly from the app. The app also offers personalized recommendations based on the user's travel history and preferences, making it easier for users to plan their trips. Additionally,

Wonderlust provides access to a wide range of travel guides, including insider tips, restaurant recommendations, and local attractions. Whether you're planning a weekend getaway or a long-term adventure, Wonderlust is the perfect travel companion.


Get DAO
Data Access Object (DAO)
Get entities from DB/ Persist changes back to DB
Entities
Get / Set entities fields value
Room Database
Android app

A lot of thought process goes into the development of any software which has teams of people working on a piece of software for months. The team of people working have to face a lot of problems, some of them being adjusting with each other, recognizing which part is done best by whom and so on. In order to give us a taste of all the difficulties of such a task under the course CSP 301 we were given a task of developing a site on the theme INCREDEBLE INDIA. The task was certainly quite an interesting one, one that gave us a feel of the task of actual software development. The task involved learning the use of diverse elements like the FLICKR API along with the GOOGLE API and combining all the elements.

It certainly was a big learning curve for our group. Even though we did face many difficulties ultimately the careful planning and the dedication of all the team members ensured that we were able to complete all our work. In the documentation we try and list the complete process that went into development of this website.

Ranging from a HOW TO USE MANUAL to listing the difficulties we tried to present a complete picture of all the work that went into making the site into a complete structure. In the end all the group members felt that we needed to thank Prof. Anirban Mahanti for giving us this platform which helped us to work as a cohesive force and recognize the meaning of the word TEAM.

A country with natural beauty and historical places, is visited by many diverse. The most dominant purpose is tourism. The country was listed by Lonely Planet in 2011 as the "best value destination"[1]. There are some other reasons to travel the country such as political, business, official, and education purpose, and so on.

However, being foreigners in Bangladesh, they face some challenges during residing in the country such as lack of getting language and transport information. For meeting up some of the requirements by the travelers, we have come up with an online solution by developing android application.The chapter describes the objective of our project. Besides, it describes the scope and limitation of our system

The number of people interested in tourism is increasing day by day. They travel to Bangladesh for different purposes such as visiting, religious, job, business and so on. Statistics indicates "Tourism is the strongest and largest industry in the global economy world, generating an estimated 11% of the global gross domestic product (GDP) and employing 200 million people and serving 700 million tourists worldwide-a figure which is expected to double by the year 2020"

Travelers' Requirements

Although people can get some general information regarding traveling over the internet, it is sometimes problematic for the newcomers in a place to get familiar with the new environment. Basically, they face difficulties in communicating and finding proper routing information and associated costs for distinct routes.

Android Platform to Meet the Requirements

Nowadays people have been moved so much into the modern technology that they really want an intelligent living environment along with intelligent objects which contain powerful infrastructure with the most desired features. Thus android mobile applications have become very popular among the smartphone users.

Introduction to Proposed System

Considering travelers‟ requirements and popularity of android device and app, we have come up with the idea of making a mobile app for the foreigners and newcomers in one of the most attractive and tourist cities in Bangladesh which is named Chittagong, and our chosen mobile platform is Android which is open source, developed and distributed by Google.

Architechture of the Proposed System

The System has two types of interfaces. One is for transportation information and the other describes the phrases that are commonly used. The total architechture has been described below through fidelity prototype of the system

1.2 PURPOSE

A Wanderlust travel app is a mobile application that allows users to plan and organize their trips.The app provides information about popular tourist destinations, including reviews, ratings, and photos.Users can create personalized itineraries for their trips, adding flights, hotels, activities, and restaurants.The app can also provide users with recommendations for things to do places to eat, and hotels to stay at based on their preferences.Users can also access maps, directions, and weather forecasts for their destinations

The app can provide users with real-time updates and alerts, such as flight delays or changes in the weather.The purpose of wanderlust is to satisfy the innate human desire for exploration, adventure, and new experiences.Wanderlust can drive people to explore new place.

- ❖ Users can share their trip itineraries and experiences with friends family on social media. The app can also offer additional features, such as currencyconverters, language translation, and travel insurance.Some the key features of Wonderlust Travel includeDestination Inspiration:

- ❖ Users canbrowse through a collection of curated travel destinations,

- ❖ Sorted by theme or region, to get ideas and inspiration .Trip Planning: Once users have decided on a destination, they can use the app to plan their trip in detail, including booking flights, accommodation, and activities.

- ❖ The app can also generate a suggested itinerary based on the user's preferences and constraints, such as budget or travel dates.ravel Tracking: Wonderlust Travel allows users to track their travel itinerary and get reminders for upcoming bookings or activities. Users can also access maps and guides for their destination, and save important information like flight details and hotel reservations.

- Social Networking: Users can connect with other travelers, share their travel experiences, and get recommendations from a community of like-minded individualsThe app also allows users to create and join travel groups, plan trips together, and share photos and stories.

The following is the process through which we planned our development.

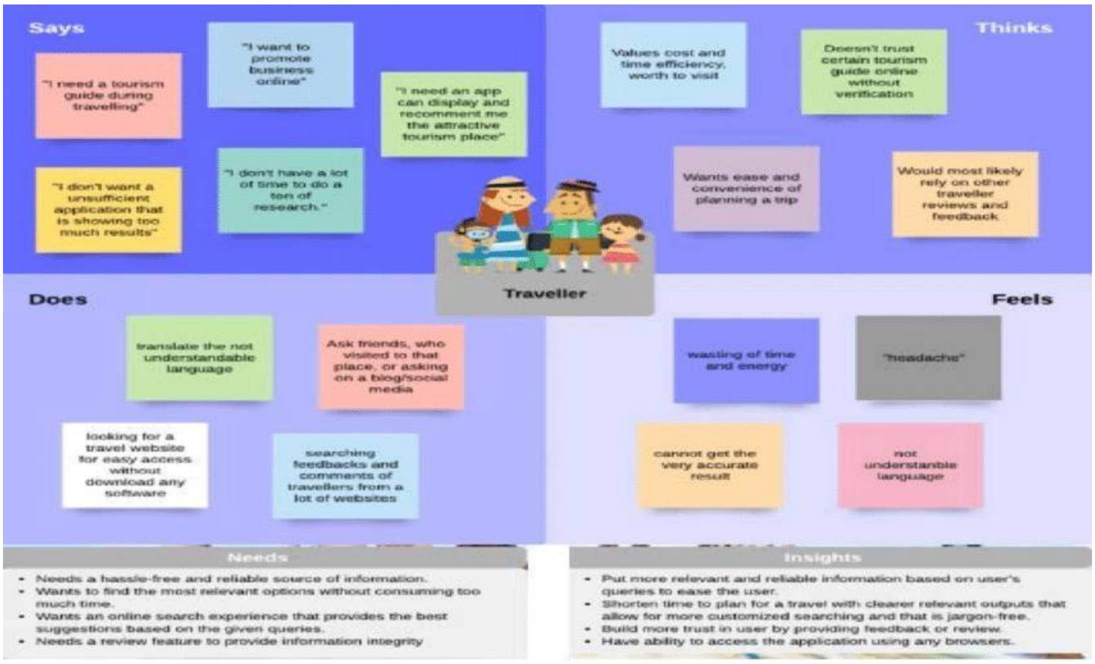
- Planning and assigment of reading work to individuals
- Reading up material for new technologies to be used
- Deciding what technologies to use and distribution of work among
- group members
- Individual work on independent components
- Integration of the core services offered.
- Embedding of core code into a standard html tempelate, create links
- among all of all static and dynamic pages.
- 7. Putting it all together in the form of a software package.

CHAPTER – II

PROBLEM DEfINITIONS & DESIGN THINKING

2.1 Empathy Map

Ideation is the creative process of generating, developing, and communicating new ideas, where an idea is understood as a basic element of thought that can be either visual, concrete, or abstract. Ideation comprises all stages of a thought cycle, from innovation, to development, to actualization..


Says
"I need a tourism guide during travelling"
"I want to promote business online"
"I need an app can display and recomment me the attractive tourism place"
"I don't want a unsufficient application that is showing too much results"
"I don't have a lot of time to do a ton of research."
Thinks
Values cost and time efficiency, worth to visit
Doesn't trust certain tourism guide online without verification
Wants ease and convenience of planning a trip
Would most likely rely on other traveller reviews and feedback
Traveller
Does
translate the not understandable language
Ask friends, who visited to that place, or asking on a blog/social media
looking for a travel website for easy access without download any software
searching feedbacks and comments of travellers from a lot of websites
Feels
wasting of time and energy
"headache"
cannot get the very accurate result
not understanble language
Needs
Insights

Needs

- Needs a hassle-free and reliable source of information.
- Wants to find the most relevant options without consuming too much time.
- Wants an online search experience that provides the best suggestions based on the given queries.
- Needs a review feature to provide information integrity

Insights

- Put more relevant and reliable information based on user's queries to ease the user.
- Shorten time to plan for a travel with clearer relevant outputs that allow for more customized searching and that is jargon-free.
- Build more trust in user by providing feedback or review.
- Have ability to access the application using any browsers.

Ideation can be conducted by individuals, organizations, or crowds. As such, it is an essential part of the design process, both in education and practice
An Activity on an Android project is the elements which interact with the user [5]. All activities are organized in classes which place inside src folder. Every android application has MainActivity.java as the main activity class.

Main Activity In MainActivity.java class, Intent is using to navigate to users‟ choice. An Intent is a class that is used to describe an operation to be performed. It is the primary way which helps developers start new activity within an application. It can also used as a tool to communicate between activities [4].

ideation & Brainstorming map

2.2 Brainstorming map

- Brainstorming is a group creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its members.

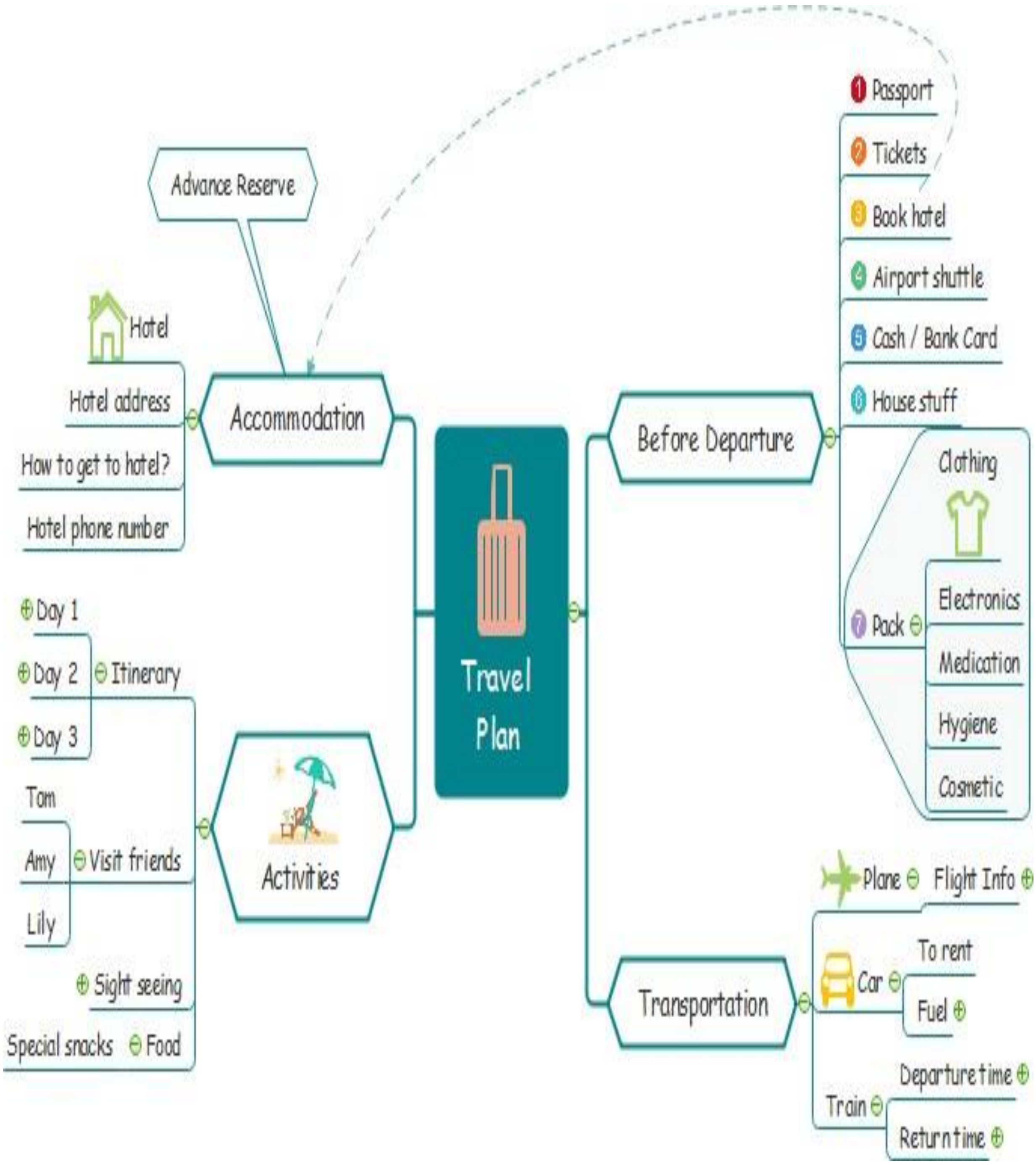
- In other words, brainstorming is a situation where a group of people meet to generate new ideas and solutions around a specific domain of interest by removing inhibitions.

- People are able to think more freely and they suggest as many spontaneous new ideas as possible.

- All the ideas are noted down without criticism and after the brainstorming session the ideas are evaluate

The phrases that we have developed so far are not real time interactive. Therefore, it might not be that meaningful to the target users (foreigners) as it"s not helping them in real time conversation through language translation.

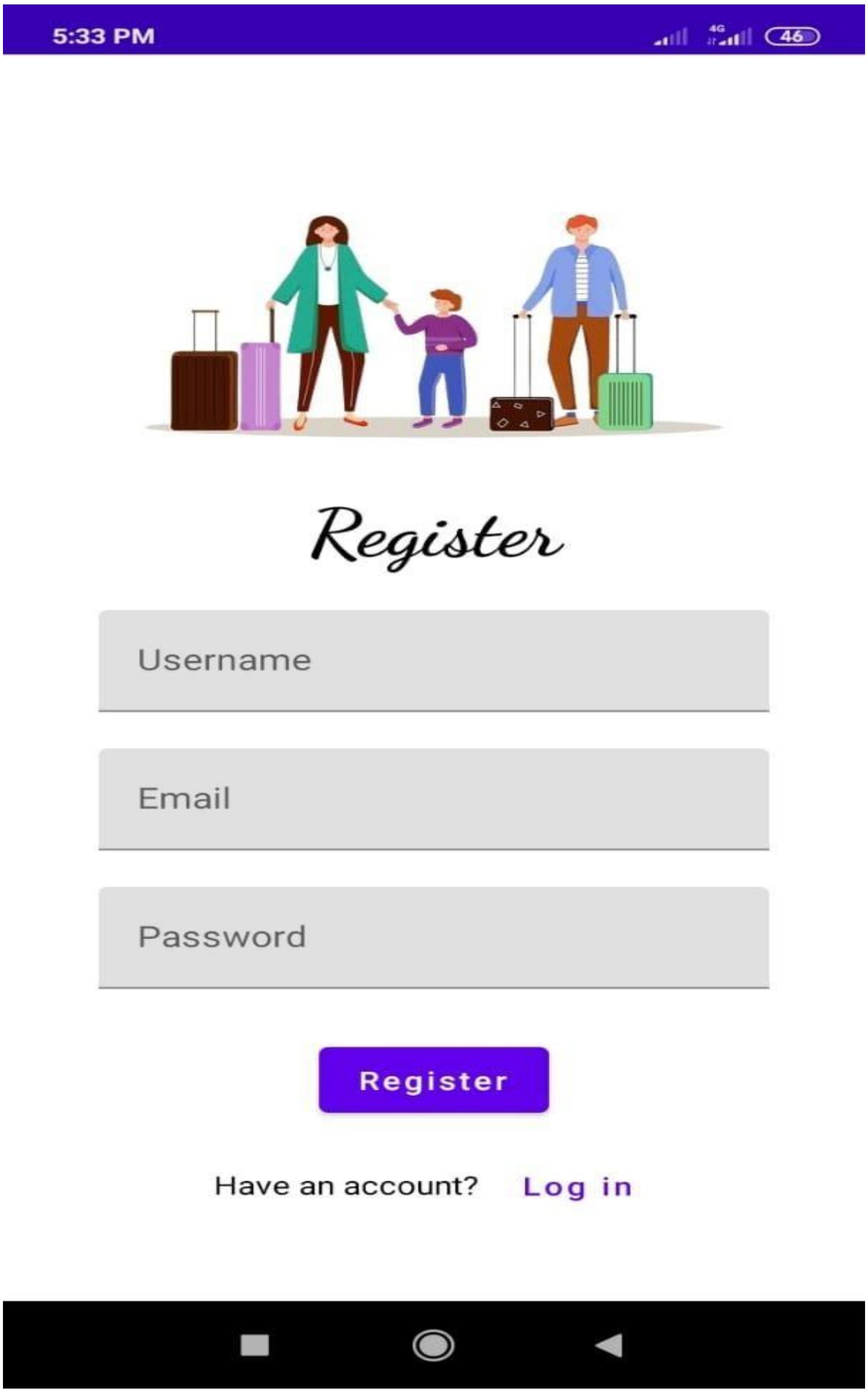
Another issue is that we are not keeping track of the user"s current location through GPS location tracker. Here in our system, a user has to provide his current location and destination to the map which is not that efficient. Furthermore, there may be different number of local buses around the city which information the user doesn"t know but really need to know so that he/she can have the idea of local buses going to different routes.


Travel Plan
Accommodation
Advance Reserve
Hotel
Hotel address
How to get to hotel?
Hotel phone number
Activities
Itinerary
Day 1
Day 2
Day 3
Visit friends
Tom
Amy
Lily
Sight seeing
Food
Special snacks
Before Departure
1 Passport
2 Tickets
3 Book hotel
4 Airport shuttle
5 Cash / Bank Card
6 House stuff
7 Pack
Clothing
Electronics
Medication
Hygiene
Cosmetic
Transportation
Plane
Flight Info
Car
To rent
Fuel
Train
Departure time
Return time

CHAPTER – III

RESULT

REGISTER PAGE



5:33 PM

Register

Username

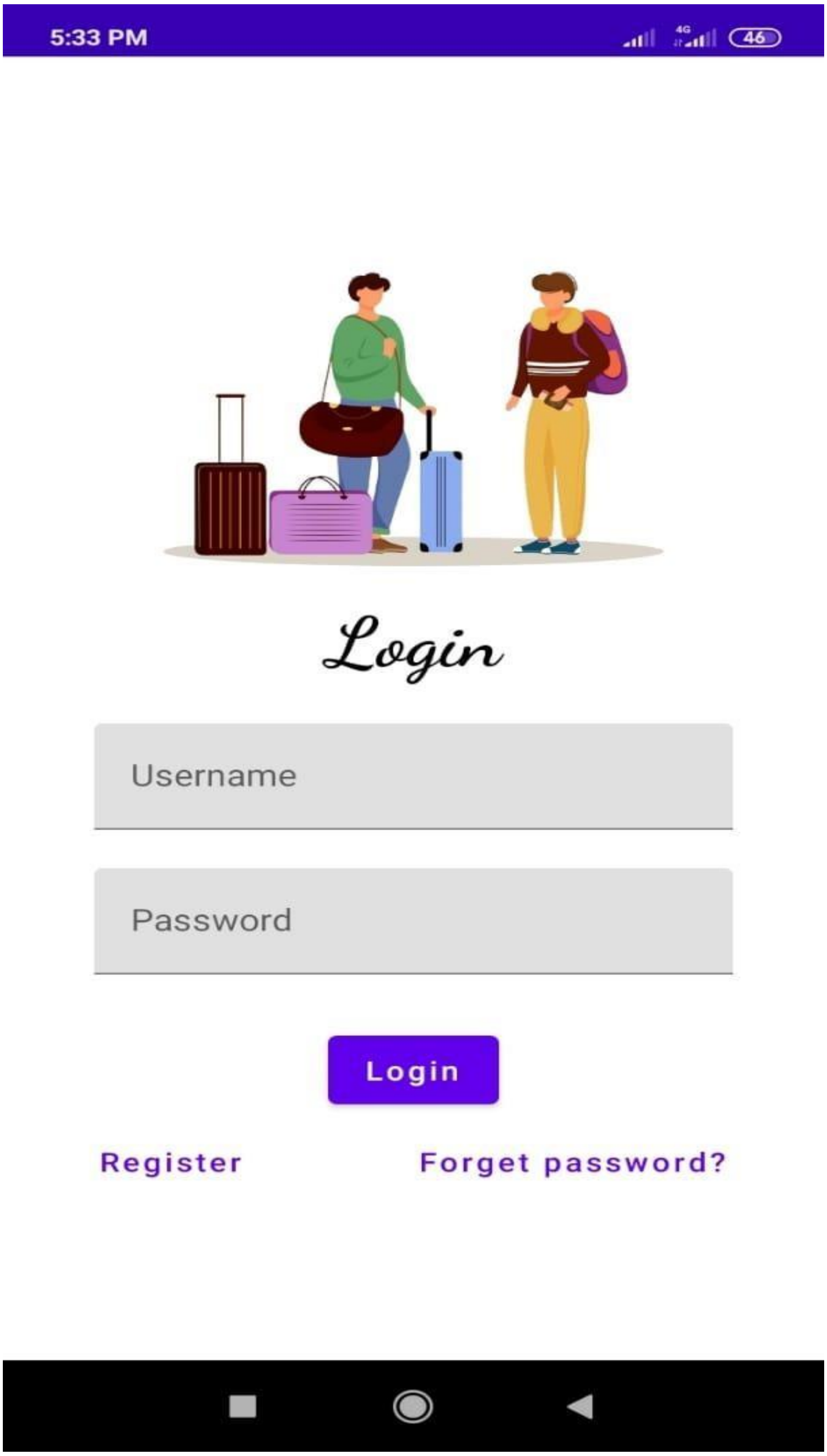
Email

Password

Register

Have an account? Log in

LOGIN PAGE



5:33 PM

Login

Username

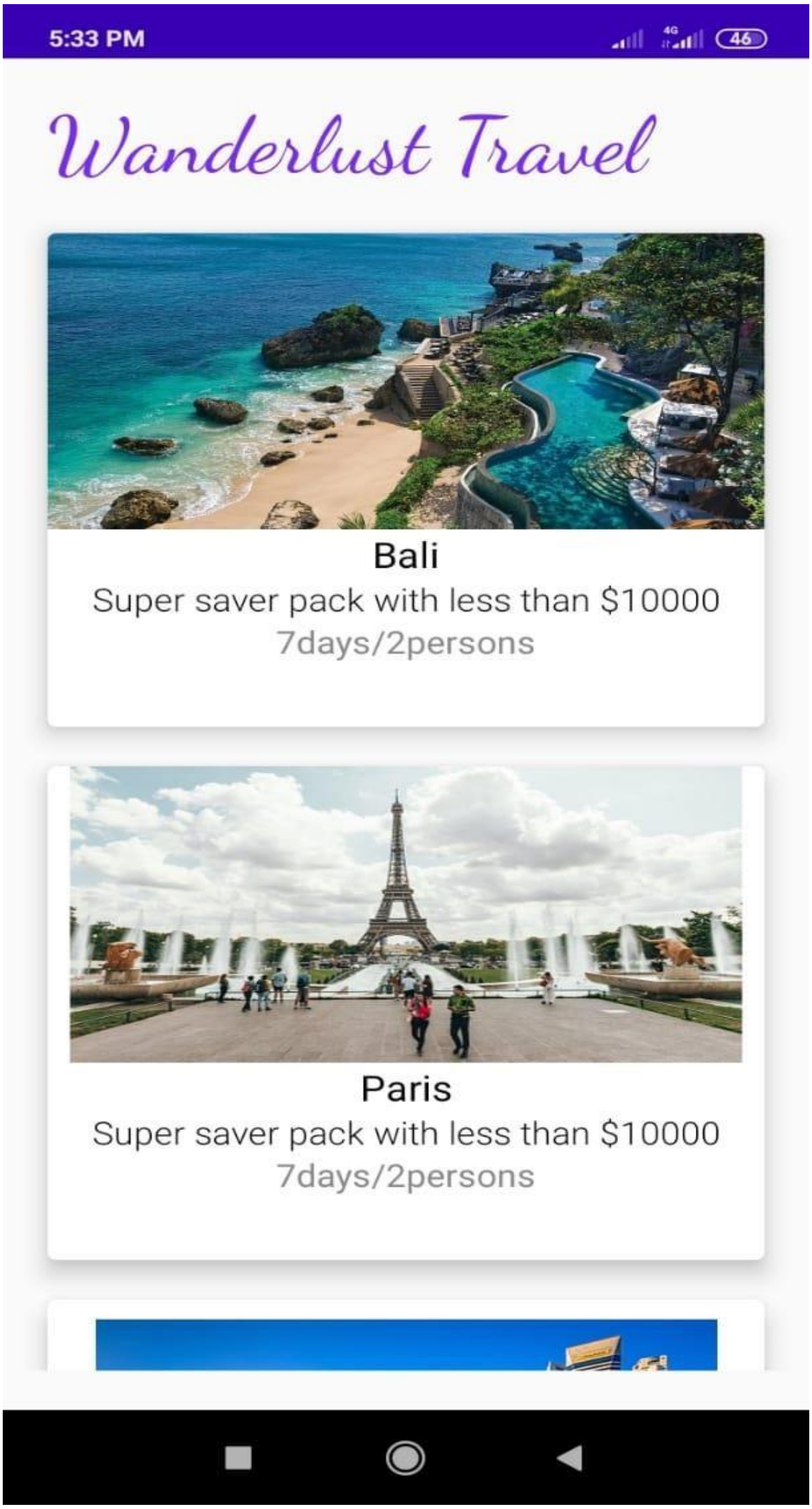
Password

Login

Register

Forget password?

MAIN PAGE



5:33 PM

Wanderlust Travel

Bali

Super saver pack with less than $10000

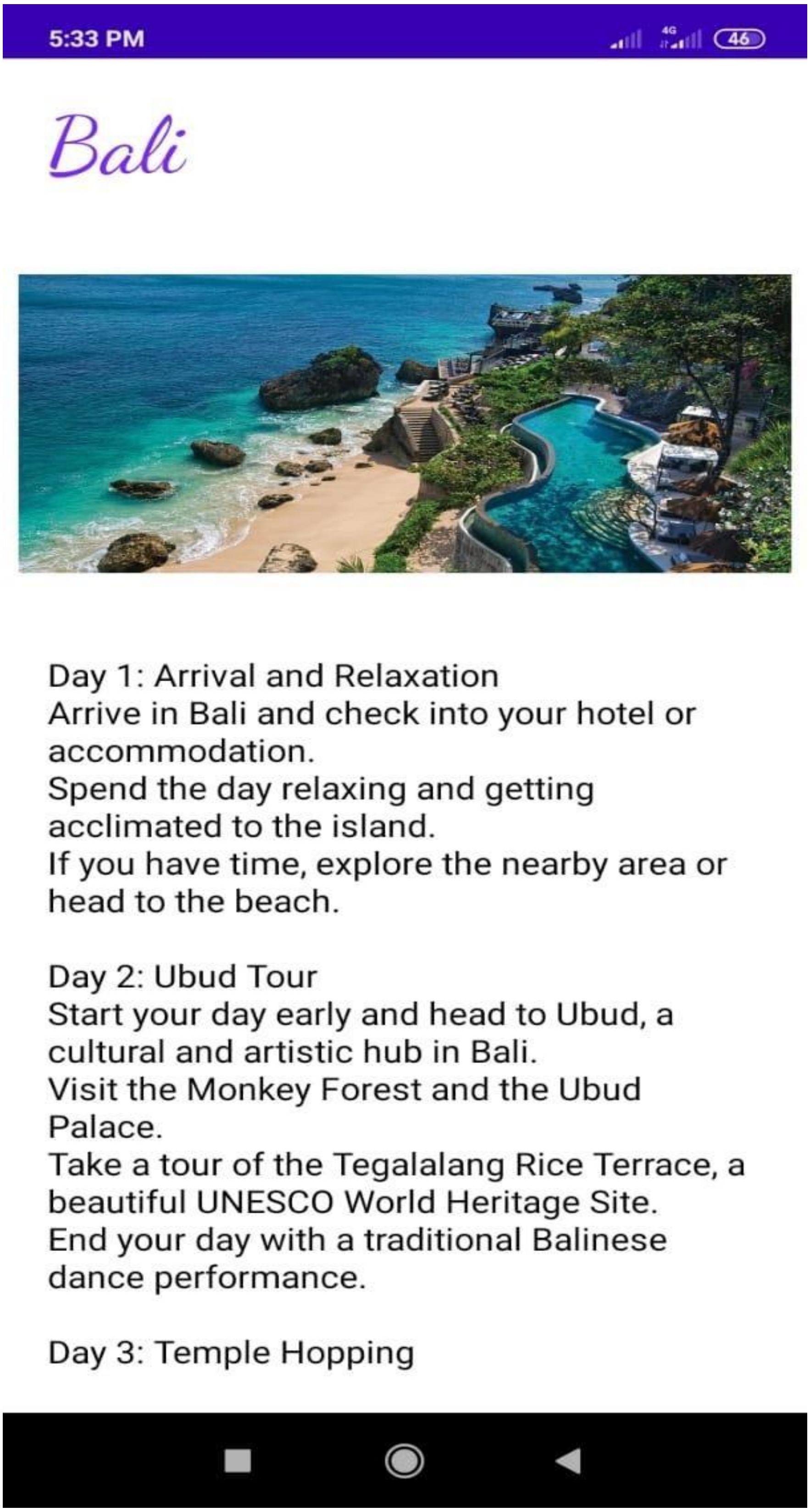
7days/2persons

Paris

Super saver pack with less than $10000

7days/2persons

LOCATION PAGE

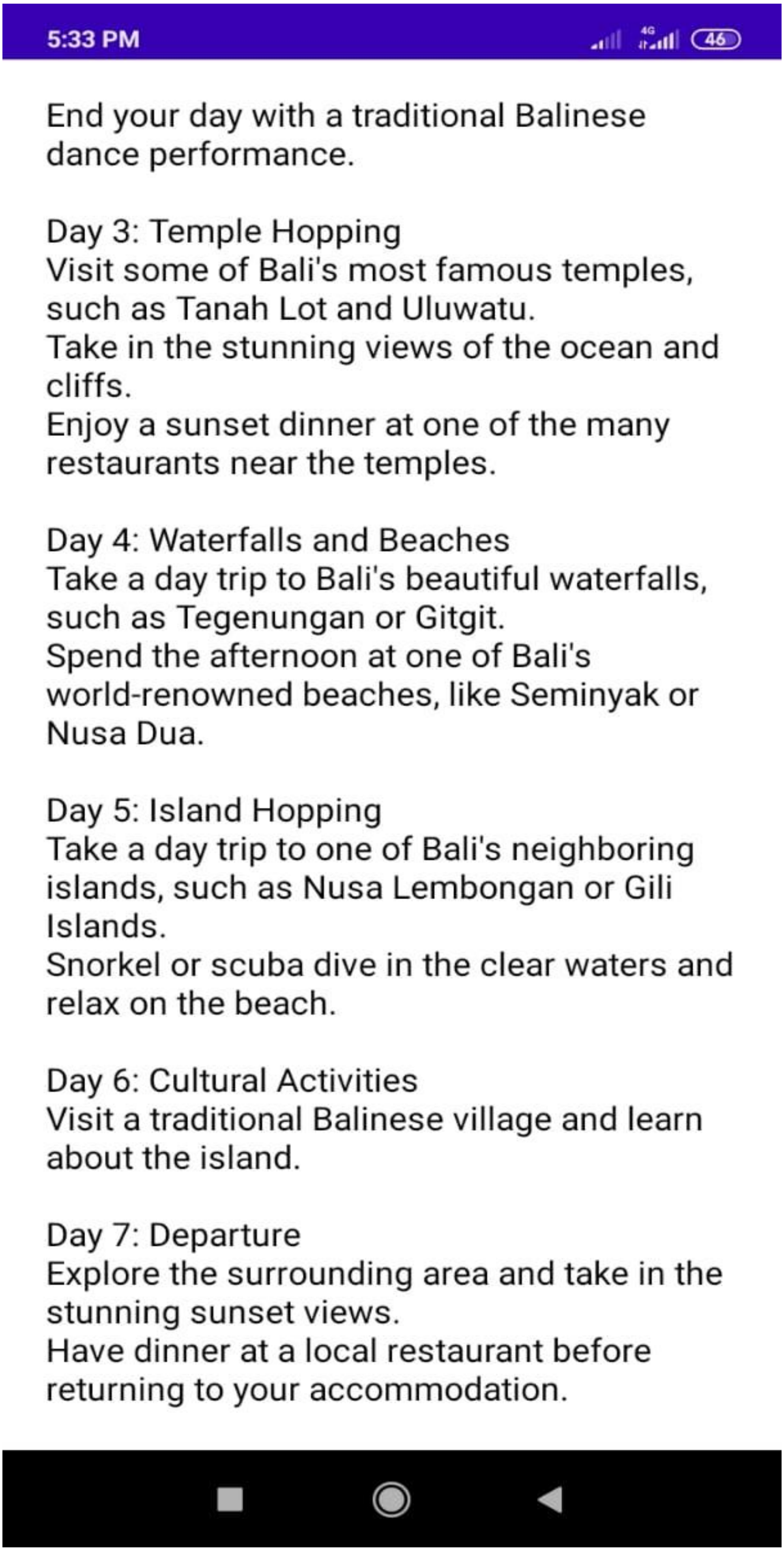


Bali

Day 1: Arrival and Relaxation
Arrive in Bali and check into your hotel or accommodation.
Spend the day relaxing and getting acclimated to the island.
If you have time, explore the nearby area or head to the beach.

Day 2: Ubud Tour
Start your day early and head to Ubud, a cultural and artistic hub in Bali.
Visit the Monkey Forest and the Ubud Palace.
Take a tour of the Tegalalang Rice Terrace, a beautiful UNESCO World Heritage Site.
End your day with a traditional Balinese dance performance.

Day 3: Temple Hopping



End your day with a traditional Balinese dance performance.

Day 3: Temple Hopping
Visit some of Bali's most famous temples, such as Tanah Lot and Uluwatu.
Take in the stunning views of the ocean and cliffs.
Enjoy a sunset dinner at one of the many restaurants near the temples.

Day 4: Waterfalls and Beaches
Take a day trip to Bali's beautiful waterfalls, such as Tegenungan or Gitgit.
Spend the afternoon at one of Bali's world-renowned beaches, like Seminyak or Nusa Dua.

Day 5: Island Hopping
Take a day trip to one of Bali's neighboring islands, such as Nusa Lembongan or Gili Islands.
Snorkel or scuba dive in the clear waters and relax on the beach.

Day 6: Cultural Activities
Visit a traditional Balinese village and learn about the island.

Day 7: Departure
Explore the surrounding area and take in the stunning sunset views.
Have dinner at a local restaurant before returning to your accommodation.

CHAPTER – IV

ADVANTAGES & DISADVANTAGES

ADVANDAGES

The usefulness of travel apps

Secondly, indeed technology has played a key role in uplifting the overall landscape of the tourism industry. Customers can thank the handy mobile application that has able to bring about the revolutionary transformation in this emerging sector.

Indeed it was not easy for planning a vacation for family members was not going to be that simple. Unless the advent of such travel and tourism apps. There was a time when the head of the family had to depend on the travel agent's plans and itinerary for making proper arrangements. These arrangements are done with travel partners to plan the trip and vacation ahead of them. But luckily this time has also passed away.

Easy Bookings Under one Roof

One of the primary reasons why vacation lovers and holiday aficionados cling on to mobile apps is because you can have access to everything under one roof and tile.

This means that if a tourist makes a prior booking on the mobile app. He can schedule and make arrangements all at a single place.

All of these includes booking and choosing the appropriate destination, booking tickets to reach the location, hotel reservation to stay for the trip, prior booking taxi service to explore the destination and site visiting, etc. The tourist does not have to open up any other separate apps to perform the other tasks. All of these things can be done through one major app that the tourist uses.

A platform that provides irresistible discounts and rebates

Snce the evolution of mobile apps and the increasing number of smartphone users and gadgets. There is a ruthless competition and the travel industry is no exception to this rule.

You must have seen that when you open the Play Store or App store you have unlimited options and countess apps to choose from. However, the users have their own specific views and thoughts about downloading a particular app from the store. But the major factor that influences their decision is the fact that how much discount or rebate

This amount justifies the reason why the user will be using your app in the first place. So if you want to increase your business ensure that you provide adequate discounts and attractive offers to the users.

- Personalized travel recommendations: Wonderlust uses advanced algorithms to analyze your travel preferences, previous trips, and interests, and provides you with personalized travel recommendations based on your unique profile.

- This means you get tailored recommendations that suit your tastes and preferences, which can save you time and make your trip planning easier.

- Comprehensive travel planning: Wonderlust allows you to plan every aspect of your trip, from flights and accommodations to activities and local events.

- It can help you create a detailed itinerary that covers everything you want to see and do, which can help you maximize your travel experience.

- Real-time travel updates: Wonderlust provides real-time updates on flight

- schedules, delays, and cancellations, as well as local weather reports and travel advisories. This can help you stay informed and adjust your plans if needed, which can save you time and money.

- Easy booking and payment: Wonderlust makes it easy to book flights, accommodations, and activities directly through the app, and it provides secure payment options for added convenience and peace of mind.

- Social networking: Wonderlust allows you to connect with other travelers,

- share your travel experiences, and get recommendations from other users.

- This can help you build a network of like-minded travelers and get insider tips on destinations, activities, and local culture.

- Budget-friendly options: Wonderlust offers a range of budget-friendly options for flights, accommodations, and activities. It can help you find the best deals and discounts for your travel needs, which can save you money and make your trip more affordable

- Sustainable travel options: Wonderlust promotes sustainable travel practices and offers eco-friendly accommodations and activities for travelers who want to reduce their environmental impact. It can help you find ecoconscious options that align with your values and travel philosophy.

DISADVANDAGES

Privacy risks

providing our personal information, such as identity, credit cards, e-mails, address, etc., we are also giving up certain rights to our privacy, allowing both the companies with which we make travel transactions and their allied companies to receive, store and use all our information.

Scames

sites offering cheap flights or hotel reservations with offers that end up being scams are very common. Therefore, it is very important to check the reputation of the site on the Internet. If we do not find much information about it, it is better to go for safer offers.

- If not be able to access your travel plans or make bookings.
- This can befrustrating and stressful, especially if you are traveling to a foreign country or unfamiliar destination.
- Limited personal touch: While Wonderlust provides personalized travel recommendations based on your profile, it may not provide the same level of personal touch and local
- knowledge thatyou would get from a human travel agent or tour guide. If you are looking for a more customized travel experience, you may prefer to work with a travel professional who can offer tailored
- With so many options and recommendations available on Wonderlust, it can be easy to feel overwhelmed and confused.
- If you are not careful, you may end up spending too much time sifting through
- information and not enough time actually enjoying your trip.

CHAPTER - V

APPLICATIONS

- Travel planning: When planning a trip, wanderlust can inspire you to explore new destinations and create unique travel experiences.
- Adventure sports: If you enjoy adventure sports, wanderlust can motivate you to try new activities and explore different terrains.
- Cultural exchange programs: Cultural exchange programs, such as study abroad programs and language programs,
- Offer opportunities to experience different cultures and satisfy your wanderlust.
- Career opportunities: Some careers, such as travel writing, photography, and tour guiding, offer opportunities to explore new places and satisfy your wanderlust while earning a living.

CHAPTER – VI

CONCLUSION

The Wonderlust travel app project is a comprehensive solution that simplifies the travel planning process and enhances the overall travel experience. The app offers a wide range of features and functionalities, including trip planning, destination discovery, budget management, sustainable travel, community building, personalized recommendations, time management, language translation, safety and security, and feedback and reviews.

The app leverages machine learning and user-generated data to provide personalized recommendations and streamline the travel planning process. It can be applied to various areas of the travel industry, catering to different types of travelers with diverse interests and preferences. Overall, The Wonderlust travel app project is an innovative and user-friendly solution that has the potential

CHAPTER – VII

FUTURE SCOPE

The Wonderlust travel app is a comprehensive solution that offers a wide range of features and functionalities to simplify and enhance the travel experience. The app leverages machine learning and user-generated data to provide personalized recommendations and streamline the travel planning process. It offers features such as trip planning, destination discovery, budget management, sustainable travel, community building, time management, language translation, safety and security, and feedback and reviews.

The app can be applied to various areas of the travel industry, catering to different types of travelers with diverse interests and preferences. Overall, the Wonderlust travel app is an innovative and user-friendly solution that has the potential to transform the way people plan and experience their travels, and offers a promising future for the travel industry.

Augmented reality: Integrating augmented reality technology can help travelers navigate and explore their destination more effectively.

For example,

users can use their phone's camera to see real-time translations, identify landmarks and historical sites, and get more detailed information about attractions.

Virtual Reality:

The integration of virtual reality technology in the app can allow travelers to experience destinations and attractions before they arrive, giving them a better idea of what to expect and making travel planning more

Efficient.Blockchain Technology:

The use of blockchain technology can enhance the security and transparency of the app's payment system, while also providing a decentralized platform for users to share their travel experiences and reviews.

Artificial Intelligence:

Further development of the app's machine learning capabilities can enable it to provide even morpersonalized recommendations, and help users make more informed decisions based on their past travel history and preferences.

Collaborations and Partnerships:

Collaborations and partnerships with airlines, hotels, and other service providers can further expand the app's offerings and provide users with more comprehensive travel solutions.

CHAPTER-VIII

APPENDIX

SOURCECODE: package
https://github.com/surenthar1332002/TRAVEL.git

BaliActivity.kt

```
package com.example.travelapp
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource

import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

class BaliActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelAppTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    PlaceOne()
                }
            }
        }
    }
}

@Composable
fun PlaceOne() {
    Column(modifier = Modifier.background(color = Color.White)
        .padding(20.dp)
        .verticalScroll(rememberScrollState())
    ) {
        Text(
            fontSize = 40.sp,

            color = Color(android.graphics.Color.rgb(120, 40, 251)),
            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_1),
        )
        Image(
            painterResource(id = R.drawable.bali), contentDescription = "",
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(
            color=Color.Black,
            text = "Day 1: Arrival and Relaxation\n" +

                    "Arrive in Bali and check into your hotel or accommodation.\n" +
                    "Spend the day relaxing and getting acclimated to the island.\n" +
                    "If you have time, explore the nearby area or head to the beach.\n" +
                    "\n" +

                    "Day 2: Ubud Tour\n" +

                    "Start your day early and head to Ubud, a cultural and artistic hub in Bali.\n"
+
                    "Visit the Monkey Forest and the Ubud Palace.\n" +
                    "Take a tour of the Tegalalang Rice Terrace, a beautiful UNESCO World
Heritage Site.\n" +
                    "End your day with a traditional Balinese dance

performance.\n" +
                    "\n" +
                    "Day 3: Temple Hopping\n" +

                    "Visit some of Bali's most famous temples, such as Tanah Lot and
Uluwatu.\n" +
                    "Take in the stunning views of the ocean and cliffs.\n" +
                    "Enjoy a sunset dinner at one of the many restaurants near the temples.\n"
+
                    "\n" +
                    "Day 4: Waterfalls and Beaches\n" +

                    "Take a day trip to Bali's beautiful waterfalls, such as Tegenungan or
Gitgit.\n" +
                    "Spend the afternoon at one of Bali's world-renowned beaches, like
Seminyak or Nusa Dua.\n" +
                    "\n" +
                    "Day 5: Island Hopping\n" +

                    "Take a day trip to one of Bali's neighboring islands, such as Nusa
Lembongan or Gili Islands.\n" +
                    "Snorkel or scuba dive in the clear waters and relax on the beach.\n" +
                    "\n" +
                    "Day 6: Cultural Activities\n" +
                    "Visit a traditional Balinese village and learn about the island.\n" +
            "\n" +
                    "Day 7: Departure\n" +
                    "Explore the surrounding area and take in the stunning sunset views.\n" +
                    "Have dinner at a local restaurant before

returning to your accommodation."
```

```
        )


    }
}
```

LoginActivity.kt

```
package com.example.travelapp

import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat

class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
            LoginScreen(this, databaseHelper)
        }
    }
}
@Composable
fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {

    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }

    Column(
        modifier = Modifier.fillMaxSize().background(Color.White),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {

        Image(painterResource(id = R.drawable.trav), contentDescription = "")

        Text(
            fontSize = 36.sp,

            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Login"
        )
        Spacer(modifier = Modifier.height(10.dp))

        TextField(
            value = username,
            onValueChange = { username = it },
            label = { Text("Username") },
            modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )

        TextField(
            value = password,
            onValueChange = { password = it },
            label = { Text("Password") },
            visualTransformation = PasswordVisualTransformation(),
            modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )

        if (error.isNotEmpty()) {
            Text(
                text = error,
                color = MaterialTheme.colors.error,
                modifier = Modifier.padding(vertical = 16.dp)
            )
        }

        Button(
            onClick = {
                if (username.isNotEmpty() && password.isNotEmpty()) {
                    val user = databaseHelper.getUserByUsername(username)
                    if (user != null && user.password == password) {
                        error = "Successfully log in"

                        context.startActivity(
                            Intent(
                                context,
                                MainActivity::class.java
                            )
                        )
                        //onLoginSuccess()
                    }
                    else {
                        error =  "Invalid username or password"
                    }

                } else {
                    error = "Please fill all fields"
                }
            },
            modifier = Modifier.padding(top = 16.dp)
        ) {
            Text(text = "Login")
```

```
        }
        Row {
            TextButton(onClick = {context.startActivity(
                Intent(
                    context,
                    RegisterActivity::class.java

                )
            )}
            )
            { Text(text = "Register") }
            TextButton(onClick = {
            })

            {
                Spacer(modifier = Modifier.width(60.dp))
                Text(text = "Forget password?")
            }
        }
    }
}
private fun startMainPage(context: Context) {
    val intent = Intent(context, MainActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

MainActivity.kt

```
package com.example.travelapp

import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Card
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale

import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelApp(this)
        }
    }


    @Composable
    fun TravelApp(context: Context) {
        Column(
            modifier = Modifier
                .padding(20.dp)
                .verticalScroll(rememberScrollState())

        ) {

            Text(
                fontSize = 40.sp,
                color = Color(android.graphics.Color.rgb(120, 40, 251)),
                fontFamily = FontFamily.Cursive,
                text = "Wanderlust Travel"
            )
            Spacer(modifier = Modifier.height(20.dp))

            // 01
            Card(
                modifier = Modifier
                    .fillMaxWidth()
                    .height(250.dp)
                    .clickable {
                    context.startActivity(
                        Intent(context, BaliActivity::class.java)

                    )
                    },
                elevation = 8.dp
            )
            {
                Column(
                    horizontalAlignment = Alignment.CenterHorizontally
                ) {
                    Image(
                        painterResource(id = R.drawable.bali),
contentDescription = "",
                        modifier = Modifier
                            .height(150.dp)
                            .scale(scaleX = 1.2F, scaleY = 1F)
                    )

                    Text(
                        text = stringResource(id = R.string.place_1),
                        fontSize = 18.sp
                    )


                    Text(
                        text = stringResource(id = R.string.description),
                        fontWeight = FontWeight.Light,
                        fontSize = 16.sp,
                        textAlign = TextAlign.Center,
                    )
```

```
        Text(
            text = stringResource(id = R.string.plan), color = Color.Gray,
            fontSize = 16.sp
        )
    }
}

Spacer(modifier = Modifier.height(20.dp))


//02


Card(
    modifier = Modifier
        .fillMaxWidth()
        .height(250.dp)
        .clickable {
        context.startActivity(
            Intent(context, ParisActivity::class.java)

        )
        },
    elevation = 8.dp
)

    Column(
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Image(
            painterResource(id = R.drawable.paris), contentDescription = "",
            modifier = Modifier
                .height(150.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )

        Text(
            text = stringResource(id = R.string.place_2),
            fontSize = 18.sp
        )
        Text(
            text = stringResource(id = R.string.description),

            fontWeight = FontWeight.Light,
            fontSize = 16.sp,
            textAlign = TextAlign.Center,
        )

        Text(
            text = stringResource(id = R.string.plan), color = Color.Gray,
            fontSize = 16.sp
        )
    }
}
Spacer(modifier = Modifier.height(20.dp))

//03
Card(
    modifier = Modifier
        .fillMaxWidth()
        .height(250.dp)
        .clickable {
        context.startActivity(
            Intent(context, SingaporeActivity::class.java)

        )
        },
    elevation = 8.dp
)
{
    Column(
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Image(

            painterResource(id = R.drawable.singapore), contentDescription = "",
            modifier = Modifier
                .height(150.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
                )

                Text(
                    text = stringResource(id = R.string.place_3),
                    fontSize = 18.sp
                )
                Text(
                    text = stringResource(id = R.string.description),
                    fontWeight = FontWeight.Light,

                    fontSize = 16.sp,
                    textAlign = TextAlign.Center,
                )

                Text(
                    text = stringResource(id = R.string.plan), color = Color.Gray,
                    fontSize = 16.sp
                )
            }
        }

        Spacer(modifier = Modifier.height(20.dp))
    }
  }
}
```

ParisActivity.kt

```
package com.example.travelapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

class ParisActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {

            TravelAppTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting()
                }
            }
        }
    }
}

@Composable
fun Greeting() {

    Column(
        modifier = Modifier.background(color = Color.White)
            .padding(20.dp)
            .verticalScroll(rememberScrollState())
    ) {
        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120, 40, 251)),
            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_2),
        )
        Image(
            painterResource(id = R.drawable.paris), contentDescription = "",
            modifier = Modifier

                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(
            color=Color.Black,
            text = "Day 1: Arrival and Introduction\n" +

                    "Check into your accommodation and freshen up\n" +
                    "Take a stroll around the neighborhood to get acquainted\n" +
                    "Visit the Eiffel Tower, preferably in the evening when it is lit up\n" +
                    "Have a relaxing dinner at a nearby restaurant\n" +
                    "\n" +
                    "Day 2: Art and History\n" +

                    "Visit the Louvre Museum to see some of the world's most famous art
pieces\n" +
                    "Stroll through the Tuileries Garden and the Place de la Concorde\n" +
                    "Visit the Orsay Museum, which houses a large collection of impressionist
art\n" +
                    "Have dinner at a local French restaurant\n" +

                    "\n" +
                    "Day 3: French Culture and Food\n" +

                    "Visit the Montmartre neighborhood to see the famous Basilique du Sacré-
Cœur and Place du Tertre\n" +
                    "Explore the historic neighborhood of Le Marais\n" +
                    "Try some delicious French pastries at a local

bakery\n" +
                    "Have dinner at a brasserie to taste some classic French cuisine\n" +

                    "\n" +

                    "Day 4: Architecture and Gardens\n" +

                    "Visit the Palace of Versailles, a UNESCO World Heritage site, and explore its beautiful gardens\n" +
                    "Walk along the Champs-Elysées and stop at the Arc de Triomphe\n" +
                    "Visit the Sainte-Chapelle, a beautiful Gothic chapel with stunning stained-glass windows\n" +
                    "Have dinner at a local restaurant in the 7th arrondissement\n" +

                    "\n" +
                    "Day 5: Shopping and Sightseeing\n" +

                    "Visit the Notre-Dame Cathedral and climb up to the top for a stunning view of the city\n" +
                    "Explore the Latin Quarter and visit the Panthéon\n" +
                    "Go shopping at the famous Galeries Lafayette or Printemps department stores\n" +
                    "Have dinner at a local bistro\n" +
                    "\n" +
                    "Day 6: Parisian Parks and Museums\n" +

                    "Visit the Musée Rodin and explore its beautiful gardens\n" +
                    "Stroll through the Luxembourg Gardens and visit the Luxembourg Palace\n" +

                    "Visit the Centre Pompidou, a modern art museum in the Marais neighborhood\n" +
                    "Have dinner at a local restaurant in the Latin Quarter\n" +
                    "\n" +
                    "Day 7: River Cruise and Farewell\n" +

                    "Take a boat cruise along the Seine River to see the city from a different perspective\n" +
                    "Visit the Musée de l'Orangerie, which houses Monet's famous water lilies paintings\n" +
                    "Have a farewell dinner at a Michelin-starred restaurant"
        )
    }
```

```
}
```

RegisterActivity.kt

```
package com.example.travelapp

import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*

import androidx.compose.material.*
import androidx.compose.runtime.*

import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat

class RegisterActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
            RegistrationScreen(this, databaseHelper)
        }
    }
}

@Composable
fun RegistrationScreen(context: Context, databaseHelper: UserDatabaseHelper) {

    var username by remember { mutableStateOf("") }

    var password by remember { mutableStateOf("") }

    var email by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }

    Column(
        modifier = Modifier.fillMaxSize().background(Color.White),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {

        Image(painterResource(id = R.drawable.tra), contentDescription = "")

        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Register"
        )
        Spacer(modifier = Modifier.height(10.dp))

        TextField(
            value = username,
            onValueChange = { username = it },
            label = { Text("Username") },
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)

        )

        TextField(
            value = email,
            onValueChange = { email = it },
            label = { Text("Email") },
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )

        TextField(
            value = password,
            onValueChange = { password = it },
            label = { Text("Password") },
            visualTransformation = PasswordVisualTransformation(),
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )


        if (error.isNotEmpty()) {
            Text(
              text = error,
                color = MaterialTheme.colors.error,
                modifier = Modifier.padding(vertical = 16.dp)
            )
        }

        Button(
            onClick = {
                if (username.isNotEmpty() && password.isNotEmpty() && email.isNotEmpty())
{
                    val user = User(

                        id = null,
                        firstName = username,
                        lastName = null,
                        email = email,
                        password = password
                    )
                    databaseHelper.insertUser(user)
                    error = "User registered successfully"
                    // Start LoginActivity using the current context
                    context.startActivity(
                        Intent(
                            context,
                            LoginActivity::class.java
                        )
                    )
                } else {
                    error = "Please fill all fields"
                }
            },
            modifier = Modifier.padding(top = 16.dp)
        ) {

            Text(text = "Register")
        }
        Spacer(modifier = Modifier.width(10.dp))
        Spacer(modifier = Modifier.height(10.dp))

        Row() {
            Text(
                modifier = Modifier.padding(top = 14.dp), text = "Have an account?"
            )

            TextButton(onClick = {
                context.startActivity(
                    Intent(
                        context,
                        LoginActivity::class.java
                    )
                )
            })

            {
                Spacer(modifier = Modifier.width(10.dp))
                Text(text = "Log in")
            }
        }
```

```
    }
}
private fun startLoginActivity(context: Context) {
    val intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

SingaporeActivity.kt

```
package com.example.travelapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*

import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

class SingaporeActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelAppTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting2()
                }
            }

        }
    }
}

@Composable
fun Greeting2() {

    Column(
        modifier = Modifier.background(color = Color.White)
            .padding(20.dp)
            .verticalScroll(rememberScrollState())
    ) {
        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120, 40, 251)),
            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_3),
        )
        Image(
            painterResource(id = R.drawable.singapore), contentDescription = "",
            modifier = Modifier

                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(
            color = Color.Black,
            text = "Day 1:\n" +
                    "Morning: Visit Gardens by the Bay and marvel at the Supertree Grove and the Flower Dome and Cloud Forest conservatories.\n" +
                    "Afternoon: Explore the Marina Bay Sands complex, which includes a casino, luxury shopping mall, and observation deck with a stunning view of the city.\n" +
                    "\n" +
                    "Day 2:\n" +

                    "Morning: Explore the historic district of Chinatown, including the Buddha Tooth Relic Temple and Museum and the Sri Mariamman Temple.\n" +
                    "Afternoon: Visit the nearby Clarke Quay for lunch and to explore its waterfront restaurants, bars, and shops.\n" +
                    "\n" +
                    "Day 3:\n" +

                    "Morning: Take a tour of the UNESCO-listed Botanic Gardens, one of the world's most famous and significant tropical gardens.\n" +
                    "Afternoon: Head over to the National Museum of Singapore, which houses a vast collection of historical and cultural artifacts.\n" +

                    "\n" +

                    "Day 4:\n" +

                    "Morning: Visit the Singapore Zoo and admire the wildlife, including orangutans, tigers, and elephants.\n" +
                    "Afternoon: Head over to Sentosa Island and relax at one of its many beaches or try some of the many attractions such as Universal Studios Singapore or Adventure Cove Waterpark.\n" +
                    "\n" +
                    "Day 5:\n" +

                    "Morning: Go on a nature walk at MacRitchie Reservoir, which offers hiking trails and stunning views of the city skyline.\n" +
                    "Afternoon: Visit Little India, a vibrant and colorful neighborhood, and explore the shops, temples, and food stalls.\n" +
                    "\n" +
                    "Day 6:\n" +

                    "Morning: Explore the trendy neighborhood of Tiong Bahru, known for its hip cafes and boutiques, as well as its Art Deco architecture.\n" +
                    "Afternoon: Visit the National Gallery Singapore, which houses the largest public collection of modern art in Singapore and Southeast Asia.\n" +
                    "\n" +
                    "Day 7:\n" +

                    "Morning: Take a day trip to the nearby island of Pulau Ubin, where you can
    rent a "
            )
        }
    }
```

User.kt

```
package com.example.travelapp

import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "user_table")
data class User(
    @PrimaryKey(autoGenerate = true) val id: Int?,
    @ColumnInfo(name = "first_name") val firstName: String?,
    @ColumnInfo(name = "last_name") val lastName: String?,
    @ColumnInfo(name = "email") val email: String?,
    @ColumnInfo(name = "password") val password: String?,

    )
```

UserDao.kt

```
package com.example.travelapp

import androidx.room.*

@Dao
interface UserDao {

    @Query("SELECT * FROM user_table WHERE email = :email")
    suspend fun getUserByEmail(email: String): User?
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insertUser(user: User)

    @Update
    suspend fun updateUser(user: User)

    @Delete
    suspend fun deleteUser(user: User)
}
```

UserDatabase.kt

```
package com.example.travelapp

import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase

@Database(entities = [User::class], version = 1)
abstract class UserDatabase : RoomDatabase() {

    abstract fun userDao(): UserDao

    companion object {

        @Volatile
        private var instance: UserDatabase? = null

        fun getDatabase(context: Context): UserDatabase {
            return instance ?: synchronized(this) {
                val newInstance = Room.databaseBuilder(
                    context.applicationContext,

                    UserDatabase::class.java,
                    "user_database"
                ).build()
                instance = newInstance
                newInstance
            }
        }

    }
}
```

UserDatabaseHelper.kt

```
package com.example.travelapp

import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class UserDatabaseHelper(context: Context) :
    SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_VERSION = 1
        private const val DATABASE_NAME = "UserDatabase.db"

        private const val TABLE_NAME = "user_table"
        private const val COLUMN_ID = "id"
        private const val COLUMN_FIRST_NAME = "first_name"
        private const val COLUMN_LAST_NAME = "last_name"
        private const val COLUMN_EMAIL = "email"
        private const val COLUMN_PASSWORD = "password"
    }

    override fun onCreate(db: SQLiteDatabase?) {

        val createTable = "CREATE TABLE $TABLE_NAME (" +
                "$COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
                "$COLUMN_FIRST_NAME TEXT, " +
                "$COLUMN_LAST_NAME TEXT, " +
                "$COLUMN_EMAIL TEXT, " +
                "$COLUMN_PASSWORD TEXT" +
                ")"

        db?.execSQL(createTable)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        db?.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
        onCreate(db)
    }

    fun insertUser(user: User) {
        val db = writableDatabase
        val values = ContentValues()
        values.put(COLUMN_FIRST_NAME, user.firstName)
        values.put(COLUMN_LAST_NAME, user.lastName)
        values.put(COLUMN_EMAIL, user.email)
        values.put(COLUMN_PASSWORD, user.password)
        db.insert(TABLE_NAME, null, values)

        db.close()
    }

    @SuppressLint("Range")
    fun getUserByUsername(username: String): User? {

        val db = readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME WHERE
$COLUMN_FIRST_NAME = ?", arrayOf(username))
        var user: User? = null

        if (cursor.moveToFirst()) {
            user = User(
                id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                firstName =
cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
                lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
                email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
                password =
cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
            )
        }
        cursor.close()
        db.close()
        return user
    }
    @SuppressLint("Range")
    fun getUserById(id: Int): User? {
        val db = readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME WHERE
$COLUMN_ID = ?", arrayOf(id.toString()))
        var user: User? = null
        if (cursor.moveToFirst()) {

            user = User(
                id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                firstName =
cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
                lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
                email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
                password =
cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
            )
        }
        cursor.close()
        db.close()
        return user
    }

    @SuppressLint("Range")
    fun getAllUsers(): List<User> {

        val users = mutableListOf<User>()
        val db = readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME", null)
        if (cursor.moveToFirst()) {
            do {

                val user = User(
                    id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                    firstName =
cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
                    lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
                    email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
                    password =
cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
                )
                users.add(user)
            } while (cursor.moveToNext())
        }
        cursor.close()
        db.close()
        return users
    }

}
```