**CONCEPT**

1. **INTRODUCTION:**
   1. Overview
   2. Purpose
2. **PROBLEM DEFINITION & DESIGN THINKING:**
   1. Empathy Map
   2. Ideation & Brainstorming Map
3. **RESULT**
4. **ADVANTAGES & DISADVANTAGES**
5. **APPLICATIONS**
6. **CONCLUSION**
7. **FUTURE SCOPE**
8. **APPENDIX**

Source Code

**INTRODUCTION**

The main objective of Money Matters App control of your finances and simplify your budgeting process, download the expenses tracker app for Android today and start tracking your expenses with ease.

**1.1 Overview**:

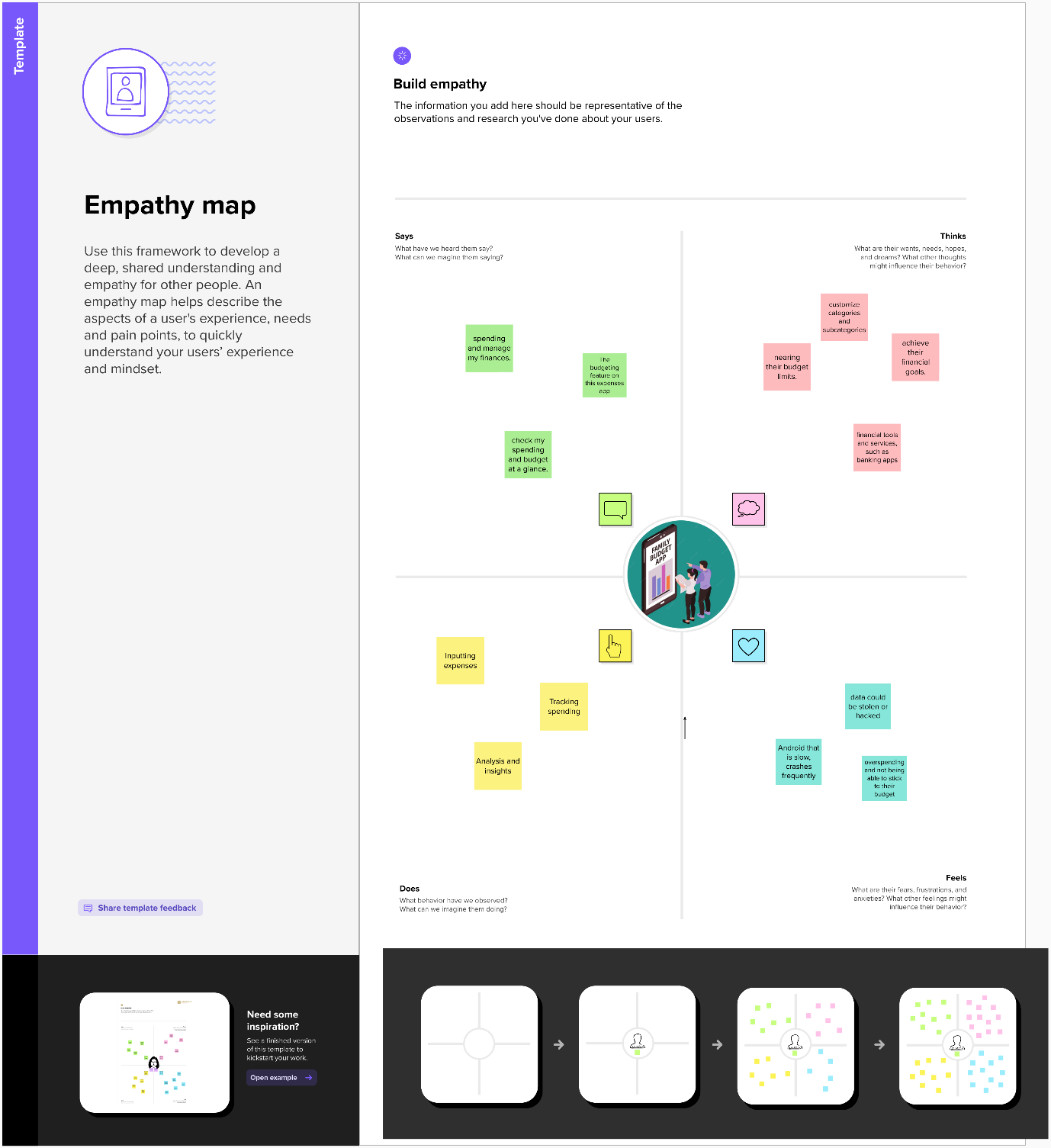
* Expense Tracking: The app should allow users to record their expenses, categorize them, and add notes or comments as needed.
* Budget Management: The app should also provide users with alerts when they are approaching or exceeding their budget limits.

**1.2 Purpose:**

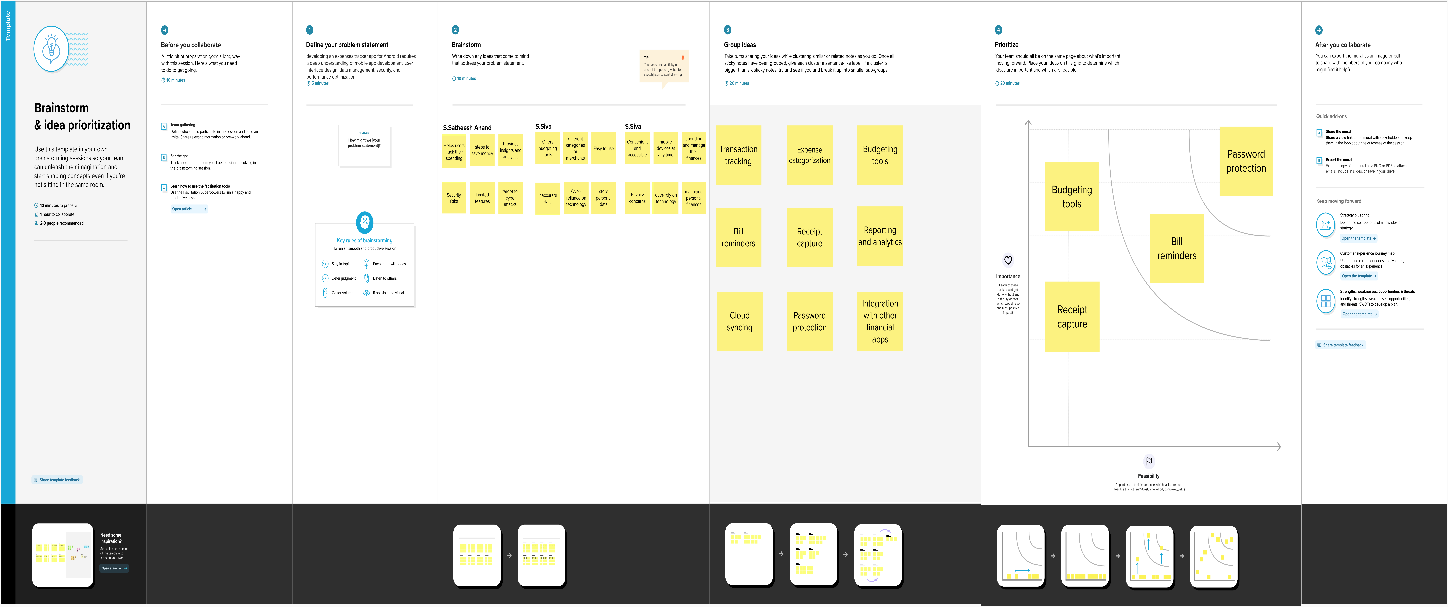
* Expense Tracking: An expenses tracker app enables users to track their expenses in real-time, as they occur.
* Financial Planning: An expenses tracker app provides users with a clear picture of their financial situation, allowing them to plan for the future.

**Problem Definition & Design Thinking**

**2.1 Empathy Map:**

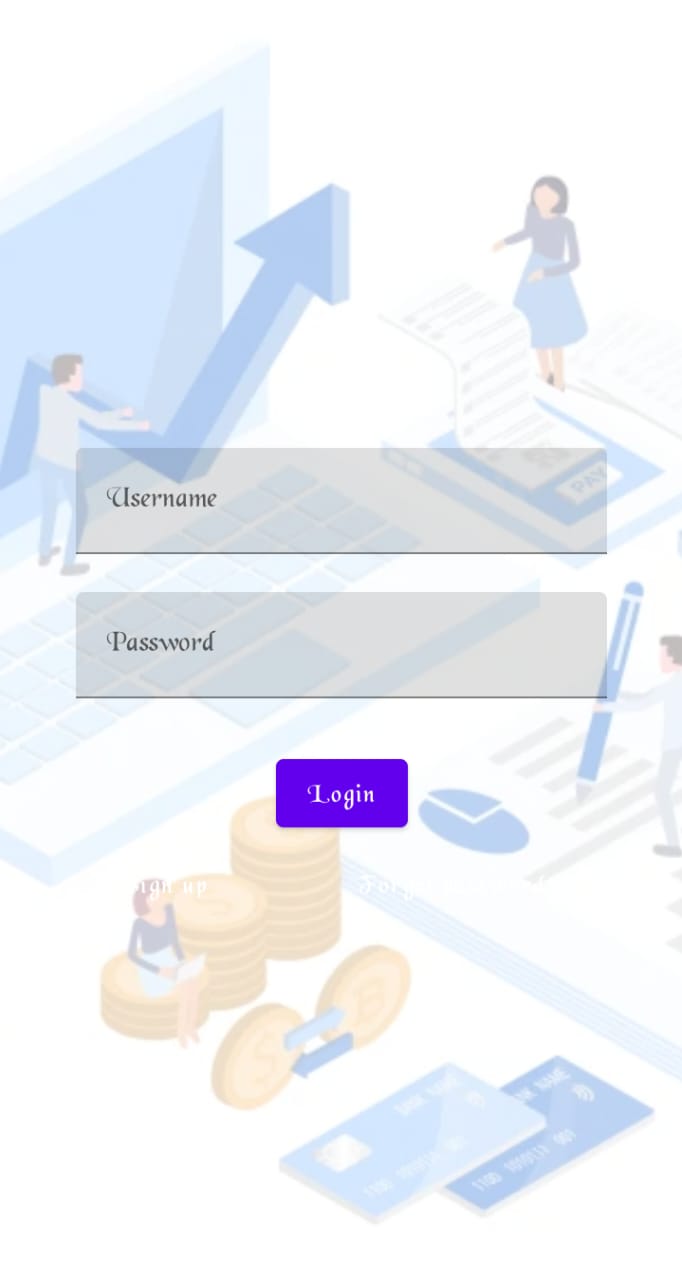


**2.2 Ideation & Brainstorming Map:**

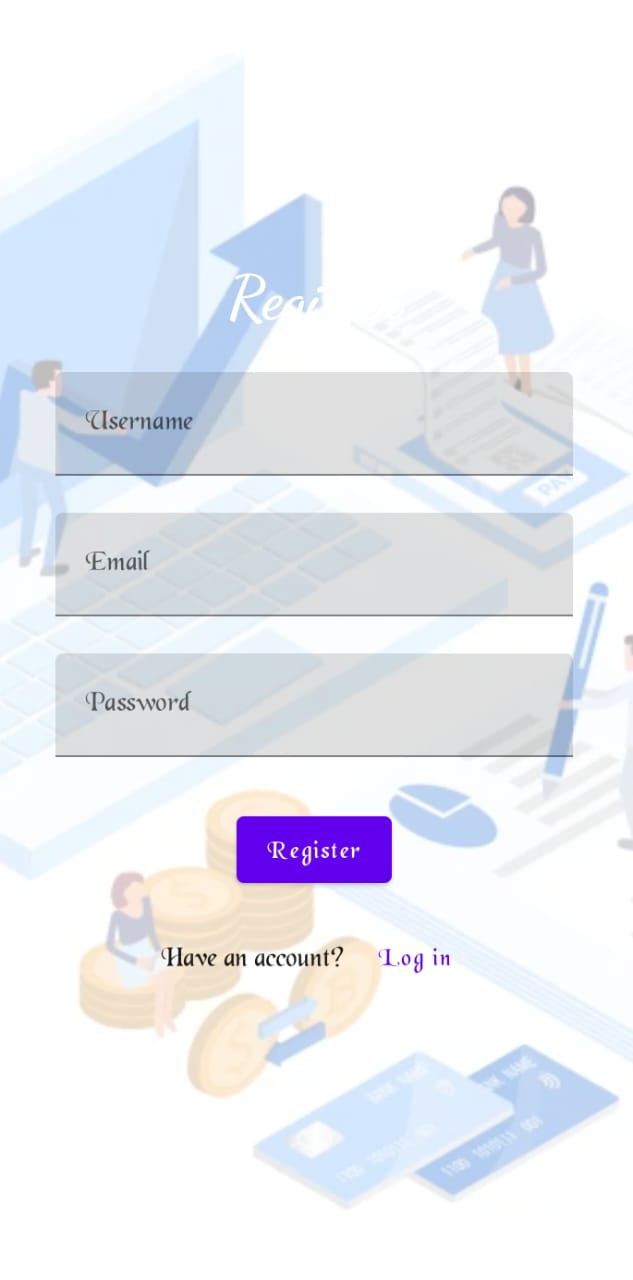


**RESULT**

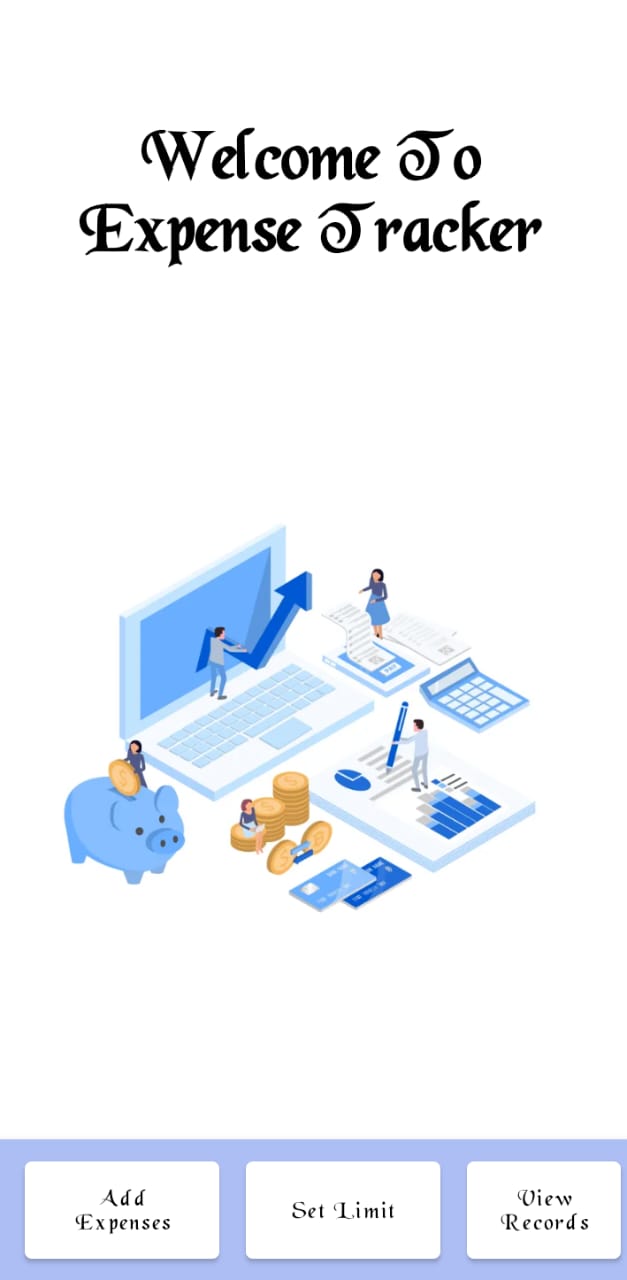
**Sign in page:**



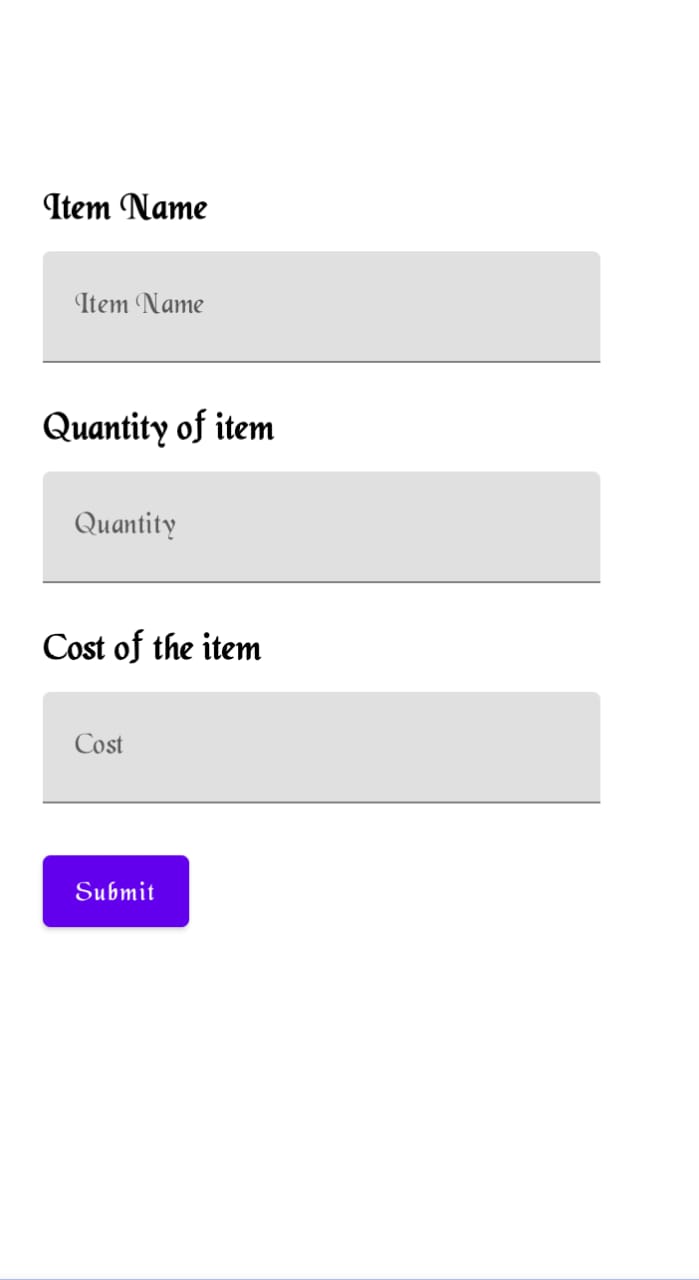
**Sign up page:**



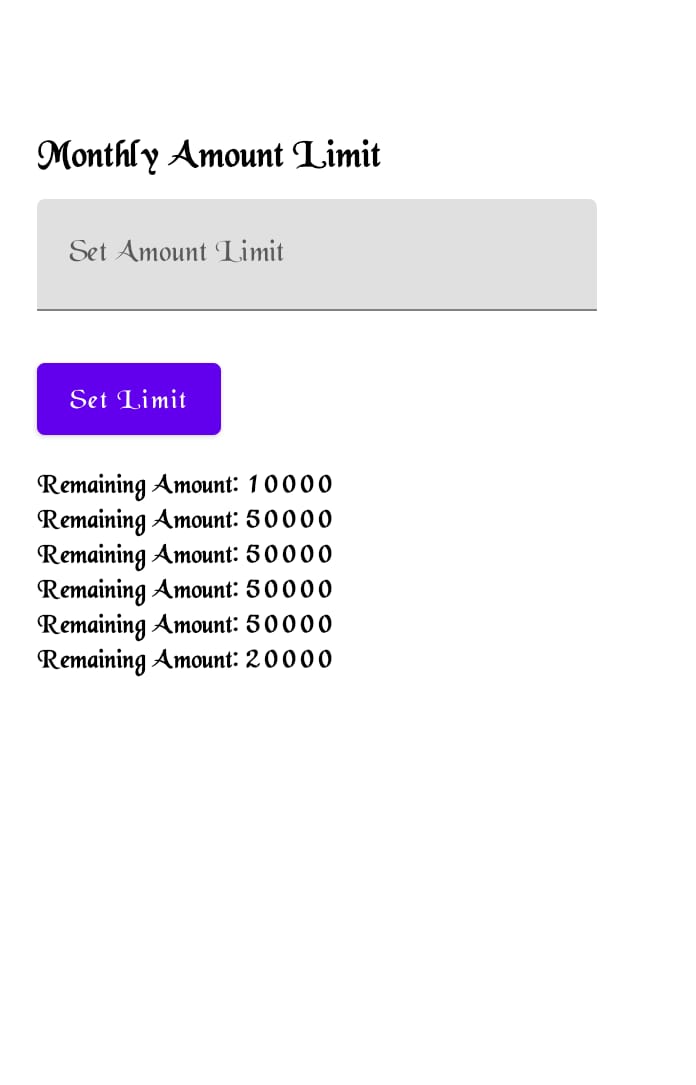
**Home page:**



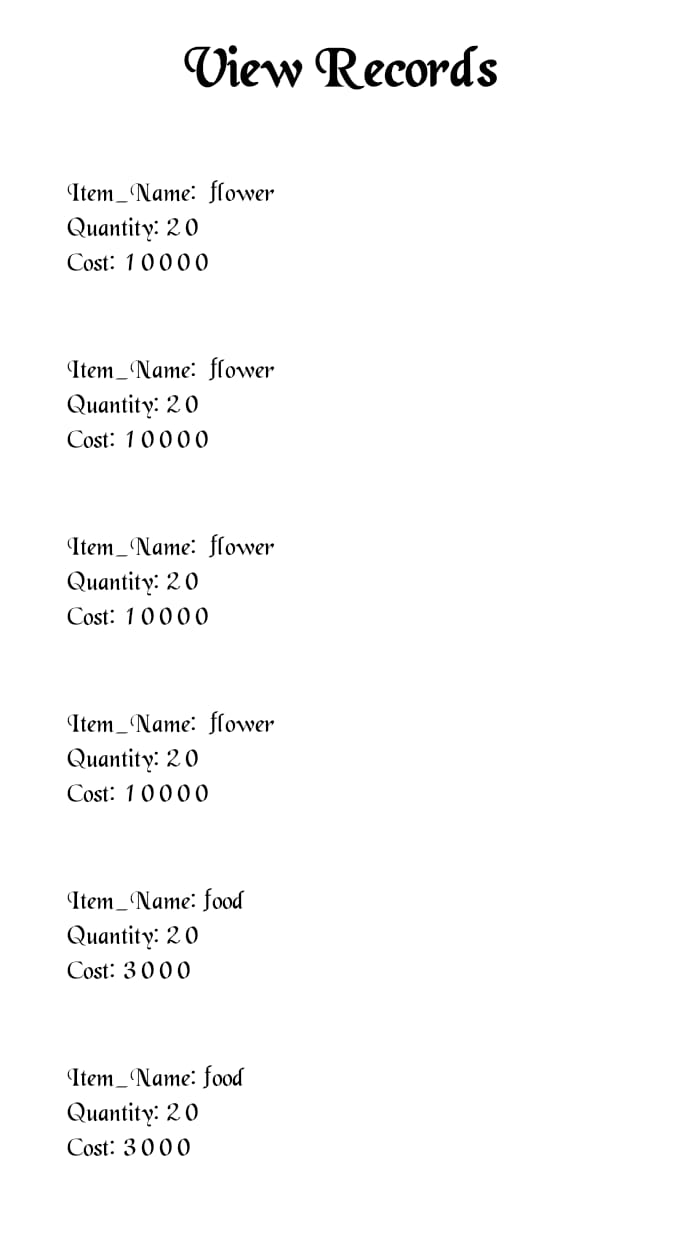
**Add Expenses page:**



Amount limit:



**View records page:**



**ADVANTAGES & DISADVANTAGES**

**Advantages:**

* Easy to use: Expenses tracker apps are designed to be user-friendly and easy to use, even for people who are not tech-savvy.
* Save time: By using an expenses tracker app, you can save time compared to manually tracking your expenses on paper or in a spreadsheet. You can easily input your expenses, categorize them, and track them all in one place.

**Disadvantages:**

* Cost: Some expenses tracker apps require a subscription or may have in-app purchases, which can add up over time.
* Learning curve: Even though expenses tracker apps are designed to be user-friendly, there may still be a learning curve to understand how to use the app effectively.
* Dependence on technology: If you rely too heavily on an expenses tracker app, you may be at a loss if you lose your phone or if there is a problem with the app.

**APPLICATIONS**

* Research and download an expenses tracker app from the Google Play Store. There are many options available, including popular apps like Mint, Personal Capital, and You Need a Budget (YNAB).
* Open the app and set up an account. You may need to enter personal information, such as your name, email, and password.
* Link your financial accounts, such as bank accounts, credit cards, and investment accounts. This will allow the app to automatically track your expenses and income.
* Create a budget by setting up categories for your expenses, such as groceries, entertainment, and transport.

**CONCLUSION**

In conclusion, an expenses tracker app can be a valuable tool for managing your finances on an Android device. It can help you stay organized, save time, avoid overspending, and make better financial decisions. However, it's important to choose an app that aligns with your financial goals and values, and to be aware of the potential disadvantages, such as cost, learning curve, and privacy concerns. By using an expenses tracker app in Android, you can take control of your finances and achieve your financial goals.

**FUTURE SCOPE**

In future scope of expenses tracker app in Android is promising, as more and more people are becoming interested in managing their finances through their smartphones. Here are some potential areas of growth and development for expenses tracker apps in the future. With the use of AI, expenses tracker apps can learn your spending habits and make personalized recommendations for saving money. This can include suggestions on how to reduce expenses or earn more income.

**APPENDIX**

**Source Code:**

https://github.com/Satheesh-Anand/Money-Matters

**Code:**

1. **AddExpensesActivity.kt**

package com.example.expensestracker

import android.annotation.SuppressLint

import android.content.Context

import android.content.Intent

import android.os.Bundle

import android.widget.Toast

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

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.platform.LocalContext

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 AddExpensesActivity : ComponentActivity() {

private lateinit var itemsDatabaseHelper: ItemsDatabaseHelper

private lateinit var expenseDatabaseHelper: ExpenseDatabaseHelper

@SuppressLint("UnusedMaterialScaffoldPaddingParameter")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

itemsDatabaseHelper = ItemsDatabaseHelper(this)

expenseDatabaseHelper = ExpenseDatabaseHelper(this)

setContent {

Scaffold(

// in scaffold we are specifying top bar.

bottomBar = {

// inside top bar we are specifying

// background color.

BottomAppBar(backgroundColor = Color(0xFFadbef4),

modifier = Modifier.height(80.dp),

// along with that we are specifying

// title for our top bar.

content = {

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {startActivity(Intent(applicationContext,AddExpensesActivity::class.java))},

colors=ButtonDefaults.buttonColors(backgroundColor=color.white),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "Add Expenses", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

SetLimitActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "Set Limit", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

ViewRecordsActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "View Records", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

}

)

}

) {

AddExpenses(this, itemsDatabaseHelper, expenseDatabaseHelper)

}

}

}

}

@SuppressLint("Range")

@Composable

fun AddExpenses(context: Context, itemsDatabaseHelper: ItemsDatabaseHelper, expenseDatabaseHelper: ExpenseDatabaseHelper) {

Column(

modifier = Modifier

.padding(top = 100.dp, start = 30.dp)

.fillMaxHeight()

.fillMaxWidth(),

horizontalAlignment = Alignment.Start

) {

valmContext = LocalContext.current

var items by remember { mutableStateOf("") }

var quantity by remember { mutableStateOf("") }

var cost by remember { mutableStateOf("") }

var error by remember { mutableStateOf("") }

Text(text = "Item Name", fontWeight = FontWeight.Bold, fontSize = 20.sp)

Spacer(modifier = Modifier.height(10.dp))

TextField(value = items, onValueChange = { items = it },

label = { Text(text = "Item Name") })

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

Text(text = "Quantity of item", fontWeight = FontWeight.Bold, fontSize = 20.sp)

Spacer(modifier = Modifier.height(10.dp))

TextField(value = quantity, onValueChange = { quantity = it },

label = { Text(text = "Quantity") })

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

Text(text = "Cost of the item", fontWeight = FontWeight.Bold, fontSize = 20.sp)

Spacer(modifier = Modifier.height(10.dp))

TextField(value = cost, onValueChange = { cost = it },

label = { Text(text = "Cost") })

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

if (error.isNotEmpty()) {

Text(

text = error,

color = MaterialTheme.colors.error,

modifier = Modifier.padding(vertical = 16.dp)

)

}

Button(onClick = {

if (items.isNotEmpty() &&quantity.isNotEmpty() &&cost.isNotEmpty()) {

val items = Items(

id = null,

itemName = items,

quantity = quantity,

cost = cost

)

val limit= expenseDatabaseHelper.getExpenseAmount(1)

valactualvalue = limit?.minus(cost.toInt())

// Toast.makeText(mContext, actualvalue.toString(), Toast.LENGTH\_SHORT).show()

val expense = Expense(

id = 1,

amount = actualvalue.toString()

)

if (actualvalue != null) {

if (actualvalue< 1) {

Toast.makeText(mContext, "Limit Over", Toast.LENGTH\_SHORT).show()

} else {

expenseDatabaseHelper.updateExpense(expense)

itemsDatabaseHelper.insertItems(items)

}

}

}

}) {

Text(text = "Submit")

}

}

}

1. **LoginActivity.kt**

package com.example.expensestracker

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.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.FontWeightimportandroidx.compose.ui.text.input.PasswordVisualTransformation

import androidx.compose.ui.text.input.VisualTransformation

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import androidx.core.content.ContextCompat

import com.example.expensestracker.ui.theme.ExpensesTrackerTheme

class LoginActivity : ComponentActivity() {

private lateinit var databaseHelper: UserDatabaseHelper

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

databaseHelper = UserDatabaseHelper(this)

setContent {

ExpensesTrackerTheme {

// A surface container using the 'background' color from the theme

Surface(

modifier = Modifier.fillMaxSize(),

color = MaterialTheme.colors.background

) {

LoginScreen(this, databaseHelper)

}

}

}

}

}

@Composable

fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {

Image(

painterResource(id = R.drawable.img\_1), contentDescription = "",

alpha =0.3F,

contentScale = ContentScale.FillHeight,

)

var username by remember { mutableStateOf("") }

var password by remember { mutableStateOf("") }

var error by remember { mutableStateOf("") }

Column(

modifier = Modifier.fillMaxSize(),

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center

) {

Text(

fontSize = 36.sp,

fontWeight = FontWeight.ExtraBold,

fontFamily = FontFamily.Cursive,

color = Color.White,

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") },

modifier = Modifier.padding(10.dp)

.width(280.dp),

visualTransformation = PasswordVisualTransformation()

)

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(color = Color.White,text = "Sign up") }

TextButton(onClick = {

})

{

Spacer(modifier = Modifier.width(60.dp))

Text(color = Color.White,text = "Forget password?")

}

}

}

}

private fun startMainPage(context: Context) {

val intent = Intent(context, MainActivity::class.java)

ContextCompat.startActivity(context, intent, null)

}

1. **MainActivity.kt**

package com.example.expensestracker

import android.annotation.SuppressLint

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.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.res.painterResource

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.text.style.TextAlign

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import com.example.expensestracker.ui.theme.ExpensesTrackerTheme

class MainActivity : ComponentActivity() {

@SuppressLint("UnusedMaterialScaffoldPaddingParameter")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

Scaffold(

// in scaffold we are specifying top bar.

bottomBar = {

// inside top bar we are specifying

// background color.

BottomAppBar(backgroundColor = Color(0xFFadbef4),

modifier = Modifier.height(80.dp),

// along with that we are specifying

// title for our top bar.

content = {

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {startActivity(Intent(applicationContext,AddExpensesActivity::class.java))},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "Add Expenses", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

SetLimitActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "Set Limit", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

ViewRecordsActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "View Records", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

}

)

}

) {

MainPage()

}

}

}

}

@Composable

fun MainPage() {

Column(

modifier = Modifier.padding(20.dp).fillMaxSize(),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

Text(text = "Welcome To Expense Tracker", fontSize = 42.sp, fontWeight = FontWeight.Bold,

textAlign = TextAlign.Center)

Image(painterResource(id = R.drawable.img\_1), contentDescription ="", modifier = Modifier.size(height = 500.dp, width = 500.dp))

}

}

1. **RegisterActivity.kt**

package com.example.expensestracker

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.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

import com.example.expensestracker.ui.theme.ExpensesTrackerTheme

class RegisterActivity : ComponentActivity() {

private lateinit var databaseHelper: UserDatabaseHelper

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

databaseHelper = UserDatabaseHelper(this)

setContent {

ExpensesTrackerTheme {

// A surface container using the 'background' color from the theme

Surface(

modifier = Modifier.fillMaxSize(),

color = MaterialTheme.colors.background

) {

RegistrationScreen(this,databaseHelper)

}

}

}

}

}

@Composable

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

Image(

painterResource(id = R.drawable.img\_1), contentDescription = "",

alpha =0.3F,

contentScale = ContentScale.FillHeight,

)

var username by remember { mutableStateOf("") }

var password by remember { mutableStateOf("") }

var email by remember { mutableStateOf("") }

var error by remember { mutableStateOf("") }

Column(

modifier = Modifier.fillMaxSize(),

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center

) {

Text(

fontSize = 36.sp,

fontWeight = FontWeight.ExtraBold,

fontFamily = FontFamily.Cursive,

color = Color.White,

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") },

modifier = Modifier

.padding(10.dp)

.width(280.dp),

visualTransformation = PasswordVisualTransformation()

)

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)

}

1. **SetLimitActivity.kt**

package com.example.expensestracker

import android.annotation.SuppressLint

import android.content.Context

import android.content.Intent

import android.os.Bundle

import android.util.Log

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.layout.\*

import androidx.compose.foundation.lazy.LazyColumn

import androidx.compose.foundation.lazy.LazyRow

import androidx.compose.foundation.lazy.items

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.text.font.FontWeight

import androidx.compose.ui.text.style.TextAlign

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import com.example.expensestracker.ui.theme.ExpensesTrackerTheme

class SetLimitActivity : ComponentActivity() {

private lateinit var expenseDatabaseHelper: ExpenseDatabaseHelper

@SuppressLint("UnusedMaterialScaffoldPaddingParameter")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

expenseDatabaseHelper = ExpenseDatabaseHelper(this)

setContent {

Scaffold(

// in scaffold we are specifying top bar.

bottomBar = {

// inside top bar we are specifying

// background color.

BottomAppBar(backgroundColor = Color(0xFFadbef4),

modifier = Modifier.height(80.dp),

// along with that we are specifying

// title for our top bar.

content = {

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

AddExpensesActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "Add Expenses", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

SetLimitActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "Set Limit", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

ViewRecordsActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "View Records", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

}

)

}

) {

val data=expenseDatabaseHelper.getAllExpense();

Log.d("swathi" ,data.toString())

val expense = expenseDatabaseHelper.getAllExpense()

Limit(this, expenseDatabaseHelper,expense)

}

}

}

}

@Composable

fun Limit(context: Context, expenseDatabaseHelper: ExpenseDatabaseHelper, expense: List<Expense>) {

Column(

modifier = Modifier

.padding(top = 100.dp, start = 30.dp)

.fillMaxHeight()

.fillMaxWidth(),

horizontalAlignment = Alignment.Start

) {

var amount by remember { mutableStateOf("") }

var error by remember { mutableStateOf("") }

Text(text = "Monthly Amount Limit", fontWeight = FontWeight.Bold, fontSize = 20.sp)

Spacer(modifier = Modifier.height(10.dp))

TextField(value = amount, onValueChange = { amount = it },

label = { Text(text = "Set Amount Limit ") })

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

if (error.isNotEmpty()) {

Text(

text = error,

color = MaterialTheme.colors.error,

modifier = Modifier.padding(vertical = 16.dp)

)

}

Button(onClick = {

if (amount.isNotEmpty()) {

val expense = Expense(

id = null,

amount = amount

)

expenseDatabaseHelper.insertExpense(expense)

}

}) {

Text(text = "Set Limit")

}

Spacer(modifier = Modifier.height(10.dp))

LazyRow(

modifier = Modifier

.fillMaxSize()

.padding(top = 0.dp),

horizontalArrangement = Arrangement.Start

) {

item {

LazyColumn {

items(expense) { expense ->

Column(

) {

Text("Remaining Amount: ${expense.amount}", fontWeight = FontWeight.Bold)

}

}

}

}

}

}

}

1. **ViewRecordActivity.kt**

package com.example.expensestracker

import android.annotation.SuppressLint

import android.content.Intent

import android.os.Bundle

import android.util.Log

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.ScrollState

import androidx.compose.foundation.layout.\*

import androidx.compose.foundation.lazy.LazyColumn

import androidx.compose.foundation.lazy.LazyRow

import androidx.compose.foundation.lazy.items

import androidx.compose.foundation.verticalScroll

import androidx.compose.material.\*

import androidx.compose.runtime.Composable

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.text.style.TextAlign

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import com.example.expensestracker.ui.theme.ExpensesTrackerTheme

class ViewRecordsActivity : ComponentActivity() {

private lateinit var itemsDatabaseHelper: ItemsDatabaseHelper

@SuppressLint("UnusedMaterialScaffoldPaddingParameter", "SuspiciousIndentation")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

itemsDatabaseHelper = ItemsDatabaseHelper(this)

setContent {

Scaffold(

// in scaffold we are specifying top bar.

bottomBar = {

// inside top bar we are specifying

// background color.

BottomAppBar(backgroundColor = Color(0xFFadbef4),

modifier = Modifier.height(80.dp),

// along with that we are specifying

// title for our top bar.

content = {

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

AddExpensesActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "Add Expenses", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

SetLimitActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "Set Limit", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

Spacer(modifier = Modifier.width(15.dp))

Button(

onClick = {

startActivity(

Intent(

applicationContext,

ViewRecordsActivity::class.java

)

)

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),

modifier = Modifier.size(height = 55.dp, width = 110.dp)

)

{

Text(

text = "View Records", color = Color.Black, fontSize = 14.sp,

textAlign = TextAlign.Center

)

}

}

)

}

) {

val data=itemsDatabaseHelper.getAllItems();

Log.d("swathi" ,data.toString())

val items = itemsDatabaseHelper.getAllItems()

Records(items)

}

}

}

}

@Composable

fun Records(items: List<Items>) {

Text(text = "View Records", modifier = Modifier.padding(top = 24.dp, start = 106.dp, bottom = 24.dp ), fontSize = 30.sp, fontWeight = FontWeight.Bold)

Spacer(modifier = Modifier.height(30.dp))

LazyRow(

modifier = Modifier

.fillMaxSize()

.padding(top = 80.dp),

horizontalArrangement = Arrangement.SpaceBetween

){

item {

LazyColumn {

items(items) { items ->

Column(modifier = Modifier.padding(top = 16.dp, start = 48.dp, bottom = 20.dp)) {

Text("Item\_Name: ${items.itemName}")

Text("Quantity: ${items.quantity}")

Text("Cost: ${items.cost}")

}

}

}

}

}

}