

**HX8001-PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY  
AND ENTREPRENEURSHIP**

# **PERSONAL EXPENSE TRACKER APPLICATION**

## **A PROJECT REPORT**

Submitted By

**TEAM ID : PNT2022TMID50153**

D.PRIYADHARSHINI

B.SNEKARANI

M.SOWMIYA

N.SUJITHA

*in partial fulfillment for the award of the degree*

Of

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

**JAYARAJ ANNAPACKIAM CSI COLLEGE OF ENGINEERING**

**NAZARETH**

**ANNA UNIVERSITY: CHENNAI 600 025**

**NOV - DEC 2022**

## **ABSTRACT**

simple words, personal finance entails all the financial decisions and activities that a Finance app makes your life easier by helping you to manage your finances efficiently. A personal finance app will not only help you with budgeting and accounting but also give you helpful insights about money management.

Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updated which will be visible to the user. Also, users can get an analysis of their expenditure in graphical forms. They have an option to set a limit for the amount to be used for that particular month if the limit is exceeded the user will be notified with an email alert.

# TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	2
1	<b>INTRODUCTION</b>	6
	1.1 Project Overview	7
	1.2 Purpose	7
2	<b>LITERATURE SURVEY</b>	8
	2.1 Existing problem	8
	2.2 References	9
	2.3 Problem Statement Definition	9
3	<b>IDEATION &amp; PROPOSED SOLUTION</b>	10
	3.1 Empathy Map Canvas	10

	3.2 Ideation & Brainstorming	11
	3.3 Proposed Solution	12
	3.4 Problem Solution fit	13
4	<b>REQUIREMENT ANALYSIS</b>	14
	4.1 Functional requirement	14
	4.2 Non-Functional requirements	14
5	<b>PROJECT DESIGN</b>	15
	5.1 Data Flow Diagrams	15
	5.2 Solution & Technical Architecture	16
	5.3 User Stories	17
6	<b>PROJECT PLANNING &amp; SCHEDULING</b>	17
	6.1 Sprint Planning & Estimation	17
	6.2 Sprint Delivery Schedule	18
7	<b>CODING &amp; SOLUTIONING</b>	18
	7.1 Feature 1	18

	7.2 Feature 2	19
8	<b>TESTING</b>	25
	8.1 Test Cases	25
	8.2 User Acceptance Testing	25
9	<b>RESULTS</b>	27
	9.1 Performance Metrics	27
10	<b>ADVANTAGES &amp; DISADVANTAGES</b>	29
11	<b>CONCLUSION</b>	31
12	<b>FUTURE SCOPE</b>	32
13	<b>APPENDIX</b>	33

**Source Code**

**GitHub & Project Demo Link**

# **CHAPTER 1**

## **INTRODUCTION**

Now a day's people are concerned about regularity of their daily expenses. This is done mainly for keep a track of the users' daily expenses to have a control of users' monthly expenses. We have developed an android application named as "Expense Tracker Application" and this application is used to manage the user's daily expenses in a more coherent and manageable way .

This application will help us to reduces the manual calculations for their daily expenses and also keep the track of the expenses. With the help of this application, user can calculate his total expenses per day and these results will stored for unique user. As the traditional methods of budgeting, we need to maintain the Excel sheets, Word Documents, notes, and files for the user daily and monthly expenses. There is no as such full-fledged solution to keep a track of our daily expenses easily. Keeping a log in diary is a very monotonous process and also may sometimes lead into problems due to the manual calculations. Looking on all the above given conditions, we are trying to satisfy the user requirements by building a mobile application which will help them reduces their burdens. "Expense Tracker Application" is an application where one can enter their daily expenses and end of the day, they know their expenses in charts.

## 1.1 Project Overview

As the name itself suggests, this project is an attempt to manage our daily expenses in an inefficient and manageable way. Sometime we can't remember where our money goes. And we can't handle our cash flow. For this problem, we need a solution that everyone can manage their expenses. So we decided to find an easier way to get rid of this problem. So, our application attempts to free the user with as much as possible the burden of manual calculation and to keep the track of the expenditure. Instead of keeping a diary or a log of the expenses, this application enables the user to not just keep the control on the expenses but also to generate and save reports. With the help of this application, the user can manage their expenses on a daily, weekly and monthly basis. Users can insert and delete transactions as well as can generate and save their reports. The graphical representation of the application is the main part of the system as it appeals to the user more and is easy to understand.

## 1.2 Purpose

To reduce manual calculations, we propose an application. This application allows users to maintain a digital automated diary. Each user will be required to register on the system at registration time, the user will be provided id, which will be used to maintain the record of each unique user. Expense Tracker application which will keep a track of Income-Expense of a user on a day to day basis. The best organizations have a way of tracking and handling these reimbursements. This ideal practice guarantees that the expenses tracked are accurately and in a timely manner. From a company perspective, timely settlements of these expenses when tracked well will certainly boost employees' morale. Additional feature of Expense and income prediction helps to better bud

## **CHAPTER-2**

### **LITERATURE SURVEY**

#### **2.1 Existing Problem**

The Expense tracker existing system does not provide the user portable device management level, existing system only used on desktop software so unable to update anywhere expenses done and unable to update the location of the expense details disruptive that the proposed system provides . In existing, we need to maintain the Excel sheets, CSV files for the user daily, weekly and monthly expenses. In existing, there is no as such complete solution to keep a track of its daily expenses easily. To do so a person as to keep a log in a diary or in a computer system, also all the calculations need to be done by the user which may sometimes results in mistakes leading to losses. The existing system is not user friendly because data is not maintained perfectly. But this project will not have any reminder to remain a person in a specific date, so that is the only drawback in which the remainder is not present. This project will be an unpopulated information because it has some disadvantages by not remind a person for each and every month. But it can used to perform calculation on income and expenses to overcome this problem we propose the new project.



## 2.2 Reference

1. <https://nevonprojects.com/daily-expense-tracker-system/>
2. <https://data-flair.training/blogs/expense-tracker-python/>
3. <https://phpgurukul.com/daily-expense-tracker-using-php-and-mysql/>
4. <https://ijarsct.co.in/Paper391.pdf>
5. [https://kandi.openweaver.com/?landingpage=python\\_all\\_projects&utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=promo\\_kandi\\_ie&utm\\_content=kandi\\_ie\\_search&utm\\_term=python\\_devs&gclid=Cj0KCQiAgribBhDkARIsAASA5bukrZgbI9UZxzpoyf0P-ofB1mZNxzc-okUP-3TchpYMclHTYFYiqP8aAmmwEALw\\_wcB](https://kandi.openweaver.com/?landingpage=python_all_projects&utm_source=google&utm_medium=cpc&utm_campaign=promo_kandi_ie&utm_content=kandi_ie_search&utm_term=python_devs&gclid=Cj0KCQiAgribBhDkARIsAASA5bukrZgbI9UZxzpoyf0P-ofB1mZNxzc-okUP-3TchpYMclHTYFYiqP8aAmmwEALw_wcB)

## 2.3 Problem Statement Definition

As for the time being, there a lot of budget planner software that are available online but some of these software fall short in helping users to actually create and stick to a budget. One of the drawbacks is the on-going maintenance, a lot of budget software offer the simplicity of integrating with all users financial accounts and consolidating their activity into one dashboard. However though, some of this existing software mostly have complicated features that are not user friendly. Also, due to the busy and hectic lifestyle people tend to overlook their budget and end up spending an excessive amount of money since they usually didn't plan their budget wisely. Last but not least, user cannot predict future expenses. While they can write down their expenses in a piece of paper or manage them in excel spreadsheet, their lack of knowledge in managing finances will be a problem

## CHAPTER 3

### IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map canvas



<https://app.mural.co/invitation/mural/personalexpensetrackerapplic1646/1664031878661?sender=uf53a97c045742871db980792&key=6eb7ca-a8a2-4b3c-adb5-3b2305614d6b>

## 3.2 Ideation & Brainstorming

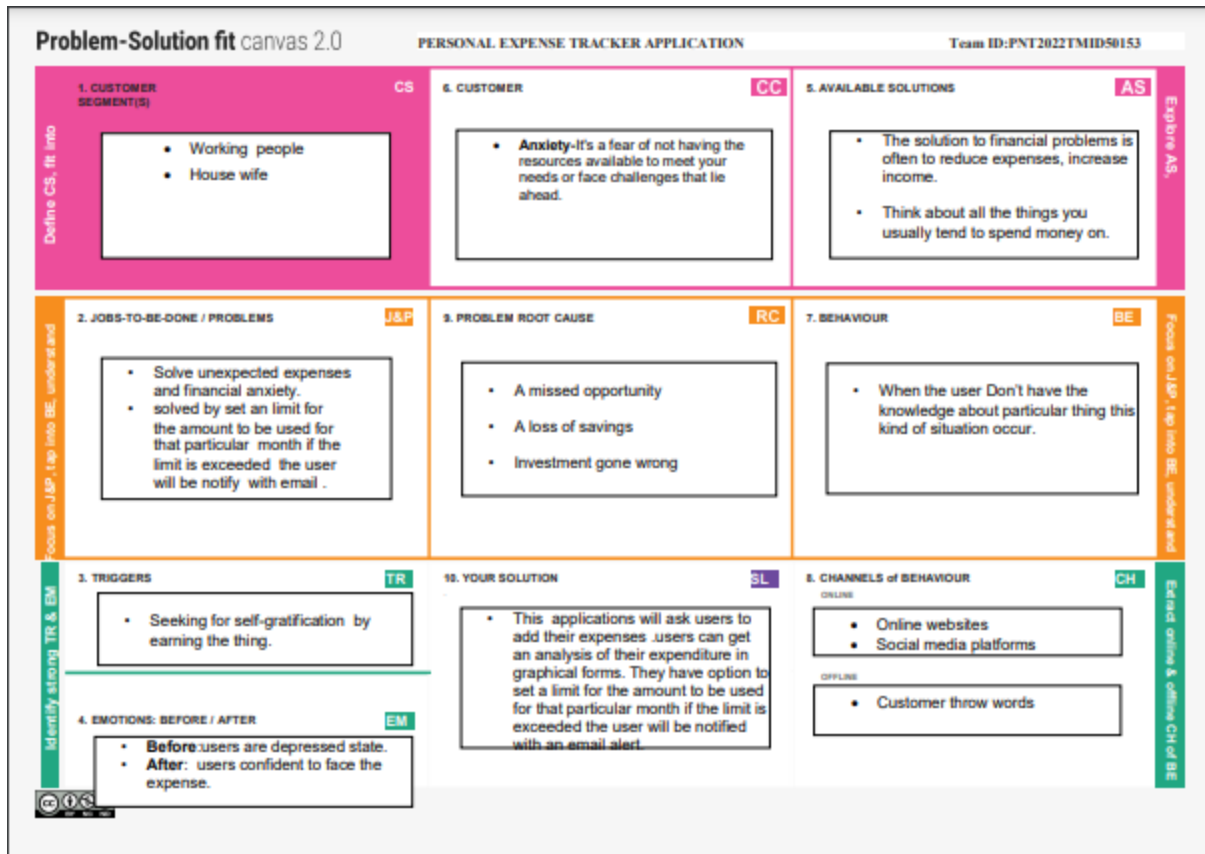


<https://app.mural.co/invitation/mural/personalexpensetrackeraplic2201/1664950020431?sender=u267c5556b03bf34dd8ef0643&key=78f156f0-a93f-49f0-a9fd-e26ce4acb6d4>

### 3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Can't receive notifications properly. The transaction have not been updated properly.
2.	Idea / Solution description	Personal finance applications will ask users to add their expenses and based on their expenses balance will be updated which will be visible to the user. Also, users can get an analysis of their expenditure in graphical forms. They have an option to set a limit for the amount to be used for that particular month if the limit is exceeded the user will be notified with an email alert.
3.	Novelty / Uniqueness	Notification can be receive through email.
4.	Social Impact / Customer Satisfaction	Personal Expense Tracker apps and design to be more user friendly and to enhance customer experience so that more people can transact and fulfill their essential banking needs through mobile app. This will make a impact on Mobile Banking for Customers' Satisfaction.
5.	Business Model (Revenue Model)	The Personal Expense Tracker app is a business model .but, Each people need to use the finance app to gain the benefit.
6.	Scalability of the Solution	Yes, it is scalable solution. we will develop project to get a better user experience.

### 3.4 Proposed Solution Fit



## **CHAPTER 4**

### **REQUIREMENT ANALYSIS**

#### **4.1 Functional requirement**

<b>FR No.</b>	<b>Functional Requirement (Epic)</b>	<b>Sub Requirement (Story / Sub-Task)</b>
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email
FR-3	Calender	Personal expense tracker application shall allow user to add the data to their expenses.
FR-4	Expense planner	This application should graphically represent the daily expenses and users own budget share.
FR-5	Expense Tracker	This application should graphically represent the expense in the form of report.
FR-6	Category	This application shall allow users to add categories of their expenses.

#### **4.2 Non Functional requirement**

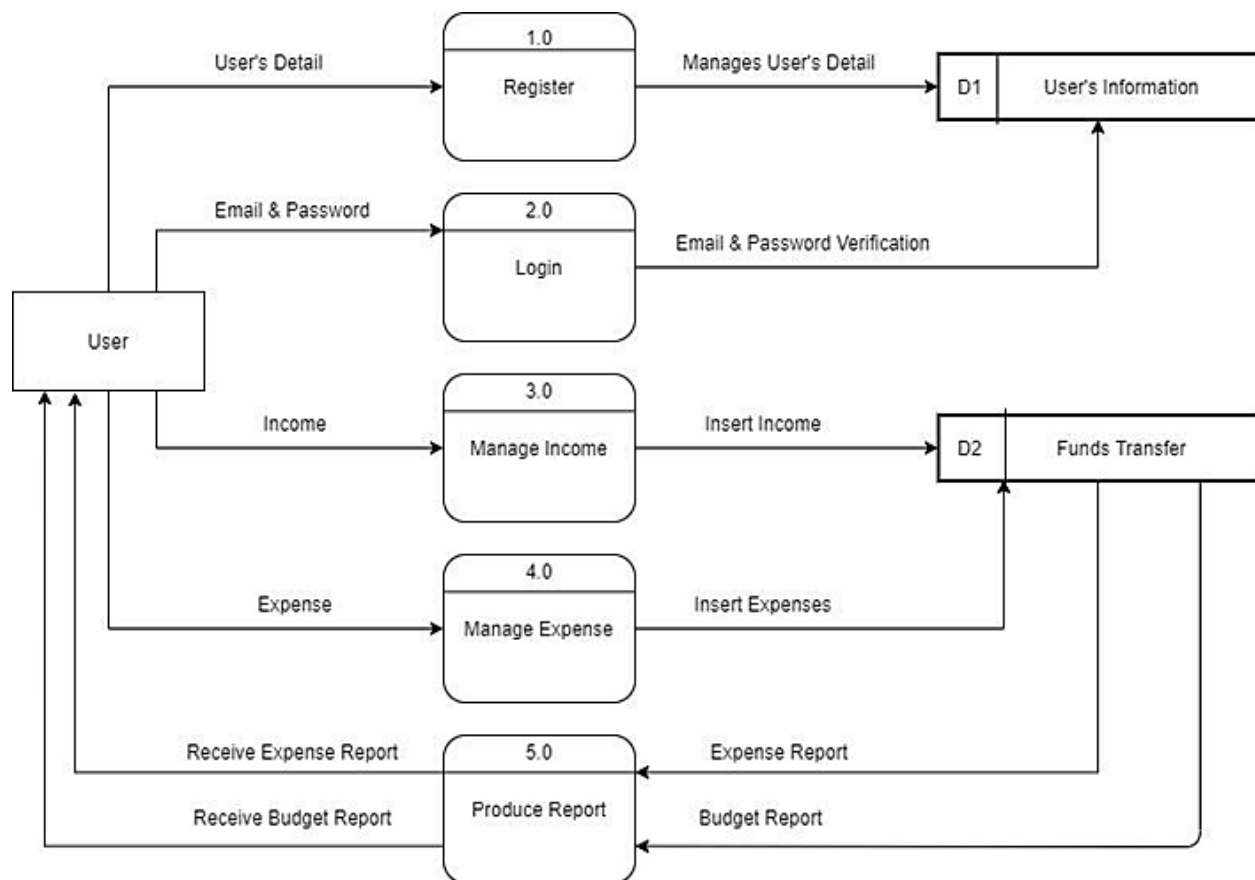
<b>FR No.</b>	<b>Non-Functional Requirement</b>	<b>Description</b>
NFR-1	<b>Usability</b>	There is a consistency in all the modules. To ease the navigation there is a back tab to provide access to previous page.
NFR-2	<b>Security</b>	More security of the customer data and bank account details.
NFR-3	<b>Reliability</b>	Each data record is stored on a well built efficient database schema. There is no risk of data loss.
NFR-4	<b>Performance</b>	The types of expense are categories along with an option. Throughput of the system is increased due to light weight database support.
NFR-5	<b>Availability</b>	It is available all the time, no time constraint.
NFR-6	<b>Scalability</b>	The ability to appropriately handle increasing demands.

## CHAPTER 5

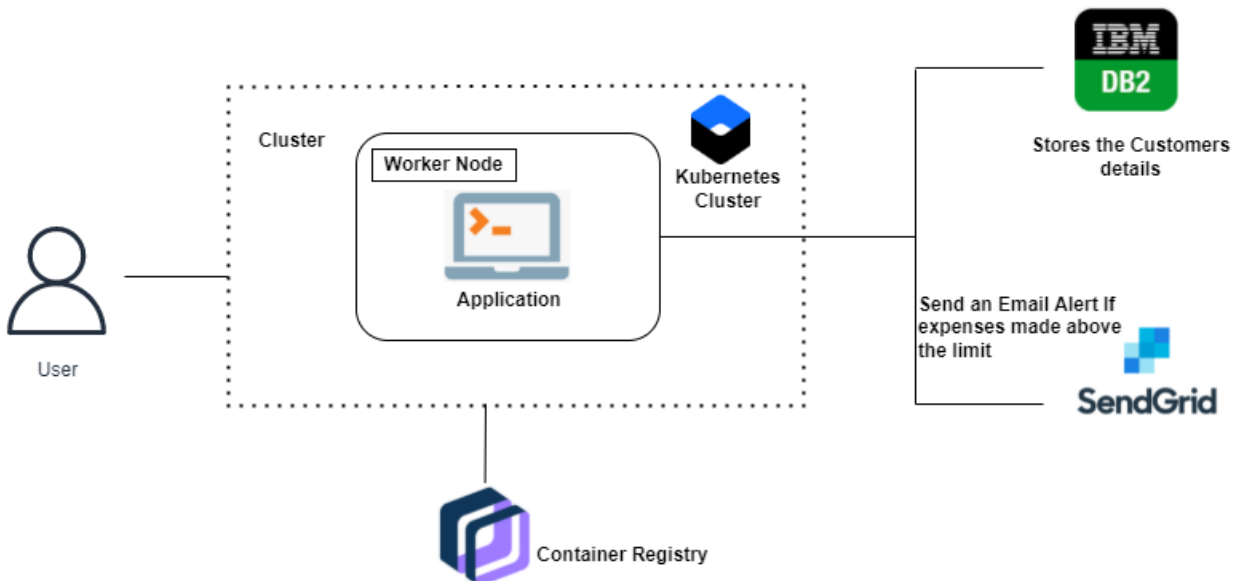
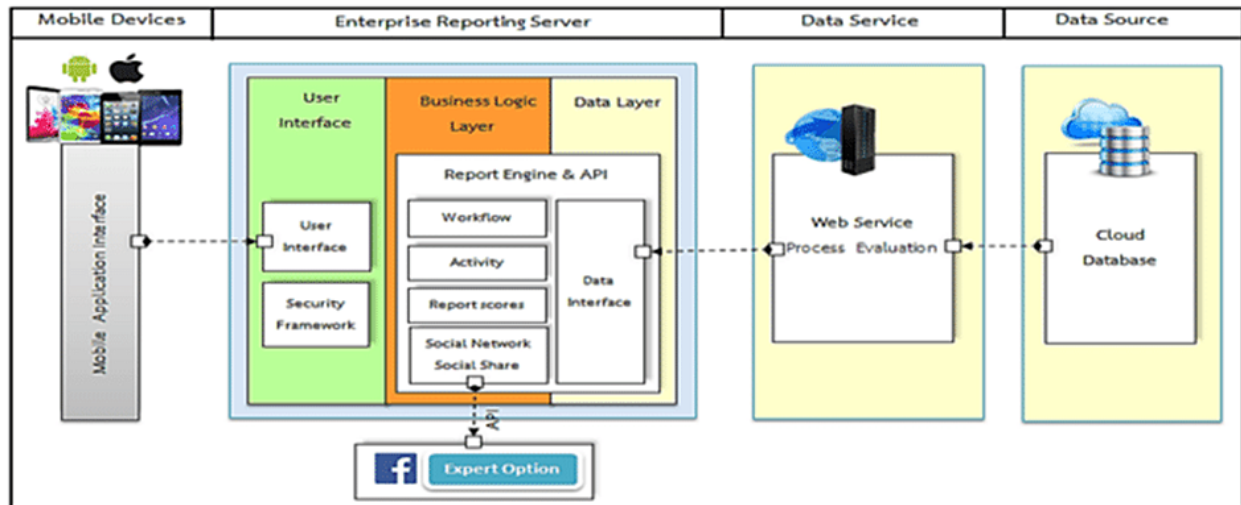
### PROJECT DESIGN

#### 5.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored..



## 5.2 Solution & Technical Architecture





## 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by bank account/e wallets.	I can access my account/ dashboard	High	Sprint-1
		USN-2	As a user, I set up a scheduled transactions	Updated the empty flow graph.	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	Not receive acceptance criteria.	Low	Sprint-2
	Login					
		USN-5	As a user, I can log into the application by entering email & password	Not receive acceptance criteria.	low	Sprint-2
	Dashboard	USN-6	As a user, I can register for the application by Bank account.	I can access my account/ dashboard	middle	Sprint-3
Customer (Web user)	Dashboard	USN-7	As a user, I can register for the application by Bank account.	I can access my account/ dashboard	middle	Sprint-3
	Registration	USN-8	As a user, I can register for the web application by entering my email ,password and conforming my password.	I can access my account/ dashboard	High	Sprint-1
Customer Care Executive	Customer support	USN - 9	As a user, I can support for customers to automatic logging of recurring transactions.	I can access my account/daskboard	High	Sprint-1
Administrative	Responsibility	USN -10	As a system administrator, track the user expenses anytime.	I can track expense	High	Sprint-1

## CHAPTER 6

### PROJECT PLANNING & SCHEDULING

#### 6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	D.Priyadharshini
		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	B.Snekarani
		USN-3	As a user, I can register for the application through the gmail	1	Medium	M.Sowmiya
	Login	USN-4	As a user, I can log into the application by entering email & password	1	High	N.Sujitha
	Dashboard	USN-5	Logging in takes to the dashboard for the logged user.	2	High	D.Priyadharshini
<i>Bug fixes, routine checks and improvisation by everyone in the team *Intended bugs only</i>						
Sprint 2	Workspace	USN-1	Workspace for personal expense tracking	2	High	D.Priyadharshini
	Charts	USN-2	Creating various graphs and statistics of customer's data	1	Medium	B.Snekarani
	Connecting to IBM DB2	USN-3	Linking database with dashboard	2	High	N.Sujitha
		USN-4	Making dashboard interactive with JS	2	High	M.Sowmiya

Sprint-3		USN-1	Wrapping up the server side works of frontend	1	Medium	D.Priyadharshini
	Watson Assistant	USN-2	Creating Chatbot for expense tracking and for clarifying user's query	1	Medium	B.Snekarani
	SendGrid	USN-3	Using SendGrid to send mail to the user about their expenses	1	Low	N.Sujitha
		USN-4	Integrating both frontend and backend	2	High	M.Sowmiya
<i>Bug fixes, routine checks and improvisation by everyone in the team *Intended bugs only</i>						
Sprint-4	Docker	USN-1	Creating image of website using docker/	2	High	D.Priyadharshini
	Cloud Registry	USN-2	Uploading docker image to IBM Cloud registry	2	High	M.Sowmiya
	Kubernetes	USN-3	Create container using the docker image and hosting the site	2	High	N.Sujitha
	Exposing	USN-4	Exposing IP/Ports for the site	2	High	B.Snekarani

## 6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	26 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	02 Nov 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	09 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	16 Nov 2022	19 Nov 2022	20	19 Nov 2022

## CHAPTER 7

### Coding And Solutioning

#### 7.1. Feature 1

**Feature (i):** Add Expense

**Feature (ii):** Update expense

**Feature (iii):** Delete Expense

**Feature (iv):** Set Limit

**Feature (v):** Send Alert Emails to users

## 7.2. Feature 2

Track your expenses anywhere, anytime. Seamlessly manage your money and budget without any financial paperwork. Just click and submit your invoices and expenditures. Access, submit, and approve invoices irrespective of time and location. Avoid data loss by scanning your tickets and bills and saving in the app. Approval of bills and expenditures in real-time and get notified instantly.

## CODING ANG SOLUTIONING

```
from flask import Flask, redirect, url_for, render_template, request, flash, session
import ibm_db
import re
```

```
app = Flask(__name__)
```

```
app.secret_key = 'a'
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECU
RITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gsk43831;PWD=6QVZyw
D7hzlVKjYy",,")
```

```
#HOME--PAGE
```

```
@app.route("/home")
```

```
def home():
```

```
    return render_template("homepage.html")
```

```
@app.route("/")
```

```
def add():
```

```
    return render_template("home.html")
```

```
#SIGN--UP--OR--REGISTER
```

```

@app.route("/signup")
def signup():
    return render_template("signup.html")

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = "

    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        stmt = ibm_db.prepare(conn,'SELECT * FROM register WHERE username = ?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'^[@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
        else:
            insert_sql = "INSERT INTO register VALUES (?, ?, ?)"
            stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email)
            ibm_db.bind_param(stmt, 3, password)
            ibm_db.execute(stmt)
            msg = 'You have successfully registered !'
            return render_template('signup.html', msg = msg)

```

#LOGIN-PAGE

```

@app.route("/signin")

```

```

def signin():
    return render_template("login.html")

@app.route('/login',methods =['GET', 'POST'])
def login():
    global userid
    msg = ""

if request.method == 'POST' :
    username = request.form['username']
    password = request.form['password']
    stmt = ibm_db.prepare(conn,'SELECT * FROM register WHERE username = ? AND
password = ?')
    ibm_db.bind_param(stmt,1,username)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print (account)
    if account:
        session['loggedin'] = True
        session['username'] = account['USERNAME']
        session['password'] = account['PASSWORD']
        msg = 'Logged in successfully !'
        return redirect('/home')
    else:
        msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

#ADDING DATA

@app.route("/add")
def adding():
    return render_template('add.html')

@app.route('/addexpense',methods=['GET', 'POST'])

```

```

def addexpense()
    date = request.form['date']
    expensename = request.form['expensename']
    amount = request.form['amount']
    paymode = request.form['paymode']
    category = request.form['category']

insert_sql = "(INSERT INTO expenses VALUES (NULL, % s, % s, % s, % s, % s, %
s),(session['id'],date, expensename, amount, paymode, category)"
    stmt = ibm_db.prepare( conn , insert_sql)
    ibm_db.bind_param(stmt, 1, session['id'])
    ibm_db.bind_param(stmt, 2, date)
    ibm_db.bind_param(stmt, 3, expensename)
    ibm_db.bind_param(stmt, 4, amount)
    ibm_db.bind_param(stmt, 5, paymode)
    ibm_db.bind_param(stmt, 6, category)
    ibm_db.execute(stmt)
    print(date + " " + expensename + " " + amount + " " + paymode + " " + category)
    return redirect("/display")

#DISPLAY graph

@app.route("/display")
def display():
    print(session["username"],session['id'])
    stmt = ibm_db.prepare(conn,'SELECT * FROM expenses WHERE userid = % s AND date
BY `expenses`.`date` DESC',(str(session['id'])))
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    return render_template('display.html' , expense = 'expense')

#delete---the--data

@app.route('/delete/<string:id>', methods = ['POST', 'GET' ])
def delete(id)

```

```

stmt = ibm_db.prepare(conn,'DELETE FROM expenses WHERE id = {0}'.format(id))
ibm_db.execute(stmt)
print('deleted successfully')
return redirect("/display")

```

#UPDATE---DATA

```

@app.route('/edit/<id>', methods = ['POST', 'GET' ])
def edit(id):
    stmt = ibm_db.prepare(conn,'SELECT * FROM expenses WHERE id = %s', (id,))
    row = ibm_db.fetch_tuple()
    print(row[0])
    return render_template('edit.html', expenses = row[0])
@app.route('/update/<id>', methods = ['POST'])
def update(id):
    if request.method == 'POST' :
        date = request.form['date']
        expensename = request.form['expensename']
        amount = request.form['amount']
        paymode = request.form['paymode']
        category = request.form['category']
        stmt = ibm_db.prepare(conn,"UPDATE `expenses` SET `date` = % s , `expensename` =%
s , `amount` = % s, `paymode` = % s, `category` = % s WHERE `expenses`.`id` = % s
",(date, expensename, amount, str(paymode), str(category),id))
        ibm_db.execute(stmt)
        print('successfully updated')
        return redirect("/display")

```

#limit

```

@app.route("/limit" )
def limit():
    return redirect('/limitn')

```

```

@app.route("/limitnum" , methods = ['POST' ])
def limitnum():
    if request.method == "POST":

```

```

number= request.form['number']

    stmt = ibm_db.prepare(conn,'INSERT INTO limits VALUES (NULL, % s, % s)
',(session['id'], number))
    ibm_db.execute(stmt)
    return redirect('/limitn')

@app.route("/limitn")
def limitn():
    stmt = ibm_db.prepare(conn,'SELECT limitss FROM `limits` ORDER BY `limits`.`id` DESC
LIMIT 1')
    x= ibm_db.fetch_assoc()
    s = x[0]

    return render_template("limit.html" , y= s)

#REPORT

#log-out
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return render_template("home.html")

if __name__ == " main ":
    app.run(debug=True)

```

**The other code features are submitted in Github:**

<https://github.com/IBM-EPBL/IBM-Project-22133-1659805737>



## **CHAPTER 8**

### **TESTING**

#### **\_8.1. TESTING:**

1. Login Page (Funcional)
2. Login Page (UI)
3. Add Expense Page (Functional)

#### **8.2. User Acceptance Testing:**

##### **1. Purpose of Document**

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

##### **2. Defect Analysis**

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	8	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	11	20
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	0	1	8

Totals	22	14	11	22	51
--------	----	----	----	----	----

### 3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

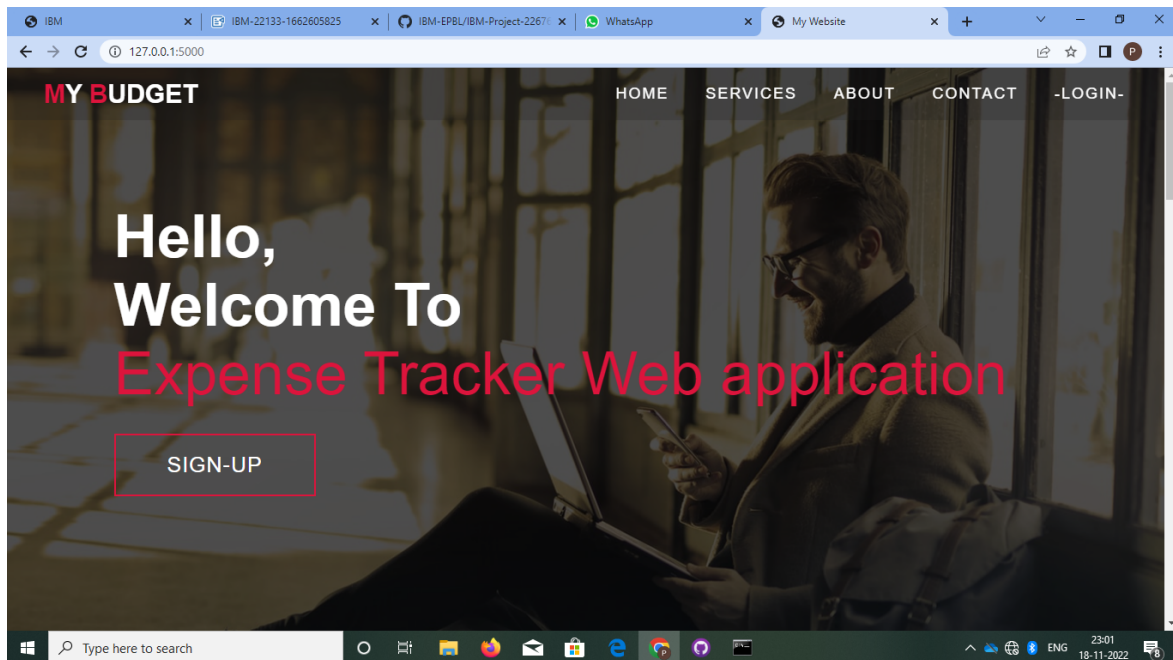
Section	Total Cases	Not Tested	Fail
Interface	7	0	0
Login	43	0	0
Logout	2	0	0

## CHAPTER 9

### RESULTS

#### 9.1 Performance Metrics

##### HomePage



## Signup page

IBM x IBM-22133-1662605825 x IBM-EPBL/IBM-Project-2267 x WhatsApp x Sign-up x +

127.0.0.1:5000/signup

Hello, Friend

Name

Example@gmail.com

Password

OR

Sign-up with

CREATE ACCOUNT

I read and agree to [Terms & Conditions](#)

Welcome, Please Fill in the blanks for sign up

23:01 18-11-2022

## Login Page

IBM x IBM-22133-1662605825 x IBM-EPBL/IBM-Project-2267 x WhatsApp x Animated Login Form x +

127.0.0.1:5000/signin

WELCOME

Username

Password

[Forgot Password?](#)

LOGIN

OR

Login with

Don't have an account? [REGISTER here](#)

23:01 18-11-2022

## **CHAPTER 10**

### **ADVANTAGES AND DISADVANTAGES**

#### **10.1. ADVANTAGES:**

- One of the major pros of tracking spending is always being aware of the state one's personal finances.
- Tracking what you spend can help you stick to your budget, not just in a general way, but in each category such as housing, food, transportation and gifts.
- While a con is that manually tracking all cash that is spent can be irritating as well as time consuming, a pro is that doing this automatically can be quick and simple.
- Another pro is that many automatic spending tracking software programs are available for free.
- Having the program on a hand-held device can be a main pro since it can be checked before spending occurs in order to be sure of the available budget.

## 10.2. DISADVANTAGES:

- A con with any system used to track spending is that one may start doing it then taper off until it's forgotten about all together.
- Yet, this is a risk for any new goal such as trying to lose weight or quit smoking.
- If a person first makes a budget plan, then places money in savings before spending any new pay period or month, the tracking goal can help.
- In this way, tracking spending and making sure all receipts are accounted for only needs to be done once or twice a month.
- Even with constant tracking of one's spending habits, there is no guarantee that financial goals will be met.
- Although this can be considered to be a con of tracking spending, it could be changed into a pro if one makes up his or her mind to keep trying to properly manage all finances.
- Another con that may occur when spending is being tracked is an error, but this may also be able to be changed into a pro if the person does regular tracking. Frequent tracking of cash spending can allow one to catch and correct errors so that the budget plan is still able to be adhered to despite the mistake.

## **CHAPTER 11**

### **CONCLUSION**

A comprehensive money management strategy requires clarity and conviction for decision making. You will need a defined goal and a clear vision for grasping the business and personal finances. That's when an expense tracking app comes into the picture. An expense tracking app is an exclusive suite of services for people who seek to handle their earnings and plan their expenses and savings efficiently. It helps you track all transactions like bills, refunds, payrolls, receipts, taxes, etc., on a daily, weekly, and monthly basis. Thus, we have developed such type of android application which help the users to reduces their effort of handling daily expenses. That the application will have various components of updating and viewing users' expenditures. As part of research, we considered adding certain components to the application to make it more useful to the user. Some of the extra Components are like enabling users to register to the application using existing email or social network account, it will synchronize the users profile data to the application

## **CHAPTER 12**

### **FUTURE SCOPE**

- Achieve your business goals with a tailored mobile app that perfectly fits your business.
- Scale-up at the pace your business is growing.
- Deliver an outstanding customer experience through additional control over the app.
- Control the security of your business and customer data.
- Open direct marketing channels with no extra costs with methods such as push notifications.
- Increase efficiency and customer satisfaction with an app aligned to their needs.
- Ability to provide valuable insights.
- Optimize sales processes to generate more revenue through enhanced data collection.
- Robo Advisors: Get expert investment advice and solutions with the Robo-advisors
- feature. This feature will analyze, monitor, optimize, and improve diversification in investments by turning data into actionable insights



in real-time. Chats: Equip your expense tracking app with a bot that can understand and answer all user queries and address their needs such as account balance, credit score, etc.

- Prediction: With the help of AI, your mobile app can predict your next purchase, according to your spending behavior. Moreover, it can recommend products and provide unique insights on saving money. It brings out the factors causing fluctuations in your expenses.
- Employee Travel Budgeting: Most businesses save money with a travel budgeting app as It helps prepare a budget for an employee's entire business trip. The feature will predict the expenses and allocate resources according to the prediction.

## **APPENDIX:**

**demo video link :**

<https://youtu.be/SuXzAAjbKkk>

**github link :**

<https://github.com/IBM-EPBL/IBM-Project-22133-1659805737>

