1. INTRODUCTION

1.1 OVERVIEW

Software has changed how people travel, where they travel, when they travel, and even how they share their travels with others. It has also allowed for the democratization of travel, enabling travellers to easily book flights, hotels, and adventure tours. With the addition of new technology, travel has become more comfortable, providing travellers with the opportunity to access the Internet in-flight, book travel accommodations in advance, and see reviews of restaurants and tourist attractions.

It is also changing how companies in the travel industry operate, enabling real-time monitoring of inventory, pricing, and demand. It enables travel companies to offer greater value to their customers by collecting data about their customers, improving their customer service, and providing opportunities for their customers to share their experiences. As the industry grows, many businesses are introducing new technology and software solutions to solve real-world problems.

A project built using the Android Compose UI toolkit. It demonstrates how to create a simple travel app using the Compose libraries. It also features a personalized feed of recommended accommodations based on the locations.

1.2 PURPOSE

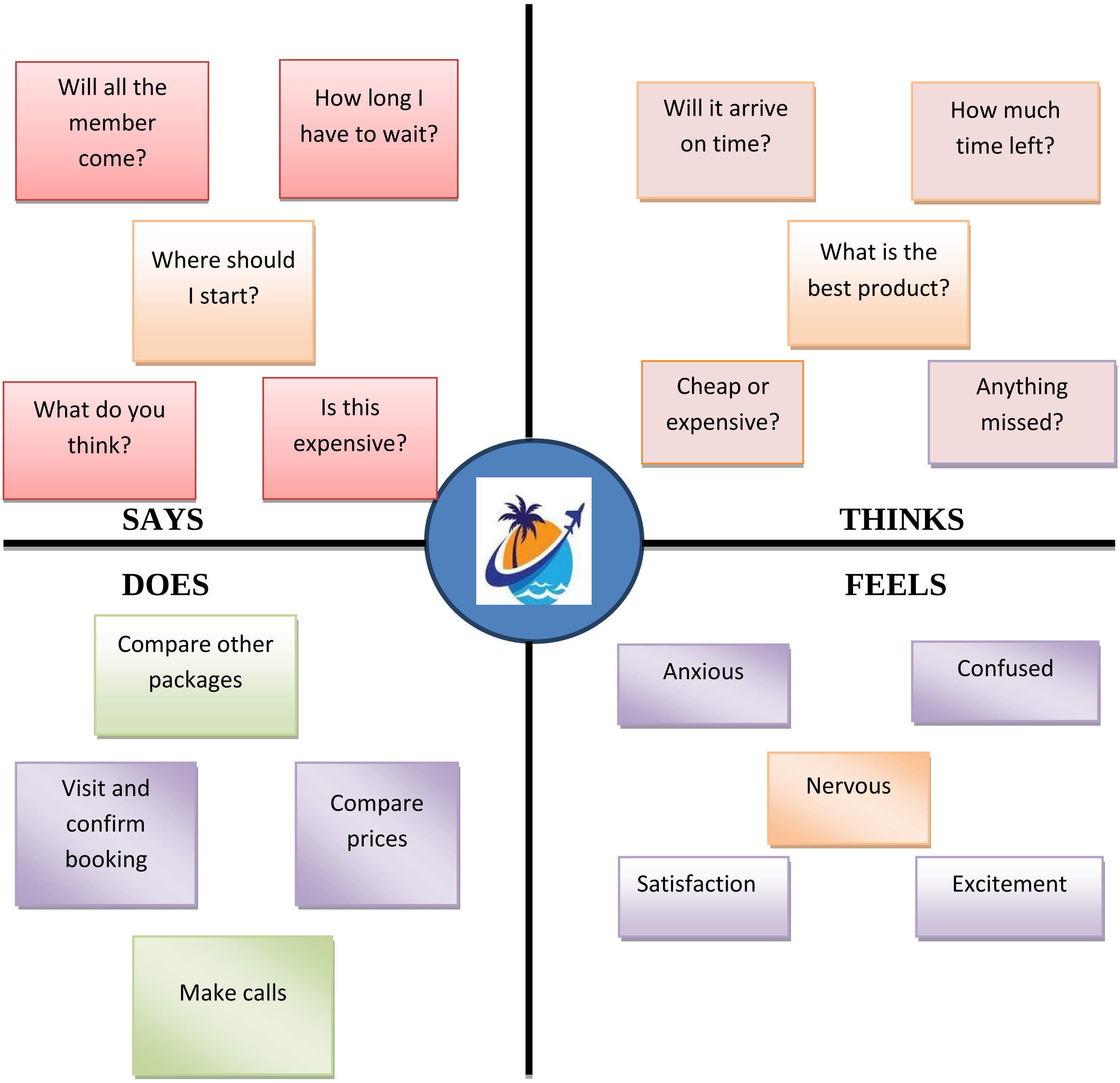
Travel planners help clients make travel arrangements. They typically work in an office environment, making reservations, booking flights, and arranging transport. Travel planners may also visit holiday resorts and destinations to help clients make informed decisions.

A mobile travel app is a software application created specifically for use on smaller devices such as smart phones and tablets that help business travelers to plan and manage their travel while on the go. Booking travel, altering bookings, and reviewing itineraries are key features available in most mobile travel applications.

A travel app helps them to organize their trip smoothly, swiftly, and cost-effectively. From finding a hotel to booking airline tickets, travel apps now include all sophisticated features to provide customers with fast and accurate services at their finger's touch.With specialized Travel Apps, you can provide your customers with a real-time traveling experience. Nowadays, Mobile Solutions for Travel Companies offer numerous benefits that improve and ease travel arrangements.

2 .PROBLEM DEFINITION ANA DESIGN THINKING

2.1 Empathy Map



SAYS

Will all the member come?

How long I have to wait?

Where should I start?

What do you think?

Is this expensive?

THINKS

Will it arrive on time?

How much time left?

What is the best product?

Cheap or expensive?

Anything missed?

DOES

Compare other packages

Visit and confirm booking

Compare prices

Make calls

FEELS

Anxious

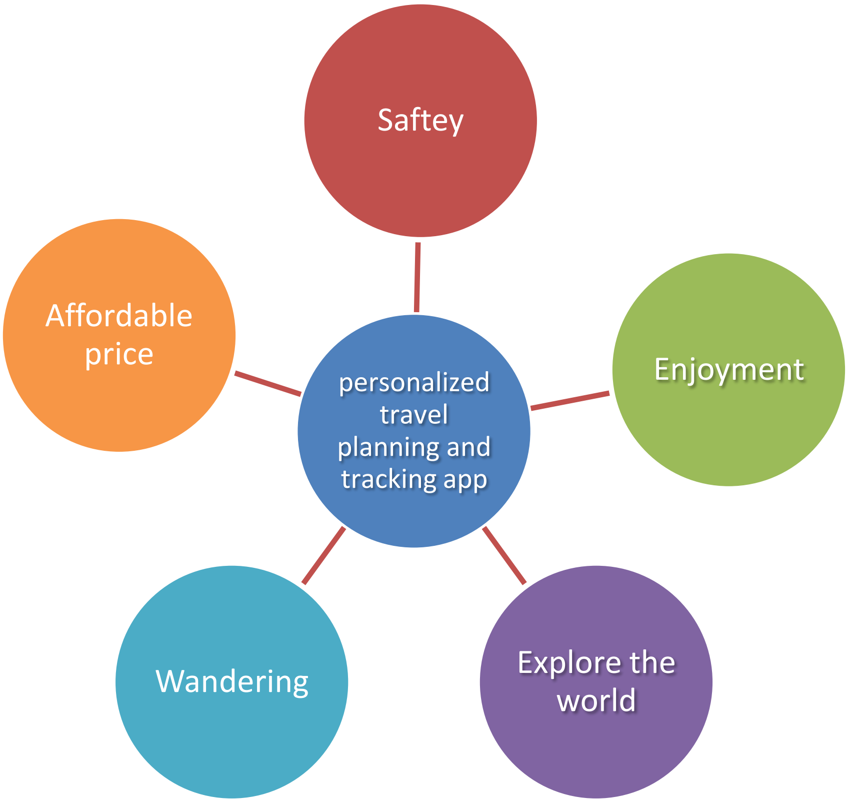
Confused

Nervous

Satisfaction

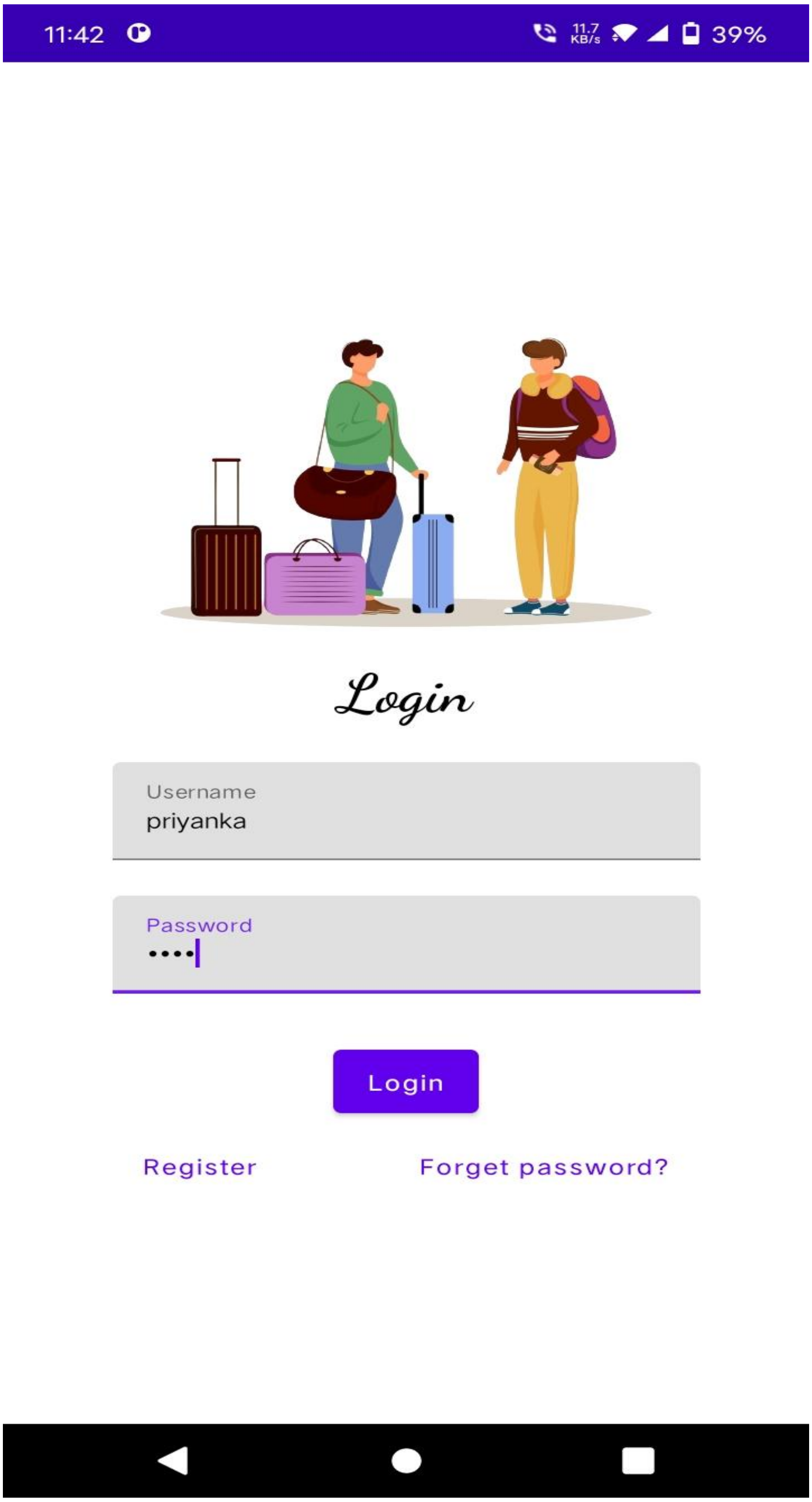
Excitement

2.2 Ideation and brainstorming map



3. RESULT

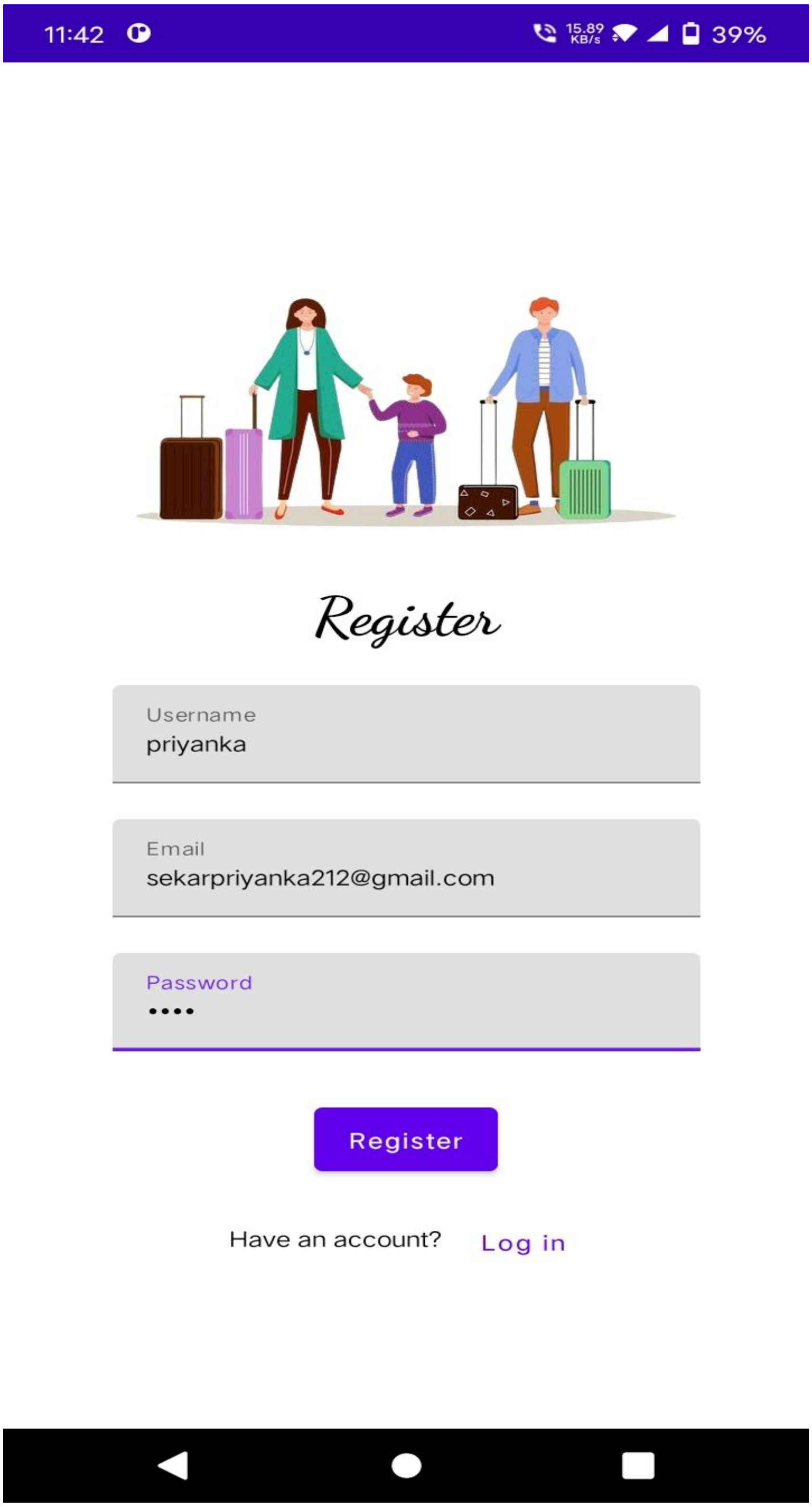
Login Page



Username
priyanka

Password
••••

Register Page



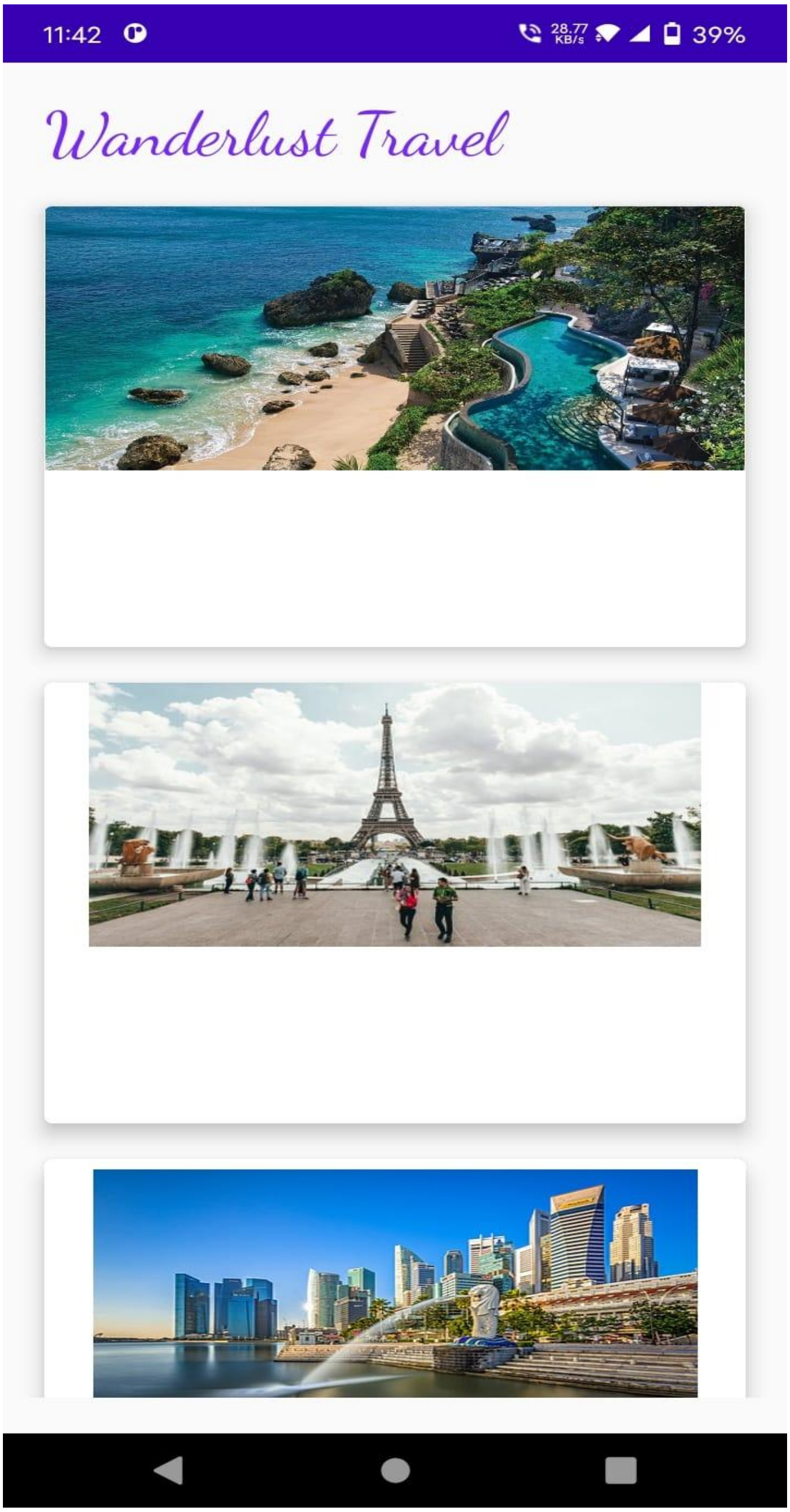
Register

Username
priyanka

Email
sekarpriyanka212@gmail.com

Password
••••

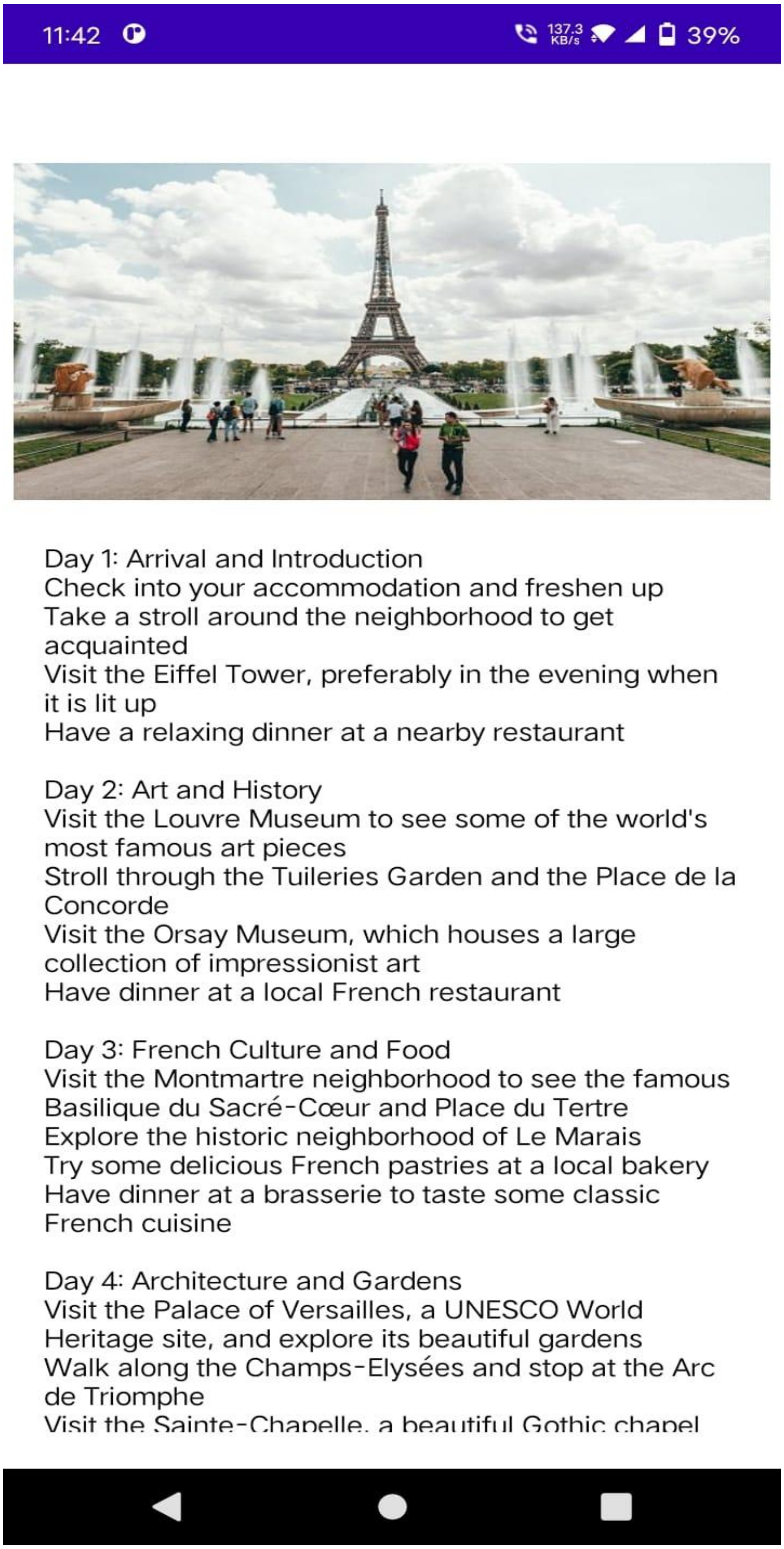
View Page



11:42

28.77 KB/s 39%

Wanderlust Travel

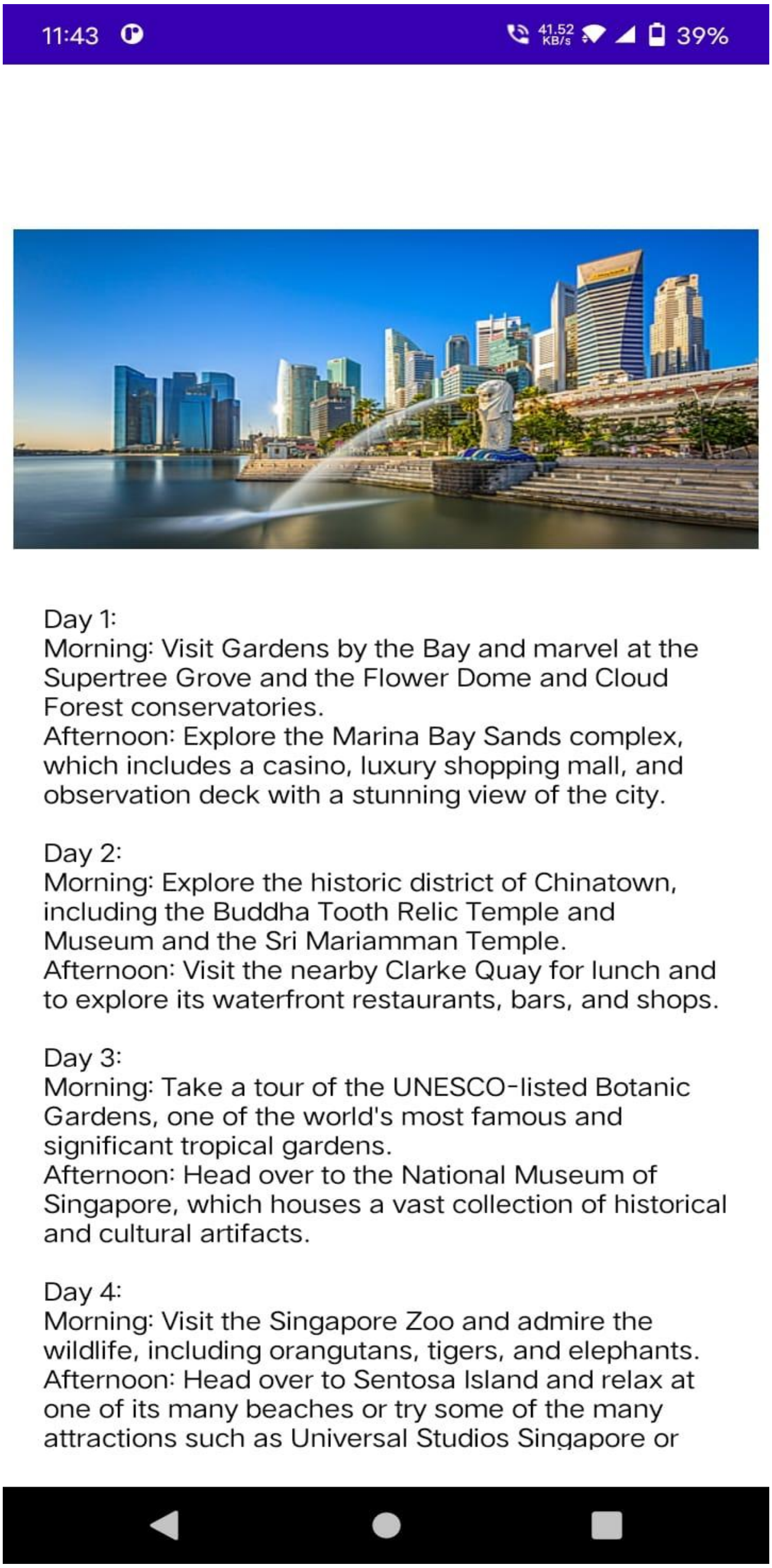


Day 1: Arrival and Introduction
Check into your accommodation and freshen up
Take a stroll around the neighborhood to get acquainted
Visit the Eiffel Tower, preferably in the evening when it is lit up
Have a relaxing dinner at a nearby restaurant

Day 2: Art and History
Visit the Louvre Museum to see some of the world's most famous art pieces
Stroll through the Tuileries Garden and the Place de la Concorde
Visit the Orsay Museum, which houses a large collection of impressionist art
Have dinner at a local French restaurant

Day 3: French Culture and Food
Visit the Montmartre neighborhood to see the famous Basilique du Sacré-Cœur and Place du Tertre
Explore the historic neighborhood of Le Marais
Try some delicious French pastries at a local bakery
Have dinner at a brasserie to taste some classic French cuisine

Day 4: Architecture and Gardens
Visit the Palace of Versailles, a UNESCO World Heritage site, and explore its beautiful gardens
Walk along the Champs-Elysées and stop at the Arc de Triomphe
Visit the Sainte-Chapelle, a beautiful Gothic chapel

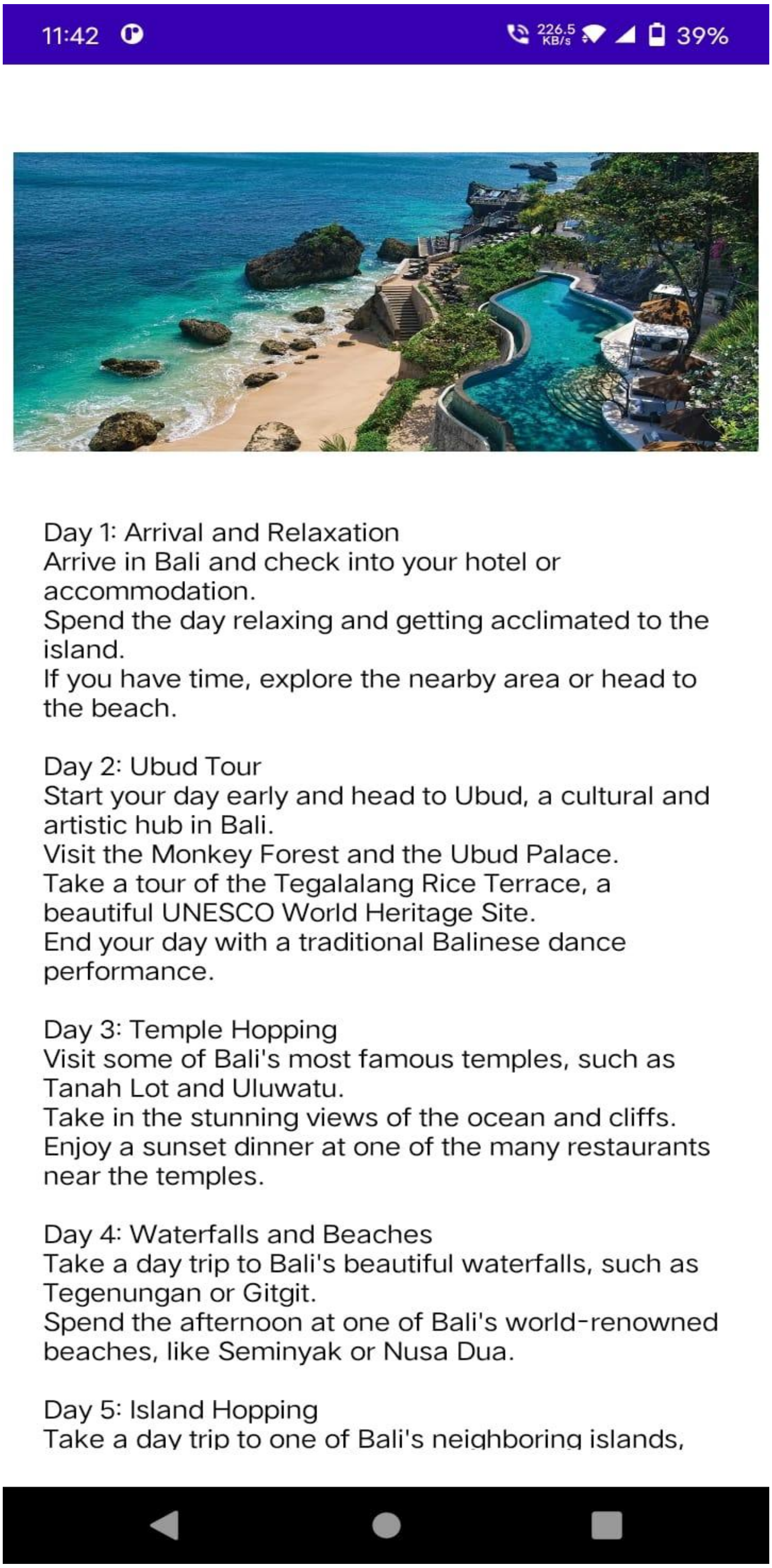


Day 1:
Morning: Visit Gardens by the Bay and marvel at the Supertree Grove and the Flower Dome and Cloud Forest conservatories.
Afternoon: Explore the Marina Bay Sands complex, which includes a casino, luxury shopping mall, and observation deck with a stunning view of the city.

Day 2:
Morning: Explore the historic district of Chinatown, including the Buddha Tooth Relic Temple and Museum and the Sri Mariamman Temple.
Afternoon: Visit the nearby Clarke Quay for lunch and to explore its waterfront restaurants, bars, and shops.

Day 3:
Morning: Take a tour of the UNESCO-listed Botanic Gardens, one of the world's most famous and significant tropical gardens.
Afternoon: Head over to the National Museum of Singapore, which houses a vast collection of historical and cultural artifacts.

Day 4:
Morning: Visit the Singapore Zoo and admire the wildlife, including orangutans, tigers, and elephants.
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



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
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,

4. ADVANTAGES & DISADVANTAGE

Advantages:

- A typical tour package will usually include plenty of destinations. Chances are that going to all these places on your own would cost a lot more, mainly because travel agents and tourism companies have good connections with the hotels and transportation companies, and also because they get maximum discounts for sending groups rather than individuals or single families.
- One of the biggest advantages is of course the convenience. As earlier suggested, planning for a full-fledged trip and visiting various places will take quite a lot of planning and due diligence. But when you opt for a tour package, you can just sit back and relax — at least as far as planning is concerned.You will be able to relax and enjoy your travel and the places without having to worry about the tickets, and hotel reservations in your next stop.
- Tour packages can be a great option when you are looking to plan a honeymoon or travel with your family. A poorly planned trip can be quite embarrassing as well as stressful; therefore, going with a reputable travel company might be a safer option.
- Safety is another big advantage of tour packages. When you are doing the planning, and choosing the destinations or activities to do, you might pick out a place or activity that might have some potential risks that you'drather avoid. Travel companiesknow places inside out, and they will usually choose risk-free destinations and activities, keeping in mind the age and preferences of the participants.

Disadvantage:

- The biggest disadvantage of a tour package is the lack of control. You need to follow the guidelines and a strict schedule. You cannot extend your stay in a certain place even if you love it. Similarly you cannot move to the next place if there's nothing to be excited about, before the scheduled time.
- You need to be extra careful while choosing one of these packages. The aforementioned advantages can easily turn into disadvantages if you don't choose the right company.
- You will have to be a little flexible with your dates and timings, in case you are looking to leave and arrive back on certain dates. You might find it hard to find an appropriate trip with the same time frame.

5. APPLICATIONS

- One of the main reasons why people use travel apps is to plan their trips. Apps which offer such functions allow picking different destinations, choosing transportation to get to those destinations, booking hotels, etc.
- Plan, track, and relive your travels. Record your route automatically, keeping your phone in your pocket and eyes on the world.
- This all-in-one travel app shows you the world’s most enticing travel destinations, gives you insider tips and plots your route, locations and photos when the journey is underway

6. CONCLUSION

The travel tracking app tour is a complete package itself to introduce India in the most beautiful way. It gives you essence of all kind of diversities in art, culture, lifestyle, architecture, history, rich heritage and others. The travel tracking app is one of the most preferred tour package among the visitors that aware the visitors with the different aspects of the vivid places of India. It can become an unforgettable journey of your lifetime as it will leave you with amazing memories. Know more about the travel tracking app in our upcoming blogs and plan your vacations now.

Our system presents a personalized travel itinerary recommendation system by implementing topical package model using data mining form social media: travelogues and community-contributed photos. The advantages of our work are that firstly, the system automatically mines user 's topical preferences including the point of interest, cost and time and secondly the recommendation is not only providing point of interest but also travel sequence order, considering both the popularity and user's travel preferences at the same time. The system also provides user with flexibility to freeze a day or two for his/her personal work (e.g., meeting, conference, etc.) and successfully managed to show the travel itinerary and hotel bookings for comfortable stay in a single framework. Out project mines and ranks famous routes based on the similarity between user and route package and then optimize the top ranked famous routes according to social similar users' travel records thereby providing user with the most efficient and feasible.

7. FUTURE SCOPE

- The app in future can cover more than one state, it can cover whole of NORTH, SOUTH, EAST or WEST of India and even the whole of India and can also cover whole around the world.
- Even microphone can also be inserted so as to provide ease to user in providing the information regarding the place he/she want to discover, the budget he/she is planning and the kind of tour they want.
- Apart from just planning the tour, more descriptive information about different places can also be provided to enhance the knowledge of user.
- In future breath-taking images relating to different places can also be added to attract the users and make their choice a easy one. In future more secure framework can be built using upcoming advanced technologies and several others up gradations required

8. APPENDIX

user Data Class

```
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 interface

```
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 class

```
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 class

```
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
   }
}
```

LoginActivity.kt with database

```
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)
}
```

RegisterActivity.kt with database

```
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)
}
```

MainActivity.Kt File

```
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))
        }
    }
}
```

BaliActivity.kt file

```
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."
    )
  }
}
```

ParisActivity.kt file

```
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
            "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 b
                "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
                "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 t
                "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 perspec
                "Visit the Musée de l'Orangerie, which houses Monet's famous water lilies painti
                "Have a farewell dinner at a Michelin-starred restaurant"
            )
          }
        }
    }
```

Modifying Android Manifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:allowBackup="true"
```

```
android:dataExtractionRules="@xml/data_extraction_rules"
android:fullBackupContent="@xml/backup_rules"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportsRtl="true"
android:theme="@style/Theme.TravelApp"
tools:targetApi="31">
<activity
    android:name=".RegisterActivity"
    android:exported="false"
    android:label="RegisterActivity"
    android:theme="@style/Theme.TravelApp" />
<activity
    android:name=".SingaporeActivity"
    android:exported="false"
    android:label="@string/title_activity_singapore"
    android:theme="@style/Theme.TravelApp" />
<activity
    android:name=".ParisActivity"
    android:exported="false"
    android:label="@string/title_activity_paris"
    android:theme="@style/Theme.TravelApp" />
```

```
        <activity
            android:name=".BaliActivity"
            android:exported="false"
            android:label="@string/title_activity_bali"
            android:theme="@style/Theme.TravelApp" />
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.TravelApp"/>
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.TravelApp">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```