Android Application Development

Money Matters: A Personal Finance Management App

Team ID: NM2024TMID05685

Yazhini M (Team Leader): 80D492EB26E71D0BF274D2DA25C406FB

Agalya P (Member 1): A3086354B1F2EE0A60A4FD79B3846DAB

Kausika S V (Member 2): 9CC36C497AD69C898DB48A0FB740D1FC

Pramothini C (Member 3): 658B467BB21D16DA72A4981868BC0752

SEMESTER-V

B.E COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR (2024-2025)

ANNA UNIVERSITY REGIONAL CAMPUS COIMBATORE

COIMBATORE-641046

NOVEMBER-2024

Table of Contents

Project Description:

In today's fast-paced world, managing personal finances effectively is more important than ever. Individuals face constant financial pressures, from paying bills to managing debts, tracking expenses, and saving for future goals. Yet, only a small percentage of people maintain an organized approach to managing their finances, often due to the lack of a straightforward and accessible tool. **Money Matters**, a personal finance management app, aims to bridge this gap by providing a user-friendly platform to simplify and enhance financial management for individuals.

Purpose of the app:

The primary purpose of Money Matters is to offer a comprehensive tool that enables users to monitor, analyze, and optimize their financial activities. The app empowers individuals to take charge of their finances through intuitive tools, clear visuals, and timely insights. By offering features such as expense tracking, budgeting, and goal-setting, Money Matters not only encourages better financial habits but also supports long-term financial wellness.

Functionalities:

Money Matters offers a range of features designed to address the diverse needs of personal finance management:

1. **Expense Tracking**: Users can record and categorize their spending to gain a clear understanding of their monthly and yearly expenses. With features that allow the tracking of spending across different categories (e.g., groceries, utilities, entertainment), users can quickly identify areas where they might be overspending.

2. **Budgeting Tools**: The app allows users to set monthly budgets and provides alerts when they are nearing their budget limits. This helps users maintain control over their spending habits, making it easier to avoid unnecessary expenses.
3. **Financial Goal Setting**: Money Matters enables users to set both short-term and long-term financial goals, such as saving for a vacation or building an emergency fund. The app tracks progress towards these goals, providing motivation and accountability.
4. **Income and Investment Tracking**: Users can track multiple income sources and monitor the performance of investments. This feature encourages users to keep an eye on wealth-building opportunities and optimize their financial growth.
5. **Reports and Analytics**: Money Matters provides regular financial reports and personalized insights that help users analyze their financial habits. With clear charts and visuals, users can easily track their financial health and adjust their habits as needed. **Category Management**
6. Allow users to categorize their expenses (e.g., Food, Entertainment, Transportation) with support for custom categories. This enables better organization and insights into spending habits.

Additional Requirements:

1. Category Management

- Allow users to categorize their expenses (e.g., Food, Entertainment, Transportation) with support for custom categories. This enables better organization and insights into spending habits.

2.Recurring Expenses

- Provide an option for users to set recurring expenses (e.g., rent, subscriptions) so these can be automatically tracked monthly or weekly, saving users time and improving accuracy.

3.Data Visualization

- Use charts and graphs (like pie or bar charts) to visually represent spending patterns and trends. This can help users quickly identify where their money is going.

4.Income Tracking

- Add functionality for users to input income sources and amounts. This allows the app to calculate the remaining budget in relation to the income, offering a fuller financial picture.

5.Export and Import Data

- Enable data export (to formats like CSV or PDF) and import, so users can back up their data, review it outside the app, or carry over past expenses to a new device.

Benefits of Money Matters:

- **Simplifies Financial Management**: Combines expense tracking, budgeting, and goal-setting in one app, reducing the need for multiple tools and spreadsheets.
- **Saves Time**: By providing a consolidated view of finances, Money Matters reduces the time spent tracking expenses and reviewing financial data.

- **Promotes Better Financial Habits**: Encourages users to consistently monitor spending, set realistic budgets, and work toward financial goals.
- **Improves Financial Awareness**: Through regular reports and analytics, users gain insight into their financial health, enabling smarter financial decisions.
- **Encourages Proactive Planning**: Helps users set and track both short-term and long-term financial goals, fostering a forward-thinking approach to money management.
- **Enhances Security**: Employs data encryption and authentication features to ensure that sensitive financial information is well-protected.

SOURCE CODE:

LoginActivity:

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

RegisterActivity:

```
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_2), 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.Black,
      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)
}
```

MainActivity:

```
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))

    }

}
```

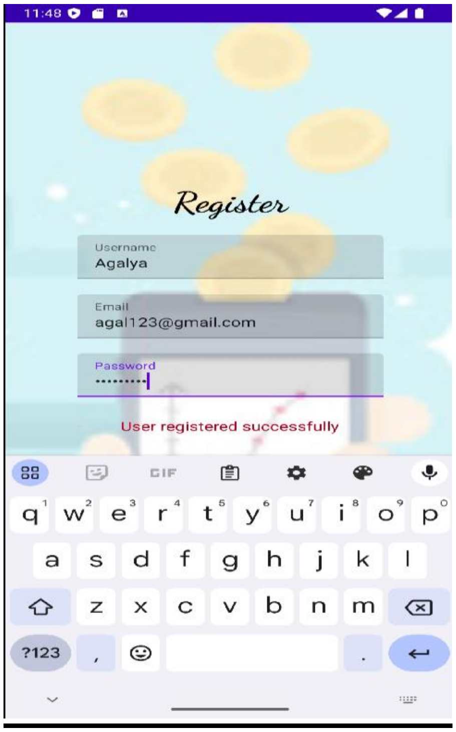
ViewRecordsActivity:

```
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}")
                    }
                }
            }
        }
    }
}
```

Output Screenshots:

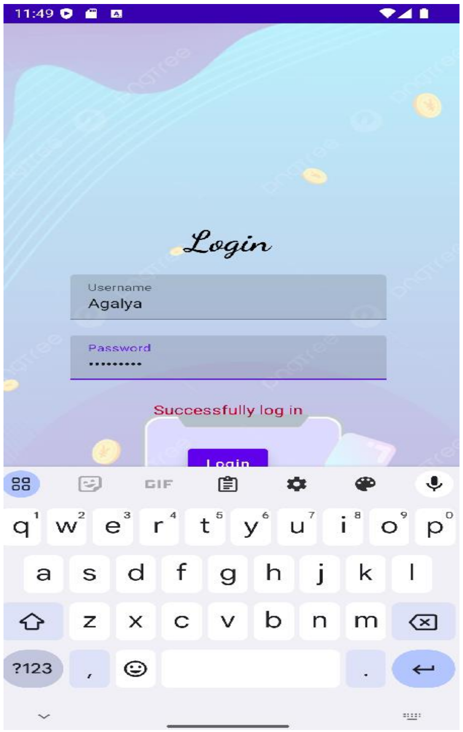


Username
Agalya

Email
agal123@gmail.com

Password
•••••••••

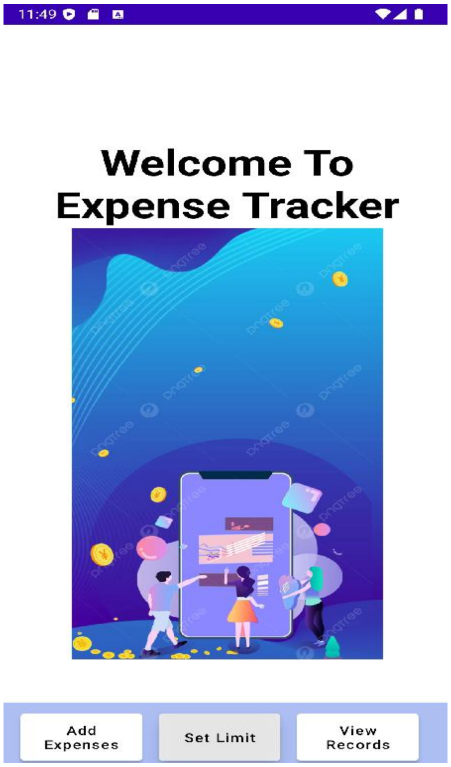
User registered successfully



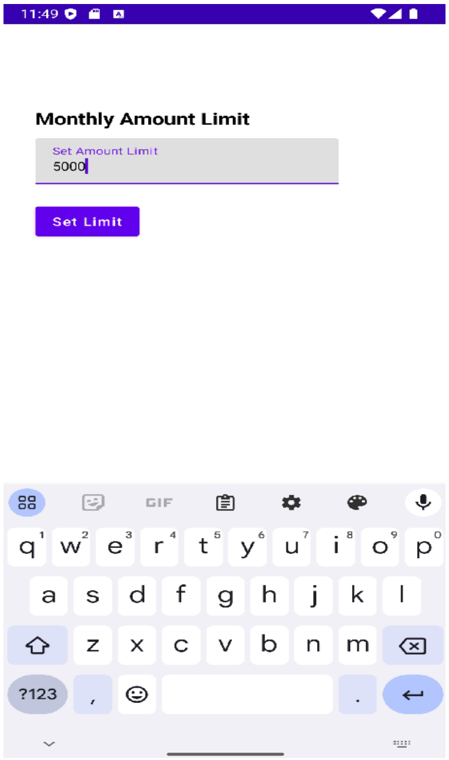
Username
Agalya

Password
•••••••••

Successfully log in

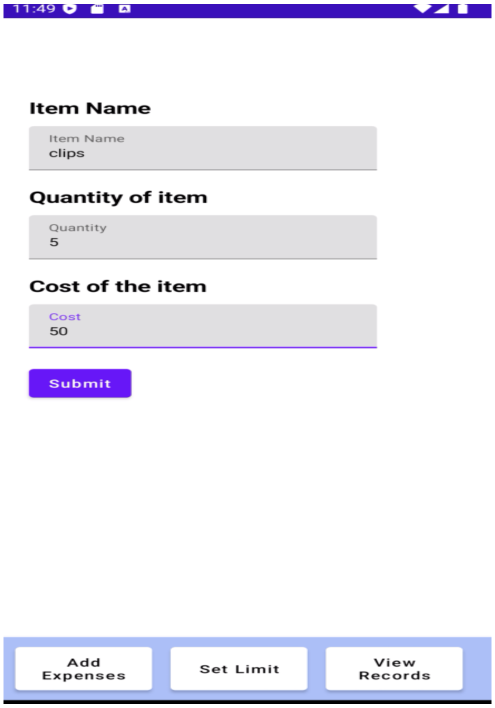


Welcome To Expense Tracker



Monthly Amount Limit

Set Amount Limit: 5000



Item Name

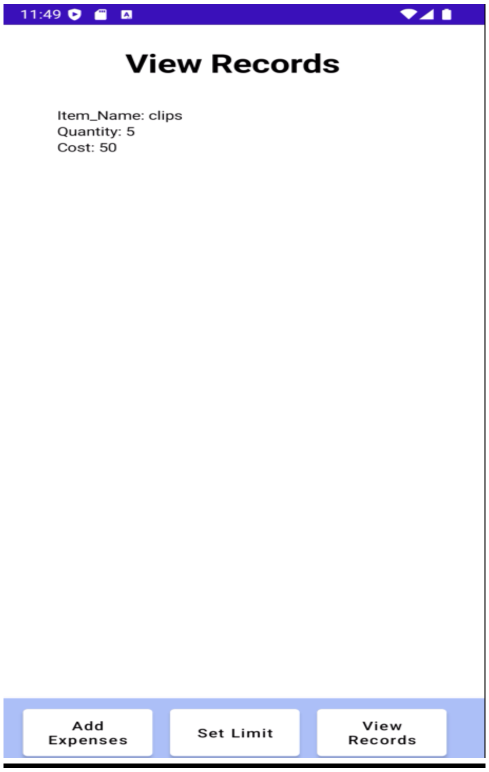
Item Name: clips

Quantity of item

Quantity: 5

Cost of the item

Cost: 50



View Records

Item_Name: clips
Quantity: 5
Cost: 50

Demo Video Link:

https://drive.google.com/file/d/1LA8M7uFCHQBNvJXy5OxTjkatm1kmcL2W/view?usp=sharing