1. INTRODUCTION

1.1 OVERVIEW

A personal finance management apps come with a range of features that can help you manage your money more efficiently. They allow you to create a budget and track your spending, categorize expenses, and set financial goals. By using a personal finance management app, you can take control of your finances, improve your financial literacy, and make informed decisions about your money.

1.2 PURPOSE

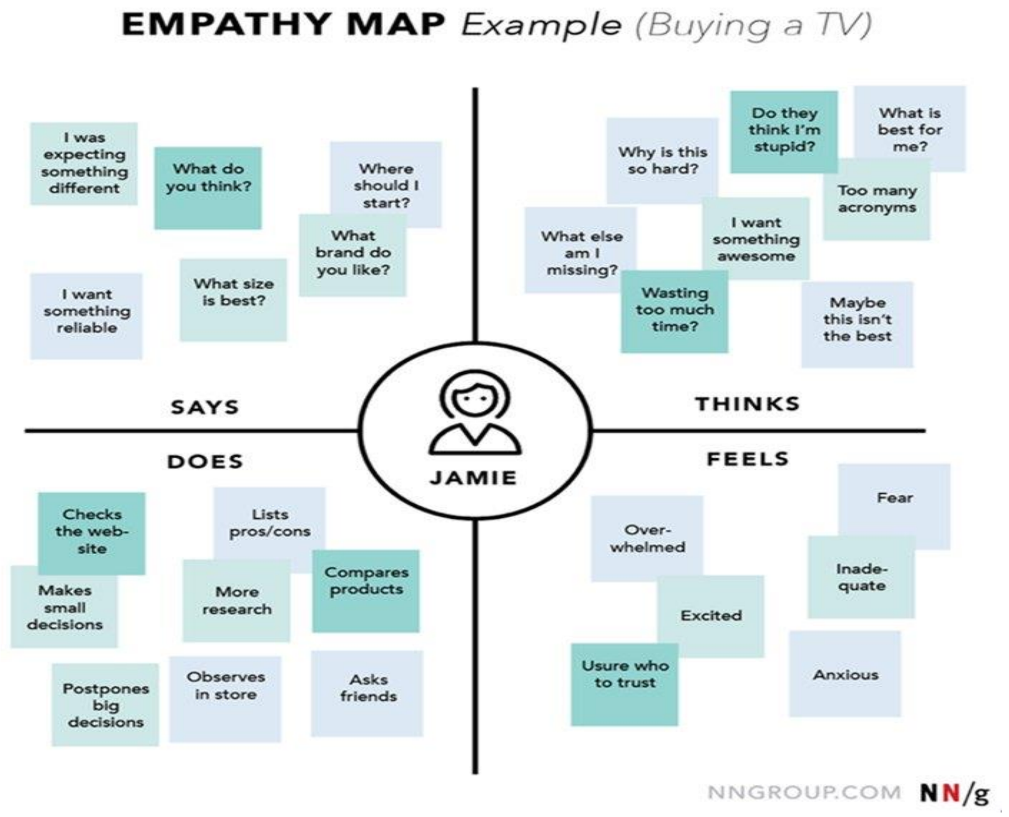
The purpose of a personal finance management app is to help individuals track their expenses, budget their money, and plan for their financial goals . It provides users with a clear understanding of their spending patterns and helps them make informed decisions about their finances.

The app also offers customized tools that allow users to create and manage their budgets, set savings goals, and monitor their progress towards achieving their financial objectives.

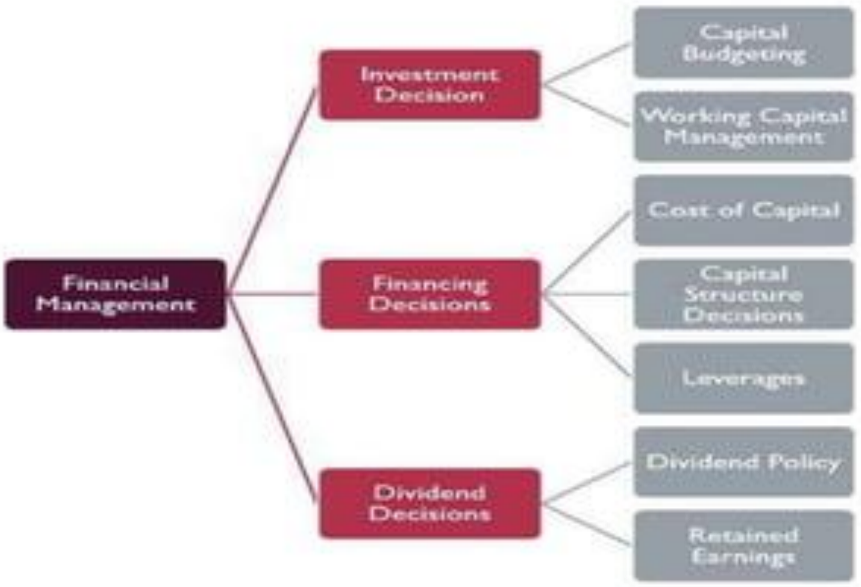
Ultimately, the app enables individuals to take control of their finances, manage their debts, and build a secure financial future.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 EMPATHY MAP

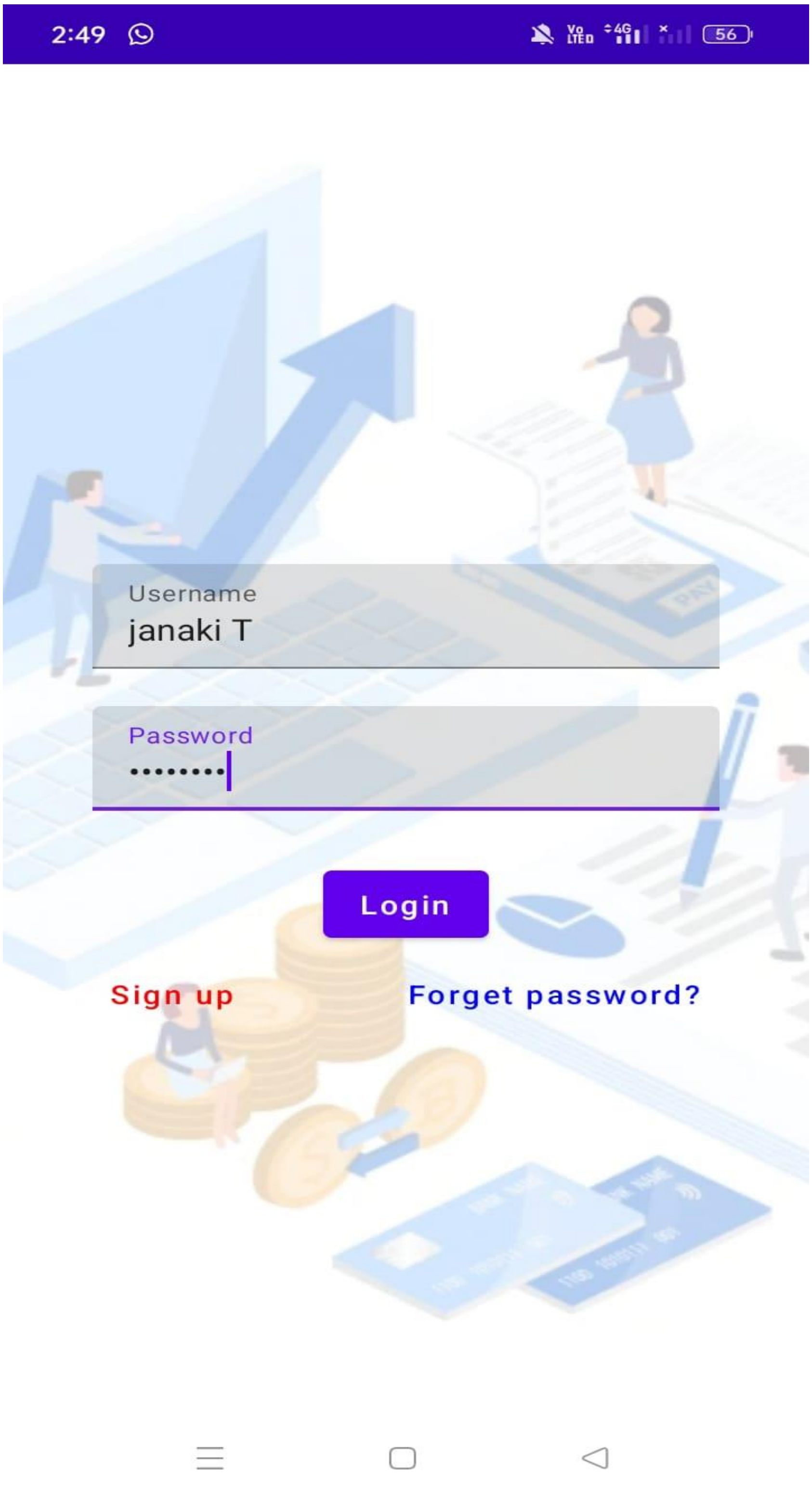


2.2 IDEATION & BRAINSTROMING MAP



3. RESULT

LOGIN PAGE



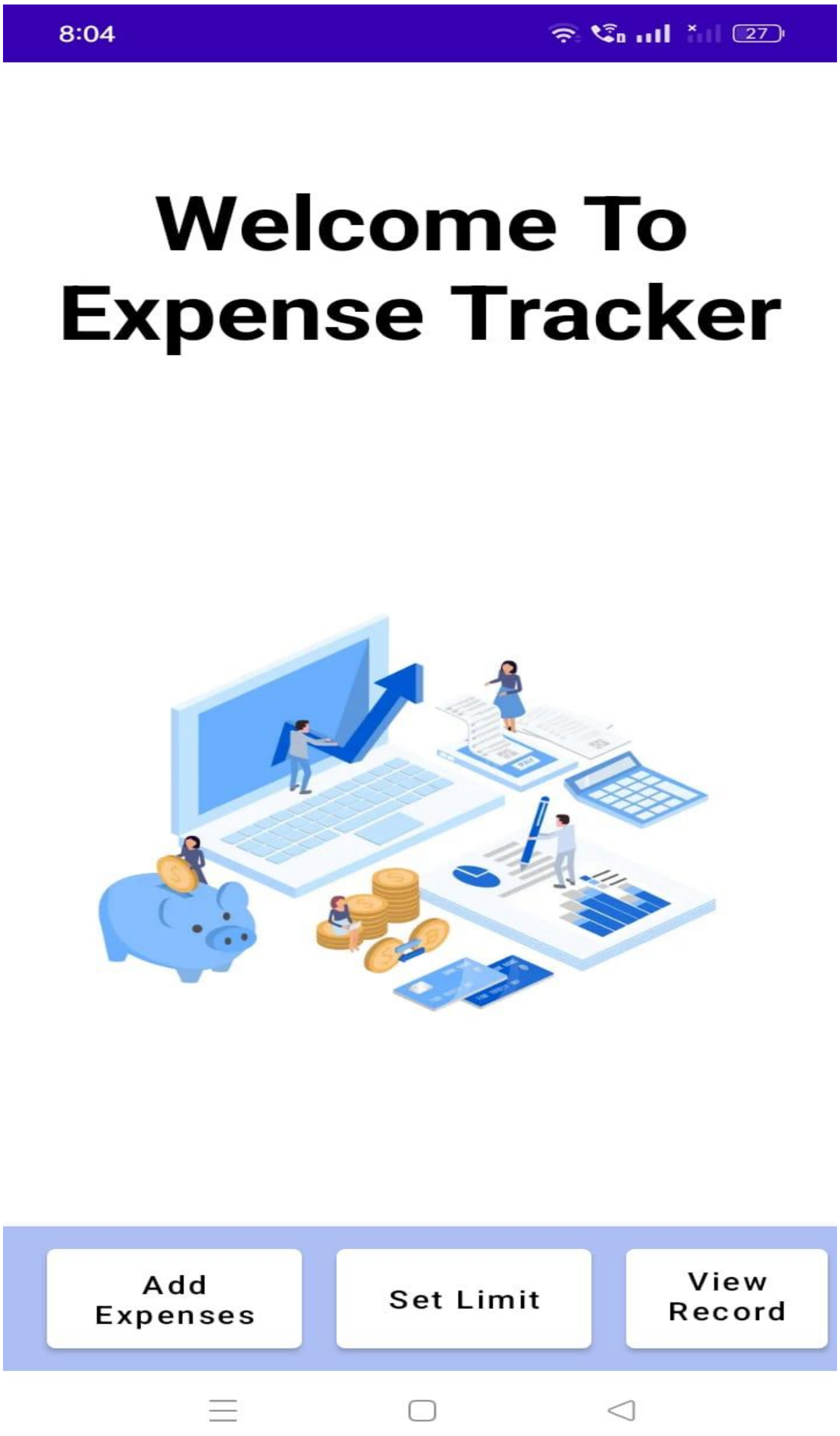
Username

janaki T

Password

•••••••••

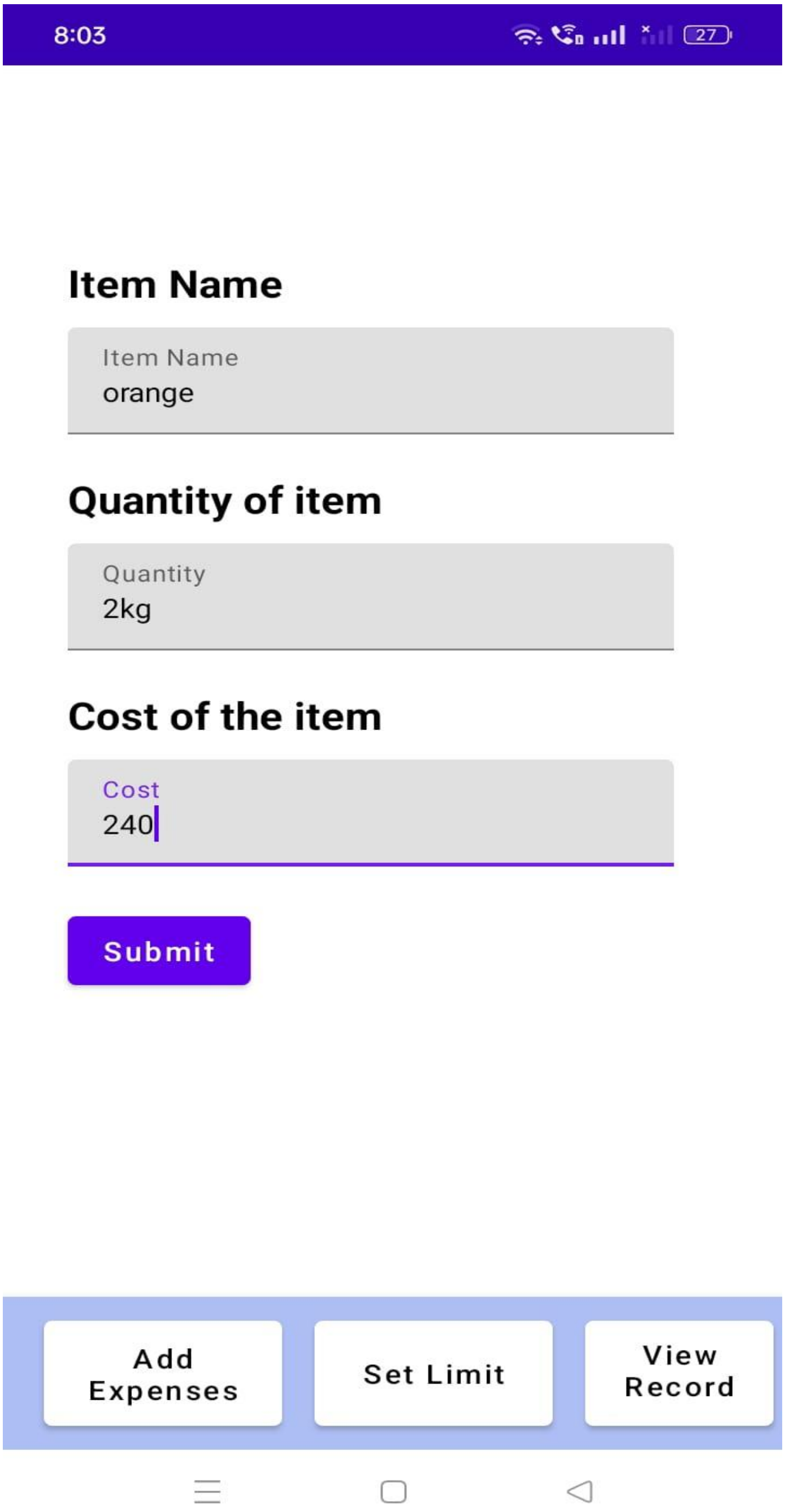
DASHBOARD PAGE



8:04

Welcome To Expense Tracker

VIEW PAGE



8:03

Item Name

Item Name
orange

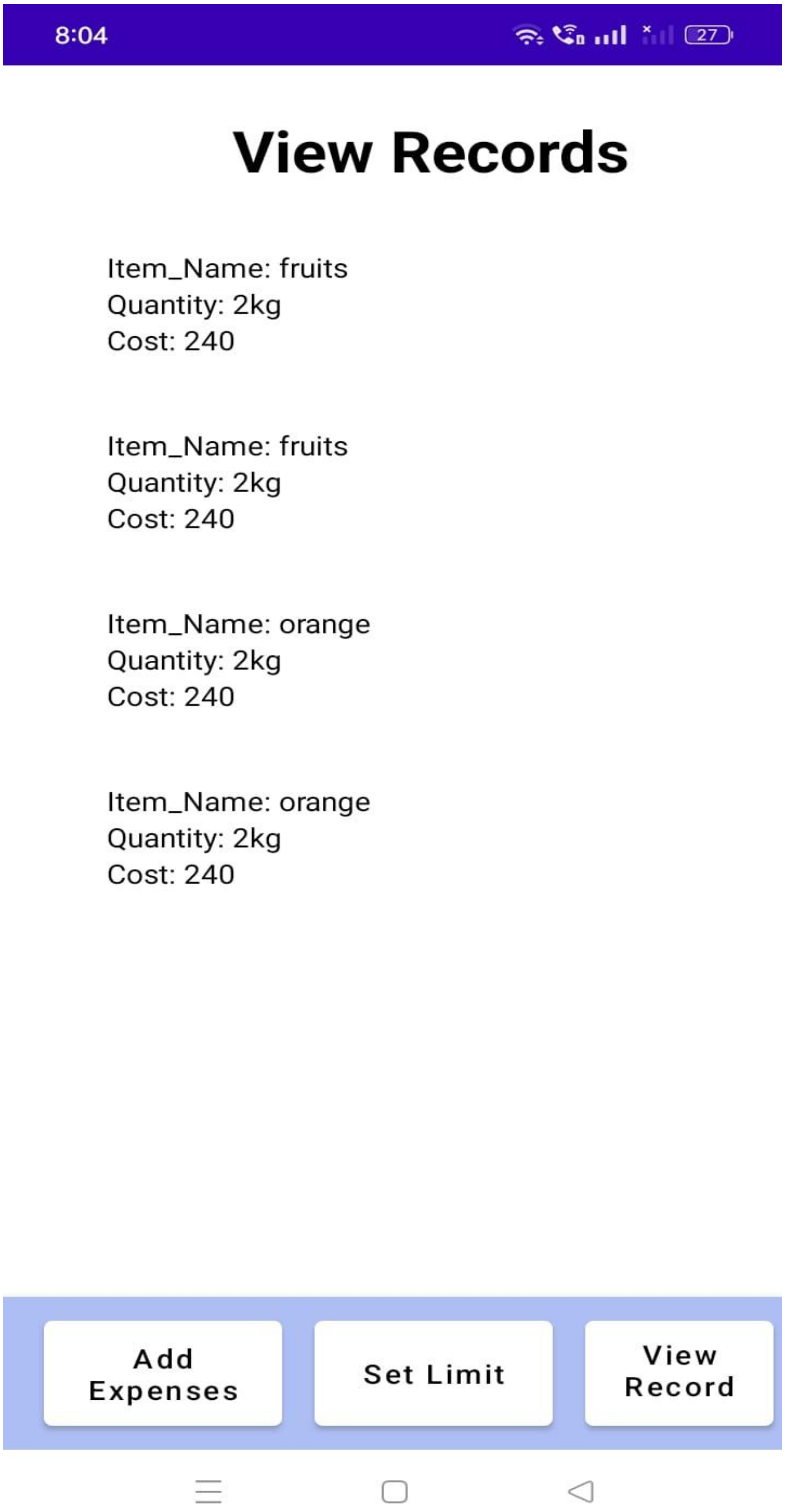
Quantity of item

Quantity
2kg

Cost of the item

Cost
240

VIEW RECORDS



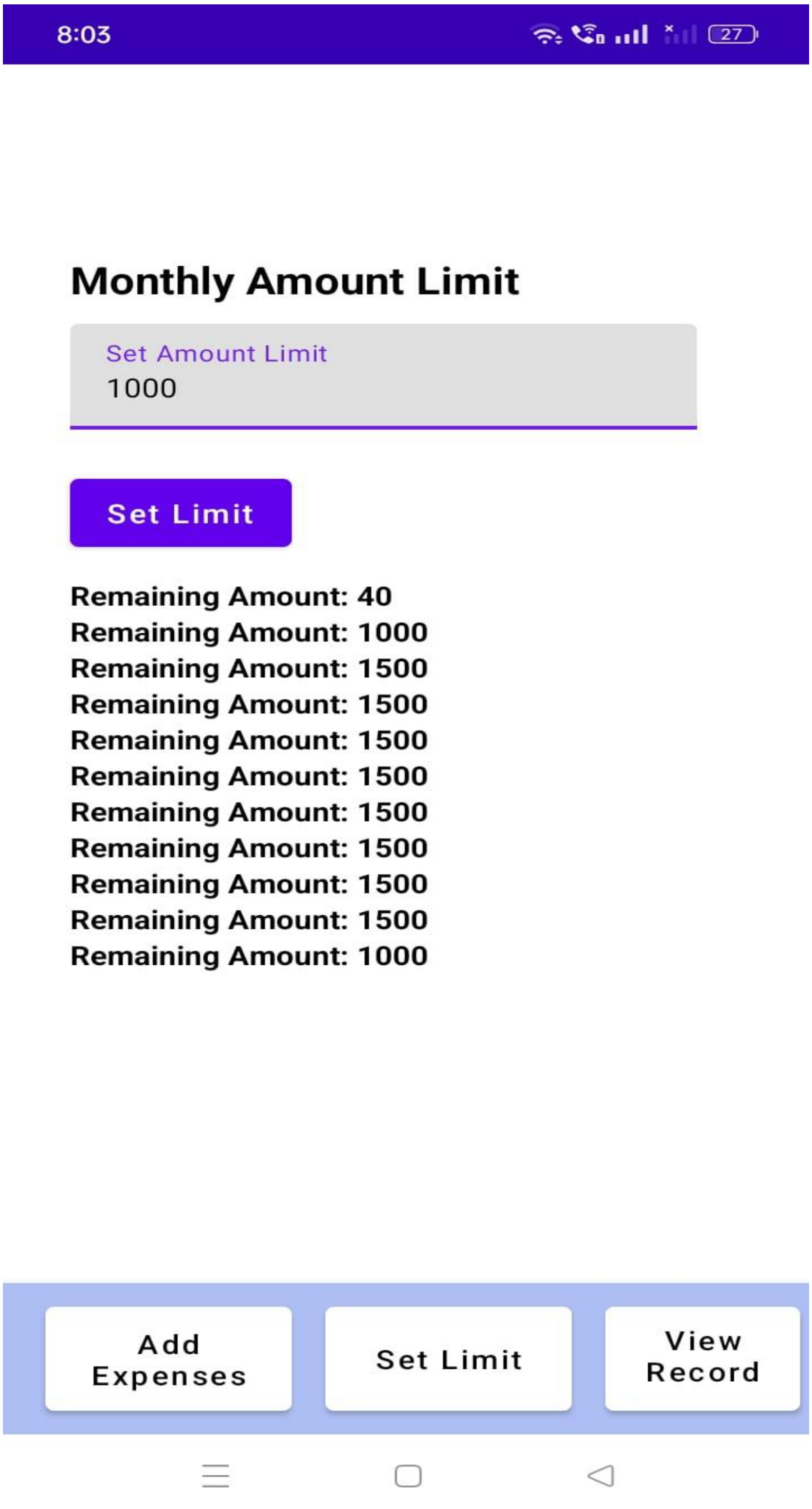
View Records

Item_Name: fruits
Quantity: 2kg
Cost: 240

Item_Name: fruits
Quantity: 2kg
Cost: 240

Item_Name: orange
Quantity: 2kg
Cost: 240

Item_Name: orange
Quantity: 2kg
Cost: 240



Monthly Amount Limit

Set Amount Limit
1000

Remaining Amount: 40
Remaining Amount: 1000
Remaining Amount: 1500
Remaining Amount: 1500
Remaining Amount: 1500
Remaining Amount: 1500
Remaining Amount: 1500
Remaining Amount: 1500
Remaining Amount: 1500
Remaining Amount: 1500
Remaining Amount: 1000

4. ADVANTAGES & DISADVANTAGES

4.1 ADVANTAGES

1. A personal finance management app can help you keep track of how much money you are spending on various things so that you can make adjustments to your spending habits.

2. Many personal finance apps offer insights into your spending habits, such as how much you spend on groceries or entertainment each month.

3. Personal finance apps are often very easy to use, saving you time and effort compared to tracking expenses manually.

4. By tracking your expenses and offering insights into your spending habits, a personal finance app can help you save money over time.

5. Many personal finance apps provide reminders for you to pay bills or to stick to your budget.

4.2 DISADVANTAGES

1. Some personal finance apps may not have adequate security measures in place to protect your personal and financial information.

2. Since personal finance apps rely on users to input data, there is a risk of errors being made in the information entered.

3. Even with the help of a personal finance app, managing your finances still requires time and effort.

4. Some personal finance apps may not offer all of the features and functionality that you need.

5. Some personal finance apps may require a subscription or change fees for certain features.

5. APPLICATION

1. The app allows users to set budgets for various categories such as groceries, rent, and entertainment. Users can track their spending against those budgets.

2. The app enables users to track their expenses by linking their bank accounts and credit cards to the app.

3. Users can rest assured that their financial with the app

6. CONCLUSION

Personal finance management apps can be an effective tool to track and manage your finances. They offer a range of features that simplify the process of budgeting, saving, investing and tracking expenses. It is important to choose an app that suits your individual needs and preferences in order to achieve financial success.

7. FEATURE SCOPE

A personal finance management app should allow users to create and track their budget. It should enable users to monitor their expenses by categorizing them and keeping a log of them. The app should allow users to access their financial data from multiple devices. The app should ensure the security and privacy of user data through encryption, two-factor authentication.

8. APPENDIX

A. SOURCE CODE

```
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
  ) {

    val mContext = 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)
```

```
                val actualvalue = 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")
        }

    }
}
```