PROJECT REPORT

1. INTRODUCTION

    1.1 Overview

    1.2 Purpose

2. PROBLEM DEFINITION & THINKING

 2.1 Empathy Map

 2.2 Ideation & Brainstorming Map

3. RESULT

4. ADVANTAGES & DISADVANTAGES

5. APPLICATIONS

6. CONCLUSION

7. FUTURE SCOPE

8.APPENDIX

8.1 Source code

INTRODUCTION

Money matters are an essential part of our daily lives. From managing our personal finances to making business decisions, money plays a crucial role. It is the medium of exchange that

you are earning, spending, saving, or investing, understanding money matters is essential to achieving financial stability and success. In today's world, financial literacy has become more critical than ever before, as individuals and organizations face complex financial challenges and opportunities.

1.2 Purpose:

Money matters are an essential part of our daily lives. From managing our personal finances to making business decisions, money plays a crucial role. It is the medium of exchange that enables us to buy goods and services, invest in assets, and pay for our expenses. Whether you are earning, spending, saving, or investing, understanding money matters is essential to achieving financial stability and success. In today's world, financial literacy has become more critical than ever before, as individuals and organizations face complex financial challenges and opportunities.

PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy map:
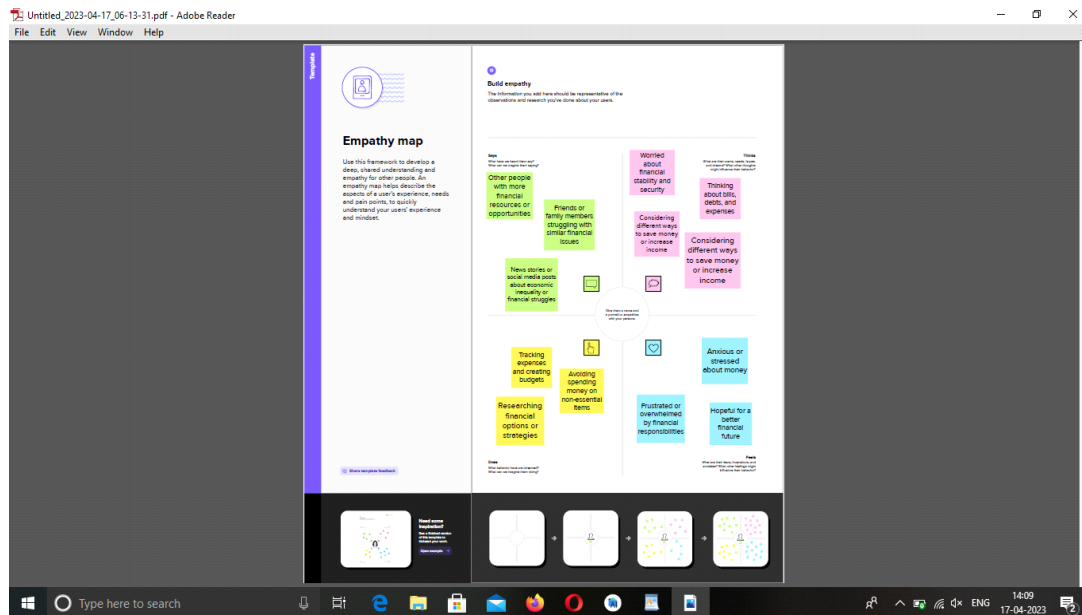
2.2 Ideation & Brainstorming Map:
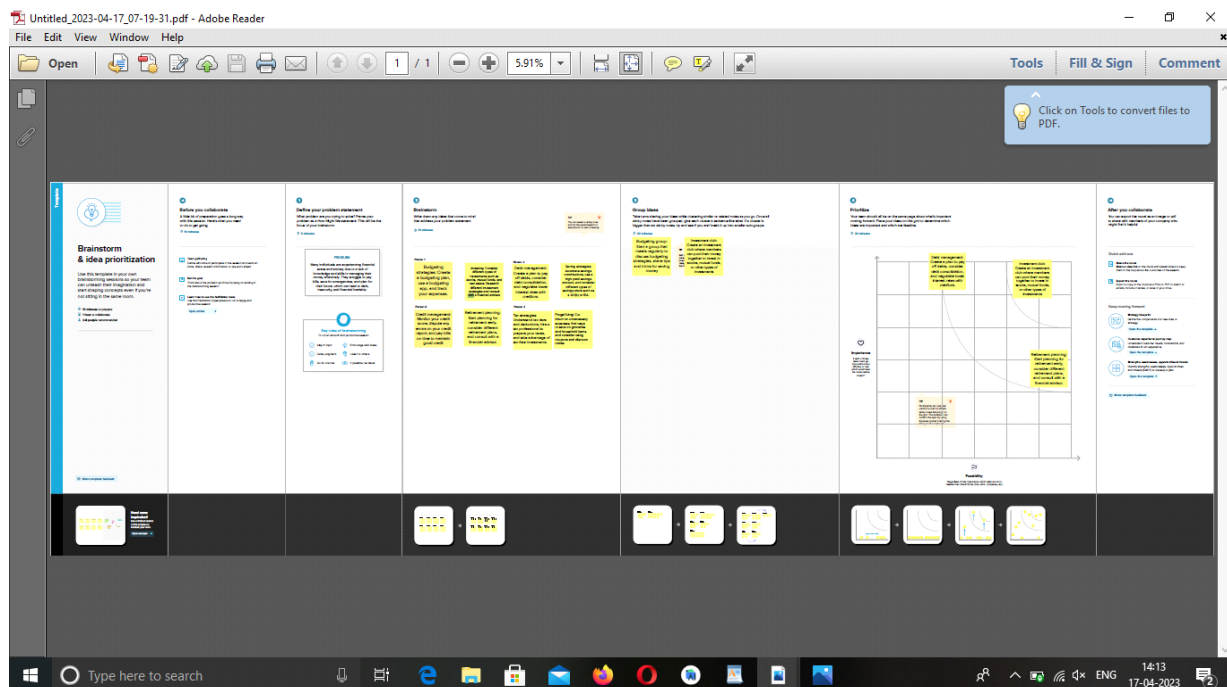
2.RESULT

Register page:

Register

Username
ramana

Email
ramana@gmail.com

Password
••••••••

Register

Login Page:

Register

Username
ramana

Email
ramana@gmail.com

Password
••••••••

Register

Home Page:

Welcome To Expense Tracker

Place Info Page:

Item Name

Item Name
chair

Quantity of item

Quantity
5

Cost of the item

Cost
4000

Submit

Advantages:

*Facilitates transactions: Money makes it easy to buy and sell goods and services, as it provides a universally accepted means of exchange. Without money, we would need to resort to barter, which would be cumbersome and inefficient.

*Enables economic growth: Money facilitates investment and capital accumulation, which, in turn, drives economic growth. With money, people can invest in businesses, buy property, and engage in other economic activities that generate wealth and create jobs.

*Promotes efficiency: Money incentivizes people to work harder and be more productive. It allows individuals and businesses to specialize in what they do best, which promotes efficiency and innovation.

*Promotes efficiency: Money incentivizes people to work harder and be more productive. It allows individuals and businesses to specialize in what they do best, which promotes efficiency and innovation.

4.2 Disadvantages

    1) Can lead to inequality: Money can exacerbate inequality as those who have more money tend to have more opportunities and power than those who have less. This can create a wealth gap, with some people experiencing poverty and others enjoying extreme wealth.

    2)Can lead to greed: Money can also lead to greed and unethical behavior as some people become obsessed with accumulating more wealth at any cost. This can result in a focus on profit over people, leading to unethical business practices and economic exploitation.

    3)Can cause financial stress: While money can provide financial security, it can also cause significant stress and anxiety. People may worry about losing their job, paying bills, or being unable to provide for their families, leading to financial stress that can negatively impact their mental and physical health.

5 APPLICATION

1)Personal finance: Managing personal finances is a critical application of money matters. It involves budgeting, saving, investing, and managing debt to achieve financial security and stability.

2)Business finance: Money is essential for running a business, and business finance involves managing cash flow, raising capital, investing in growth opportunities, and managing financial risks.

3)Investment: Investing is an application of money matters that involves using money to buy assets such as stocks, bonds, and real estate with the aim of generating returns.

6.CONCLUSION

* In conclusion, money matters are an essential part of modern society and impact various aspects of our lives. Money provides a means of exchange, promotes economic growth, facilitates investment, and provides financial security. However, money can also lead to inequality, greed, financial stress, illegal activities, and conflict.

7.FUTURE SCOPE

1)Digital currencies: Digital currencies, such as Bitcoin and other cryptocurrencies, are gaining popularity as an alternative to traditional forms of currency. As technology improves and more people adopt digital currencies, they could become an important part of the global financial system.

2)Financial technology: Financial technology, or fintech, is transforming the way we use and manage money. Fintech innovations, such as mobile banking, peer-to-peer lending, and robo-advising, are making financial services more accessible, convenient, and affordable.

3)Financial regulation: As new financial products and services emerge, regulators will need to adapt and develop new regulations to ensure consumer protection, financial stability, and fair competition.

8.APPENDIX

Source Code:

Main Activity:

```
package com.example.expensestracker

import android.annotation.SuppressLint

import android.content.Intent
```

```kotlin
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.
```

```kotlin
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)

)

{
```

```kotlin
                    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))


    }

}
```

Login Activity:

```
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
```

```kotlin
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,
```

```kotlin
    )

    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)
```

```kotlin
    )

    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(
```

```kotlin
                        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 = {
```

```kotlin
        })


        {

            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)

}
```

Register Activity:

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

```kotlin
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)

                }

            }

        }
```

```kotlin
    }
}


@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,
```

```kotlin
        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") },
```

```kotlin
        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
```

```kotlin
            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

            )
```

```kotlin
                    )
                })


                {
                    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)

    }
```