

# **PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)**

## **PERSONAL EXPENSE TRACKER**

### **A PROJECT REPORT**

*Submitted by*

**S RAHUL KUMAR (111719106122)**

**SHAAMA M (111719106140)**

**VENKATA GUNASEKHAR V (111719106171)**

**VINAY KRISHNA K N (111719106176)**

**TEAM ID : PNT2022TMID16140**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**R.M.K.ENGINEERING COLLEGE**

**(An Autonomous Institution)**

**R.S.M. Nagar, Kavaraipttai-601 206**

**NOVEMBER 2022**



# **PERSONAL EXPENSE TRACKER**

# INDEX

## **1. INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

## **2. LITERATURE SURVEY**

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

## **4. REQUIREMENT ANALYSIS**

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

## **5. PROJECT DESIGN**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

## **6. PROJECT PLANNING & SCHEDULING**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

## **7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database Schema (if Applicable)

## **8. TESTING**

- 8.1 Test Cases
- 8.2 User Acceptance Testing

## **9. RESULTS**

- 9.1 Performance Metrics

## **10. ADVANTAGES & DISADVANTAGES**

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

## **13. APPENDIX**

Source Code

GitHub & Project Demo Link

# 1. INTRODUCTION

## 1.1 Project Overview

Generally, people nowadays people don't keep track of their expenses and keep on spending beyond their income limits. The also take unwanted loans and have low credit scores due to unwanted expense which is made from their income.

The expense tracker application makes sure that the expenses are tracked category wise. It also provides monthly and daily (based graph analysis) of the expense done. Which will provide insights to the user.

The user will also be alerted via an email if the expense goes beyond a certain amount (the amount will be user set) with the help of SendGrid framework.

## 1.2 Purpose

Generally, people nowadays people don't keep track of their expenses and keep on spending beyond their income limits. The also take unwanted loans and have low credit scores due to unwanted expense which is made from their income.

The expense tracker application makes sure that the expenses are tracked category wise. It also provides monthly and daily (based graph analysis) of the expense done. Which will provide insights to the user.

The user will also be alerted via an email if the expense goes beyond a certain amount (the amount will be user set) with the help of SendGrid framework.

## THE REASONS FOR EXPENSE TRACKER

QUESTION	DESCRIPTION
<b>Who does the problem affect</b>	People who need to monitor their unwanted expenses
<b>What is the issue</b>	Unable to keep track of their own expenses
<b>When does the issue occur</b>	When user is unaware of their expenses
<b>Where is the issue occurring</b>	Due to diverse payment gateways such as UPI,Net banking etc it is highly impossible to monitor the expenses
<b>Why it is important that we fix the problem</b>	To avoid the unwanted expenses that had been made and rectify it in the future purchases

General purpose of the expense tracker 1.1

## 2.LITERATURE SURVEY

### 2.1 Existing problem

An App to manage your entire financial decision-making process by keeping track of it. By assisting you in effectively managing your funds, this software makes your life simpler. Personal Expense Tracker software will assist you with financial management, accounting, and budgeting in addition to budgeting and accounting.

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	An App to manage your entire financial decision-making process by keeping track of it. By assisting you in effectively managing your funds, this software makes your life simpler.  Personal Expense Tracker software will assist you with financial management, accounting, and budgeting in addition to budgeting and accounting.

2.	Idea / Solution description	<p>Personal finance includes all of the financial choices and tasks that finance software streamlines by assisting you in effectively managing your money. Personal Expense Tracker software will not only assist you with accounting and budgeting, but it will also provide you with valuable advice and control on money management.</p>
3.	Novelty / Uniqueness	<p>Display the costs on a monthly and weekly basis of expense in all in one app.</p>
4.	Social Impact / Customer Satisfaction	<p>People can use it to keep track of their spending and receive alerts when their budget is exceeded at times of low balance.</p>
5.	Business Model (Revenue Model)	<p>We can offer the additional program and features on a subscription basis if required.</p>
6.	Scalability of the Solution	<p>Future customers of IBM Cloud will automatically receive additional storage based on their usage and customer base.</p>

## 2.2 References

Title of the Paper			
eExpense: A Smart Approach to Track Everyday Expense			
Conference: IEEE			
Year:	2019	Author:	Shahed Anzarus,Sadman Saumik, Md.Jewel Rana,Monir Hossain.
Inferences	<ul style="list-style-type: none"><li>• By using this application, users can save their expense by simply scanning the bills or receipt copies.</li><li>• This application extracts the textual information from the receipts and saves the amount and description for further processing.</li><li>• Tracking your expenses daily can save your amount</li></ul>		

References 2.1



<b>Title of the Paper</b>			
<b>eExpense: A Smart Approach to Track Everyday Expense</b>			
<b>Conference: IEEE</b>			
<b>Year:</b>	2019	<b>Author:</b>	Shahed Anzarus,Sadman Saumik, Md.Jewel Rana,Monir Hossain.
<b>Inferences</b>	<ul style="list-style-type: none"> <li>● Tracking your expenses daily can save your amount.</li> <li>● The project successfully avoids the manual calculation for avoiding calculating the income and expense per month and save time of user. The modules are developed with efficient, reliable.</li> </ul>		

References 2.2

## References:

[1] Online Income and Expense Tracker S. Chandini<sup>1</sup>, T. Poojitha<sup>2</sup>, D. Ranjith<sup>3</sup>, V.J. Mohammed Akram<sup>4</sup>, M.S. Vani<sup>5</sup>, V. RajyalakshmiJ. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61

[2] Expense Tracker ATIYA KAZI<sup>1</sup> , PRAPHULLA S. KHERADE<sup>2</sup> , RAJ S. VILANKAR<sup>3</sup> , PARAG M. SAWANT<sup>4</sup> <sup>1</sup> Professor, Department of Information Technology, Finolex Academy of Management and Technology, Ratnagiri, Maharashtra, India. 2, 3, 4 Department of Information Technology, Finolex Academy of Management and Technology, Ratnagiri, Maharashtra, india

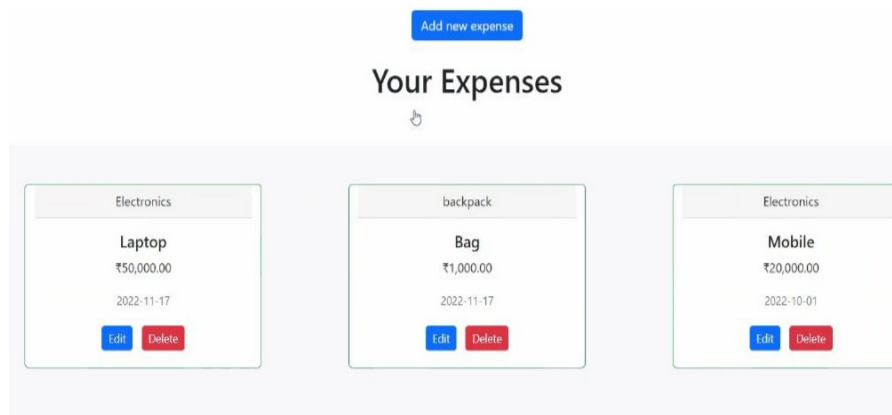
[3] Expense Tracker Application Velmurugan.R<sup>1</sup> , Mrs.P.Usha<sup>2</sup> <sup>1</sup>PG Student, Department of Computer Science, Dr.N.G.P Arts And Science College, Coimbatore-641048, Tamilnadu, India <sup>2</sup>Assistant Professor, Department of Computer Science, Dr.N.G.P Arts And Science College, Coimbatore641048, Tamil Nādu, India

[4]Expense Tracker Prof Miriam Thomas<sup>1</sup> , Lekshmi P<sup>2</sup> , and Dr. Mahalakshmi T<sup>3</sup> <sup>1</sup> Assistant Professor, Master of Computer Application<sup>1</sup> Student, Final Year Master of Computer Application<sup>2</sup> Principal, Sree Narayana Institute of Technology Master of Computer Application<sup>2</sup> Principal, Sree Narayana Institute of Technology<sup>3</sup> Sree Narayana Institute of Technology, Kollam, Kerala.

## 2.3 Problem Statement Definition

Expense Tracker At the instant, there is no as such complete solution present easily or we should say free of cost which enables a person to keep a track of its daily expenditure easily. To do so a person has to keep a log in a diary or in a computer, also all the calculations needs to be done by the user which may sometimes results in errors leading to losses. Due to lack of a complete tracking system, there is a constant overload to rely on the daily entry of the expenditure and total estimation till the end of the month. As the name itself suggests, this project is an attempt to manage our daily expenses in a more efficient and manageable way. The system attempts to free the user with as much as possible the burden of manual calculation and to keep the track of the expenditure.

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



personal expense tracker (demo)

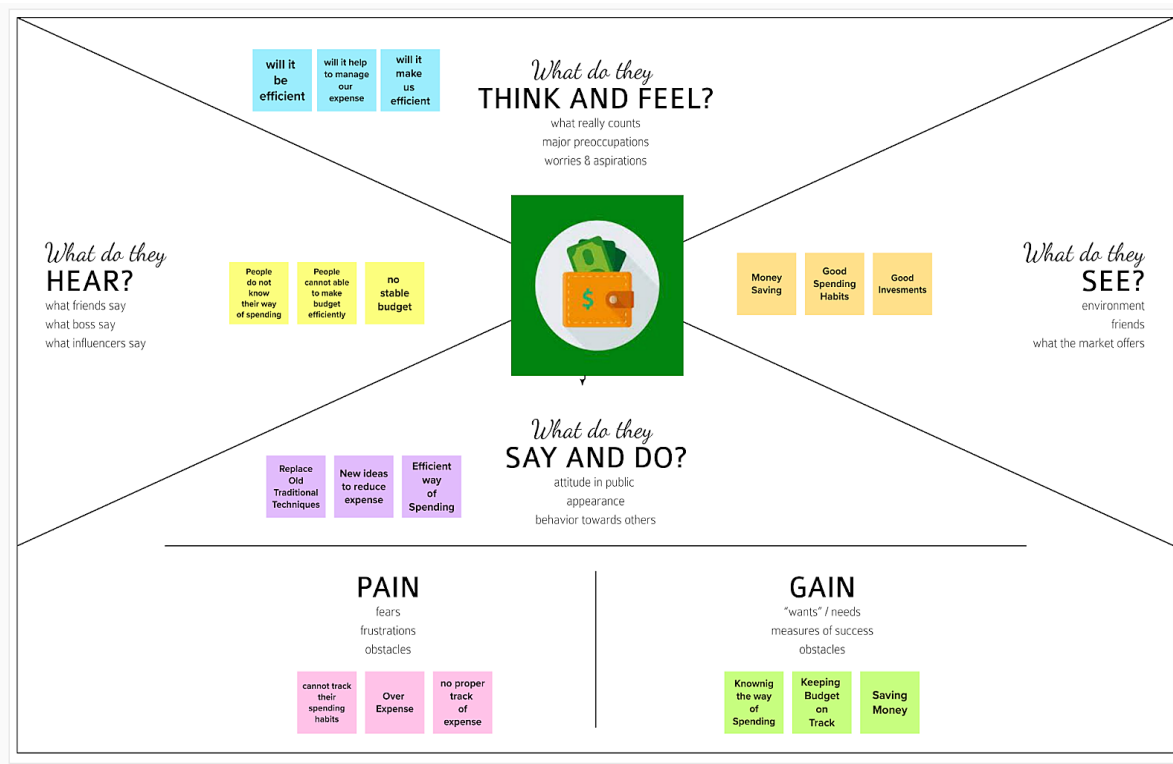
One of the drawbacks is the on-going maintenance, a lot of budget software offer the simplicity of integrating with all users financial accounts and consolidating their activity into one dashboard. However though, some of this existing software mostly have complicated features that are not user friendly.

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.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	An App to manage your entire financial decision-making process by keeping track of it. By assisting you in effectively managing your funds, this software makes your life simpler. Personal Expense Tracker software will assist you with financial management, accounting, and budgeting in addition to budgeting and accounting.
2.	Idea / Solution description	Personal finance includes all of the financial choices and tasks that finance software streamlines by assisting you in effectively managing your money. Personal Expense Tracker software will not only assist you with accounting and budgeting, but it will also provide you with valuable advice and control on money management.
3.	Novelty / Uniqueness	Display the costs on a monthly and weekly basis of expense in all in one app.
4.	Social Impact / Customer Satisfaction	People can use it to keep track of their spending and receive alerts when their budget is exceeded at times of low balance.

### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas




This empathy map give the rough idea about the project with we are focuses the problem we are there to solve ,is described as empathy map.  
The interface is described in the form of data flow or practical real time analysis .

### 3.2 Ideation & Brainstorming




This brainstorming and ideation phase, shows the innovation we have tried to implement it. We have implemented various libraries and frameworks in our project which we discuss later in our project.

Template




## Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

-  10 minutes to prepare
-  1 hour to collaborate
-  2-8 people recommended

Share template feedback



#### Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

## step 1:



### Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

 10 minutes

**A**

#### Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

**B**

#### Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

**C**

#### Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#)



## step1.1:

1

### Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

#### PROBLEM

How might we track our expense using this application?



### Key rules of brainstorming

To run an smooth and productive session



Stay in topic.



Encourage wild ideas.



Defer judgment.



Listen to others.



Go for volume.



If possible, be visual.



## step 2:

2

### Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

#### TIP



You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

#### Rahul Kumar

navigate to  
the  
dashboard

edit user  
profile

visulize the  
expense

set budjet

#### Shaama M

categorize  
the  
expenses

get report

feedback  
system

filter the  
expense

#### VenkataGunasekhar

shows cash  
flow

generate  
report

accurate  
records

edit  
income

#### Vinay Krishna

categorize  
the  
expense

advantages  
in budjet

add  
multiple  
expenses

remainder  
of expense

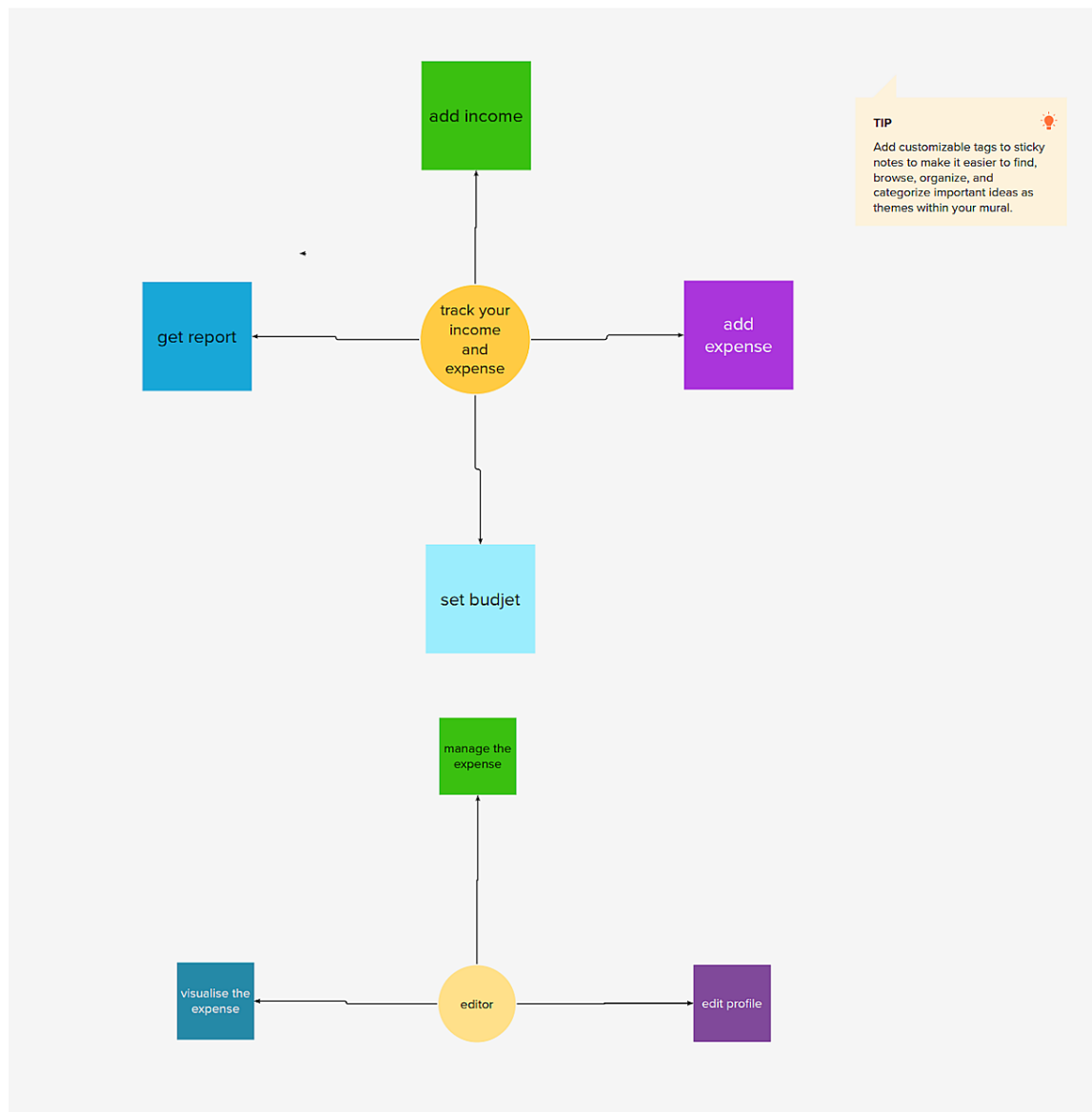
### step 3:

3

#### Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes



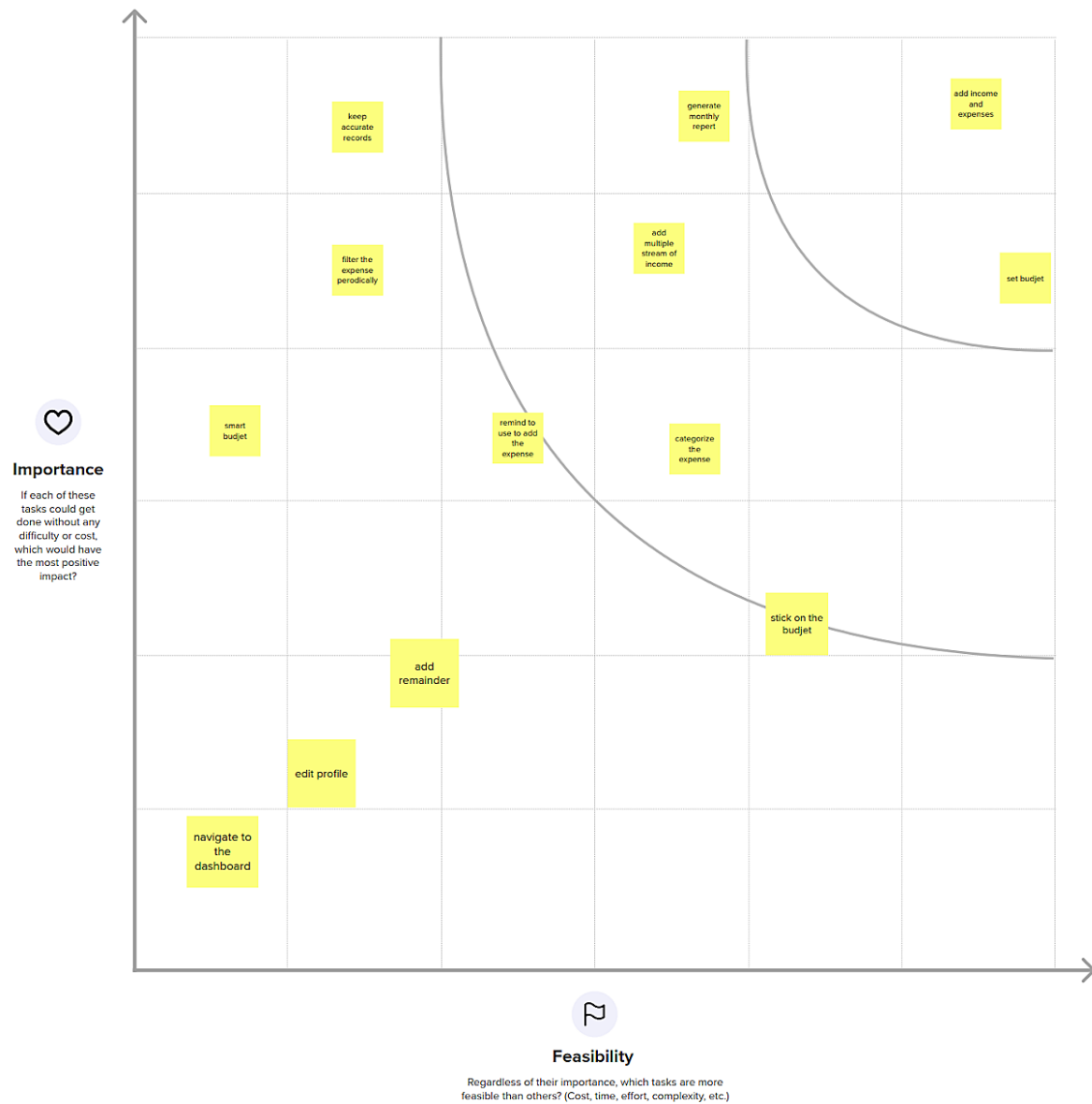
## step 4:

4

### Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

🕒 20 minutes



### 3.3 Proposed Solution

We have tabulated our proposed solution and the design approach of the project which we have developed separately.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	An App to manage your entire financial decision-making process by keeping track of it. By assisting you in effectively managing your funds, this software makes your life simpler. Personal Expense Tracker software will assist you with financial management, accounting, and budgeting in addition to budgeting and accounting.
2.	Idea / Solution description	Personal finance includes all of the financial choices and tasks that finance software streamlines by assisting you in effectively managing your money. Personal Expense Tracker software will not only assist you with accounting and budgeting, but it will also provide you with valuable advice and control on money management.
3.	Novelty / Uniqueness	Display the costs on a monthly and weekly basis of expense in all in one app.
4.	Social Impact / Customer Satisfaction	People can use it to keep track of their spending and receive alerts when their budget is exceeded at times of low balance.
5.	Business Model (Revenue Model)	We can offer the additional program and features on a subscription basis if required.
6.	Scalability of the Solution	Future customers of IBM Cloud will automatically receive additional storage based on their usage and customer base.

### 3.4 Problem Solution fit

we have made a custom fit problem solution approach to the personal expense tracker. Like for example we have allocated separate data base for each of the user who logs in the platform. We also make the separate login credentials for each user and authenticate it. We have discussed in detail about it.

Define CS, fit into CC	<div>1. CUSTOMER SEGMENT(S) <div>Who is your customer? I.e. working parents of 0-5 y.o. kids</div><div><ul style="list-style-type: none"><li>• Customers are those who spend money without keeping track of it or struggling to keep track of it</li><li>• Provides a whole lot of different categories of expenditure types to avoid mismatch of expenditure</li></ul></div></div>	<div>6. CUSTOMER CONSTRAINTS <div>What constraints prevent your customers from taking action or limit their choices of solutions? I.e. spending power, budget, no cash, network connection, available devices.</div><div><ul style="list-style-type: none"><li>• Most of the solution available in the internet hosts a lot of adds limiting its usability</li><li>• The solution proposed here has a feature to view the expense graphically</li></ul></div></div>	<div>5. AVAILABLE SOLUTIONS <div>Which solutions are available to the customers when they face the problem? or need to get the job done? What have they tried in the past? What pros &amp; cons do these solutions have? I.e. pen and paper is an alternative to digital notetaking</div><div><ul style="list-style-type: none"><li>• Expense tracker applications which are available in both android and ios.</li><li>• Personal Expense tracker developed in this project</li></ul></div></div>	Explore AS, differentiate
	<div>2. JOBS-TO-BE-DONE / PROBLEMS <div>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.</div><div><ul style="list-style-type: none"><li>• The objective of this application is to enable customers to keep track of their expenses.</li><li>• The customers are provided with categories for the expenses.</li></ul></div></div>	<div>9. PROBLEM ROOT CAUSE <div>What is the real reason that this problem exists? What is the backstory behind the need to do this job? I.e. customers have to do it because of the change in regulations.</div><div><ul style="list-style-type: none"><li>• Improper expenses lead to heavy tax.</li><li>• Makes business forecasting easier</li><li>• Saves a lot of money</li></ul></div></div>	<div>7. BEHAVIOUR <div>What does your customer do to address the problem and get the job done? I.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (I.e. Greenpeace)</div><div><ul style="list-style-type: none"><li>• Start using the expense tracker app</li><li>• Makes sure he categorize the expense done in order to save money</li></ul></div></div>	
Focus on J&P, tap into BE, understand RC				Focus on J&P, tap into BE, understand RC
Identify strong TR & EM	<div>3. TRIGGERS <div>What triggers customers to act? I.e. seeing their neighbor installing solar panels, reading about a more efficient solution in the news.</div><div><ul style="list-style-type: none"><li>• Understanding the fact the customers can save a lot of money by these expense apps</li></ul></div></div>	<div>10. YOUR SOLUTION <div>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior</div><div><ul style="list-style-type: none"><li>• Design a flask based personal expense tracker application</li><li>• Enable email based expense alerts using sendgrid framework.</li></ul></div></div>	<div>8. CHANNELS of BEHAVIOR <div>8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7</div><div><ul style="list-style-type: none"><li>• Expense trackers online come with a lot of ads which on clicking steals data like account number if provided</li></ul></div><div>8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.</div><div><ul style="list-style-type: none"><li>• Make sure they are aware of the tax rules by reading the available books to make them tax read</li></ul></div></div>	Identify strong TR & EM
	<div>4. EMOTIONS: BEFORE / AFTER <div>How do customers feel when they face a problem or a job and afterwards? I.e. lost, insecure &gt; confident, in control - use it in your communication strategy &amp; design.</div><div><ul style="list-style-type: none"><li>• They feel a lot clear about the income and expenses made</li></ul></div></div>			

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

The technical aspect of the project is tabulated as the functional requirement.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email  Confirmation via OTP
FR-3	User monthly expense tentative data	Data to be registered in the app
FR-4	User monthly income data	Data to be registered in the app
FR-5	Alert/ Notification	Alert through E-mail Alert through SMS
FR-6	User Budget Plan	Planning and Tracking of user expense vs budget limit

#### 4.2 Non-Functional requirements

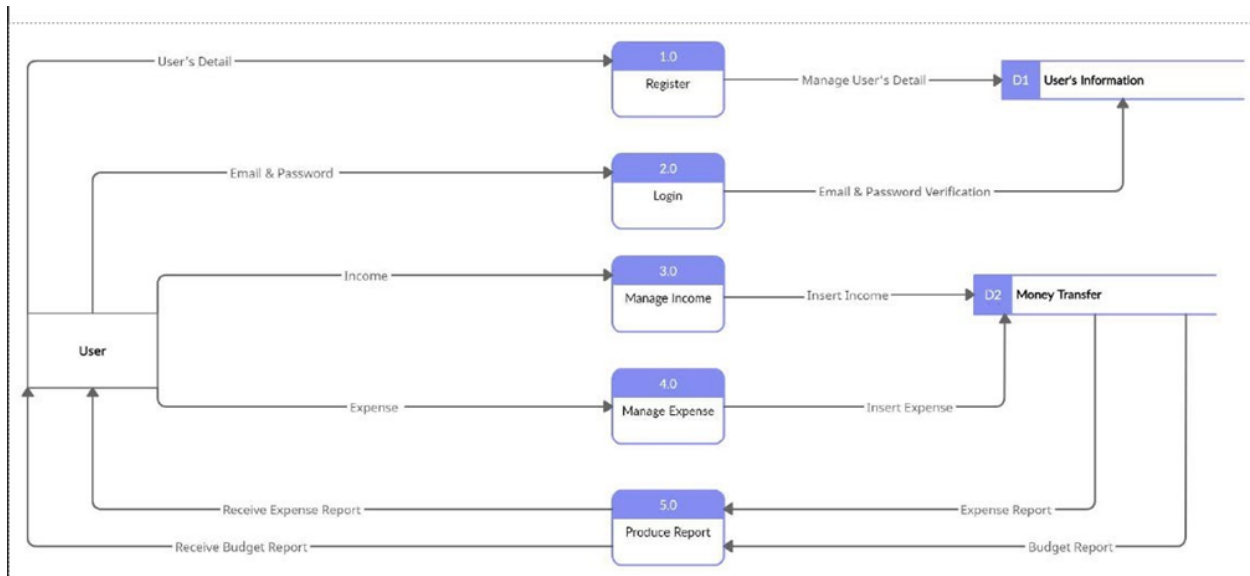
The non technical aspect of the project is tabulated here.

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	Effectiveness, efficiency and overall satisfaction of the user while interacting with our application.
NFR-2	<b>Security</b>	Authentication, authorization, encryption of the application.
NFR-3	<b>Reliability</b>	Application is more reliable.
NFR-4	<b>Performance</b>	how the application is functioning and how responsive the application is to the end-users.
NFR-5	<b>Availability</b>	Without near 100% availability, application reliability and the user satisfaction will affect the solution.
NFR-6	<b>Scalability</b>	Capacity of the application to handle growth, especially in handling more users.

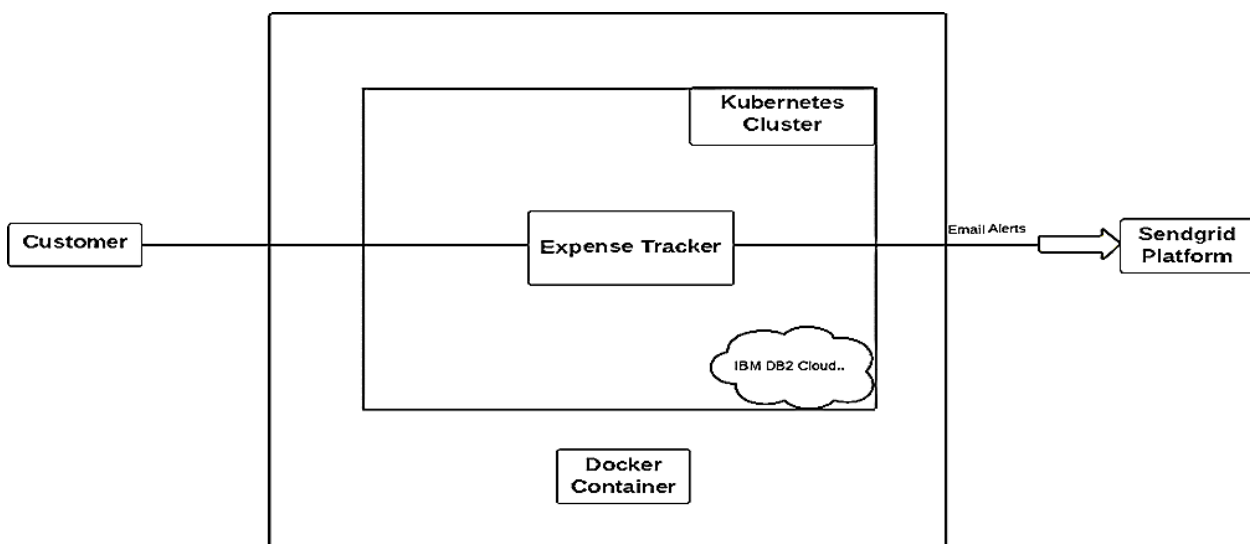
## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams

The data flow diagram depicts the flow of the project (personal expense tracker). This is give the easy flow understanding of the project

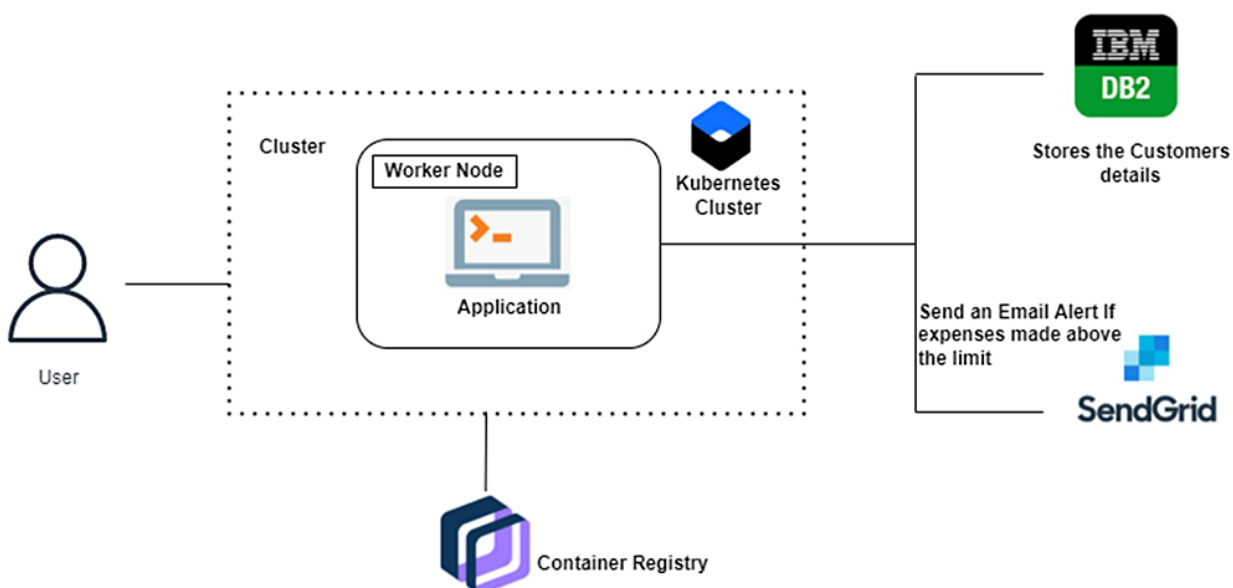


**Solution architecture:**





## Technical Architecture:

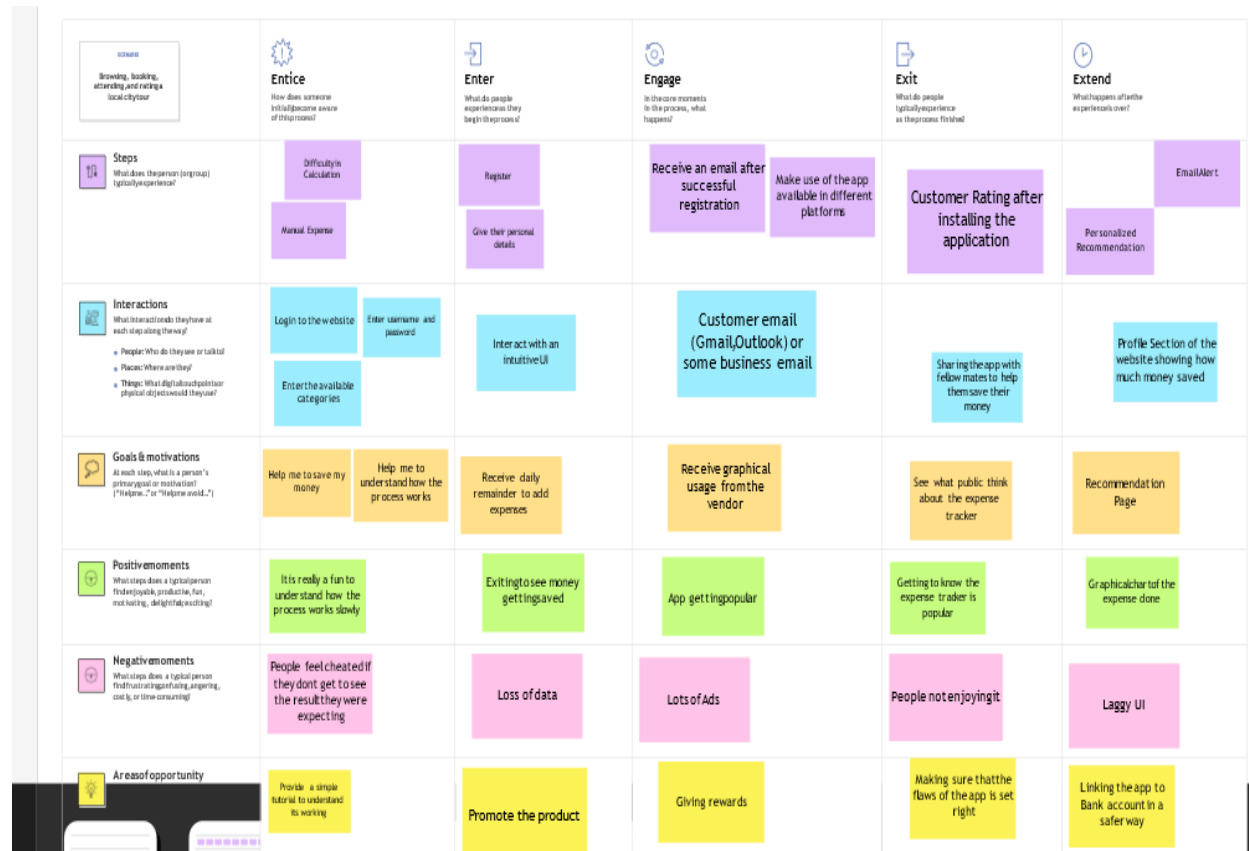


## Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Flask-Python
3.	Cloud Database	Database Service on Cloud	IBM DB2
4.	File Storage	File storage requirements	IBM Object Storage, Container registry
5.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	IBM Cloud , Kubernetes.

### 5.3 User Stories

This gives the user experience of the user.  
we have made the customer map of the user.



## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

The phase by phase development of the project has been specified as sprint plan.

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.	3	High	Venkata Gunashekar, Shaama
Sprint-1	Home	USN-2	As a user, I can log into the application by entering email & password	2	High	Rahul kumar, Gunashekar, Shaama
Sprint-2	Login	USN-3	As a user enters the application see's this page first	3	High	Venkata Gunashekar, Shaama
Sprint-3	Profile	USN-4	This page shows the information of the user	6	High	Rahul Kumar, Venkata

						Gunashekar
Sprint-4	Dashboard	USN-5	This page consists the total expense and total budget, List of expense done by the user	9	High	Rahul kumar, Venkata Gunashekar

## 6.2 Sprint Delivery Schedule

The deliverable are listed as per the schedule and they are tabulated.

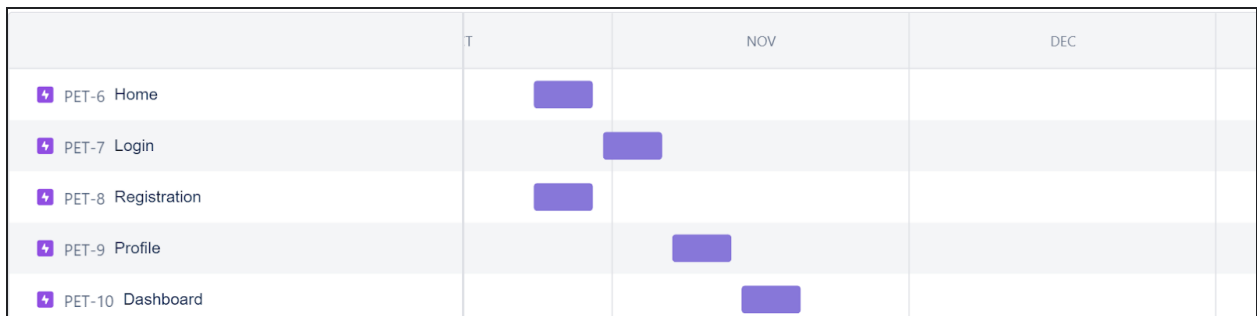
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	5	6 Days	24 Oct 2022	29 Oct 2022	5	29 Oct 2022
Sprint-2	3	6 Days	31 Oct 2022	05 Nov 2022	3	05 Oct 2022
Sprint-3	6	6 Days	7 Oct 2022	12 Oct 2022	6	12 Oct 2022
Sprint-4	9	6 Days	14 Oct 2022	19 Oct 2022	9	19 Oct 2022

## 6.3 Reports from JIRA

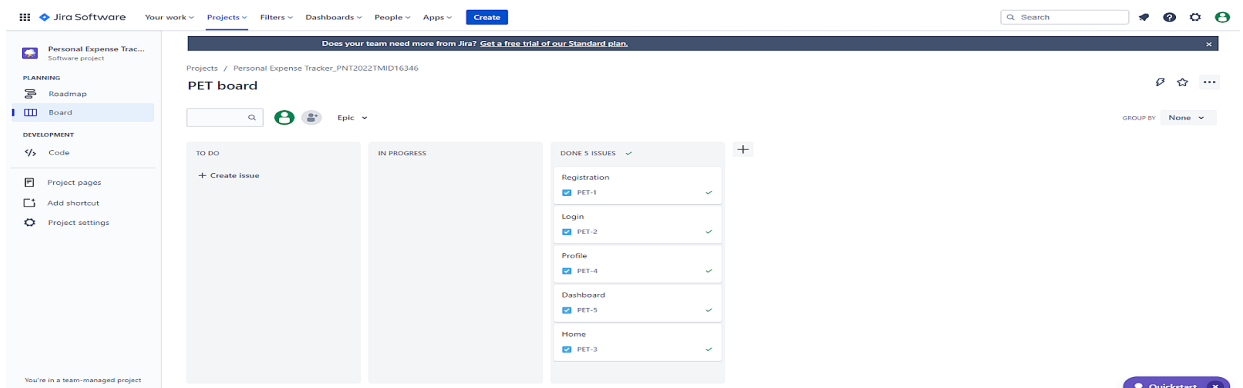
What is analytics Jira



Analytics Plus for Jira Software is **designed to serve as a unified platform with insights on multi-project metrics, workflows, roadmaps, issues, backlogs, and bottlenecks.**



JIRA REPORT ANALYSIS 1.1



JIRA REPORT ANALYSIS 1.2

## 7. CODING & SOLUTIONING (Explain the features added in the project along with code)

### 7.1 Feature 1

IBM-managed cloud databases feature native security capabilities powered by IBM Cloud Security, **protecting in-flight data and at-rest data**. Databases support any data you bring to IBM Cloud: structured, unstructured, SQL, NoSQL, IoT, blockchain and more

```
ibm_db_conn = ibm_db.connect(''DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.
                               bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;
                               SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;
                               UID=cgz83208;PWD=K56N6PSUZD5ZcQ3f'', '', '')

connection = ibm_db_dbi.Connection(ibm_db_conn)
cursor = connection.cursor()
# connection = sqlite3.connect('database/db.sqlite', check_same_thread=False)
# cursor = connection.cursor()
cursor.execute(''CREATE TABLE IF NOT EXISTS expense (
    id INTEGER NOT NULL,
    user_id INTEGER NOT NULL,
    title VARCHAR(50) NOT NULL,
    category VARCHAR(50) NOT NULL,
    amount FLOAT NOT NULL,
    date DATE ,
    PRIMARY KEY (id)
)''')
cursor.execute(''CREATE TABLE IF NOT EXISTS user (
    id INTEGER NOT NULL,
    email VARCHAR(100) NOT NULL,
    password VARCHAR(100),
    name VARCHAR(1000),
    budget FLOAT,
    number VARCHAR(10),
    gender VARCHAR(100),
    address VARCHAR(100),
    PRIMARY KEY (id),
    UNIQUE (email)
)''')
```

IBM DB

## 7.2 Feature 2

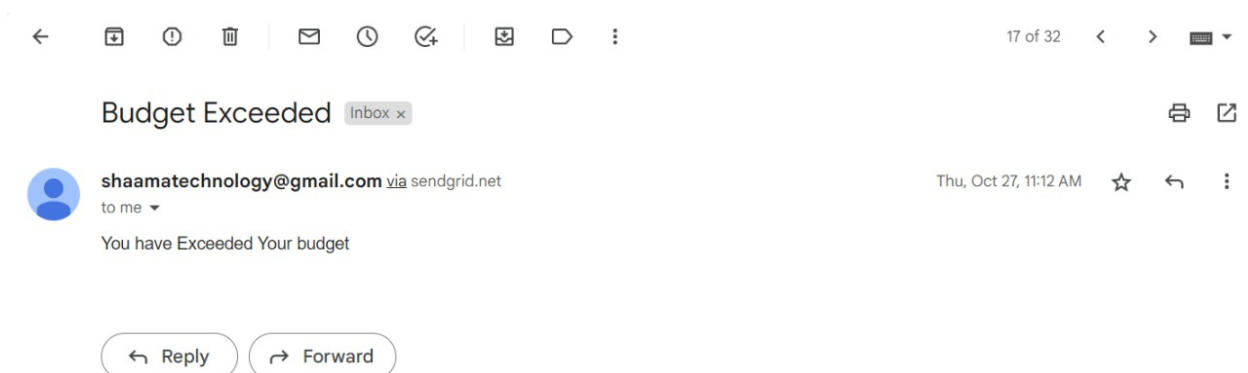
SendGrid is a cloud-based SMTP provider that **allows you to send email without having to maintain email servers**. SendGrid manages all of the technical details, from scaling the infrastructure to ISP outreach and reputation monitoring to whitelist services and real time analytics.

```
21
22 SUBJECT = "Budget"
23 s = smtplib.SMTP('smtp.gmail.com', 587)
24 def sendgridmail(user,TEXT):
25     sg = sendgrid.SendGridAPIClient('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')
26     from_email = Email("shaamatechology@gmail.com") # add the team senders mail id
27     to_email = To(user)
28     subject = "Budget Exceeded"
29     content = Content("text/plain",TEXT)
30     mail = Mail(from_email, to_email, subject, content)
31     mail_json = mail.get()
32     response = sg.client.mail.send.post(request_body=mail_json)
33     print(response.status_code)
34     print(response.headers)
```

Send grid functional code

```
if totalBudgetNow<totalExpenseNow and totalBudgetNow!=0:
    sendgridmail(current_user.email,"You have Exceeded Your budget")
```

if conditional code



### 7.3 Database Schema

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

Find schemas or tables

Refresh

SQL

Schemas

Tables

Table definition

Name

EXPENSE

CGZ83208

...

USER

CGZ83208

...

Total: 2, selected: 0

Table definition

USER

Approximate 4 rows (32.0 KB)

Updated on 2022-11-17 13:25:22

Name	Data type	Nullable	Length	Scale
ID	INTEGER	N		0
EMAIL	VARCHAR	N	100	0
PASSWORD	VARCHAR	Y	100	0
NAME	VARCHAR	Y	1000	0
BUDGET	DOUBLE	Y		0
NUMBER	VARCHAR	Y	10	0
GENDER	VARCHAR	Y	100	0
ADDRESS	VARCHAR	Y	100	0

View data



## 8.TESTING

### 8.1 Test Cases


Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute
LoginPage_TC_Q1	Functional	Home Page	Verify user is able to see the LoginSignup popup when user clicked on My account button	Text field should be enabled	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not
LoginPage_TC_Q2	UI	Home Page	Verify the UI elements in LoginSignup popup	Text field should be enabled	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link
LoginPage_TC_Q3	Functional	Home page	Verify user is able to log into application with Valid credentials	Text field should be enabled	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box
LoginPage_TC_Q4	Functional	Login page	Verify user is able to log into application with Invalid credentials	Text field should be enabled	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button
LoginPage_TC_Q4	Functional	Login page	Verify user is able to log into application with Invalid credentials	Text field should be enabled	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box
LoginPage_TC_Q5	Functional	Login page	Verify user is able to log into application with Invalid credentials	Text field should be enabled	1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button

Login - Personal Expense Tracker

Not secure | 169.51.203.77:30948/login

VSV

Home Sign in



Sign in

Email address

Password

☐ Remember me

Sign in

Create an Account

© vsv

## 8.2 User Acceptance Testing

Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Username: vinay@gmail.com password: Testing123	Login/Signup popup should display	Working as expected	Pass	Nil	Y	Nil	Rahul
Username: vinay@gmail.com password: Testing123	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass	good	Y	Nil	vinay
Username: vinay@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	Pass	nice	Y	Nil	venkata
Username: vinaykrishna174@gmail.com password: vina123	Application should show 'Incorrect email or password' validation message.	Working as expected	Pass	Good	Y	Nil	shaama
Username: shaama@gmail.com password: Testing123678686	Application should show 'Incorrect email or password' validation message.	Working as expected	Pass	good	Y	Nil	shaama
Username: vinaykrishna174@gmail.com	Application should show 'Incorrect email or password' validation message.	Working as expected	Pass	Nice	Y	Nil	rahul

□      —

**Acceptance Testing  
UAT Execution & Report Submission**

Date	03 November 2022
Team ID	PNT2022TMID16140
Project Name	Personal Expense Tracker
Maximum Marks	4 Marks

### 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	3	19
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	24	9	9	24	66

### 3. Test Case Analysis

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


Section	Total Cases	Not Tested	Fail	Pass
Functional	7	0	0	7
Unit	5	0	0	5
Security	2	0	0	2
Version Control	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4

<b>Test Scenarios</b>	
1 Verify user is able to see login page	yes
2 Verify user is able to log into application or not?	yes
3 Verify user is able to navigate to create your account page?	yes
4 Verify user is able to recovery password	no
5 Veriify login page elements	yes
<b>Search</b>	
1 Verify user is able to search by entering keywords in search box	no
2 Verify user is able to see suggestions based on keyword entered in search box	no
3 Verify user is able to see related auto suggestions displaying based on keyword entered in search box	yes
4 Verify user is able to see no matches found message when no results are matching with entered keyword	yes
5 Verify user is able to see seach detailed page when nothing entered in textbox	no

Sign Up - Personal Expense Tracker

Not secure | 169.51.203.77:30948/signup

VSV Home Sign in



### Sign Up

Username  
User

Email address  
vgsece@gmail.com

Password  
\*\*\*\*\*

Sign Up

© vsv

## 9.RESULTS

### 9.1Performance Metrics

			NFT - Risk Assessment						
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volumen Changes	Risk Score	Justification
1	Personal Expense Tracker	Achieve your business goals	Low	No Changes	Moderate	Unenforceable Policies	>5 to 10%	ORANGE	As we have seen the changes
		Boost the productivity				Processing Costs			
		valuable insights				Costs of (Lacking) Integration			
		Avoid data loss							
			NFT - Detailed Test Plan						
			S.No	Project Name	NFT Test approach	assumptions/Dependencies/Risk	Approvals/SignOff		
			1	Personal Expense Tracker	Performance Testing	web based application	Approved		
					Security Testing	Flask	Approved		
						Kubernetes			
						Docker			
			End Of Test Report						
S.No	Project Name	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	(Detected/Closed/Open)	Approvals/SignOff	
1	Personal Expense Tracker	Performance Testing	90	High	Go	No recommendation	closed	Approved	
		Security Testing	95	High security	Go	No recommendation	closed	Approved	

## 10.ADVANTAGES & DISADVANTAGES

### Pros

It's simple to set up and use. When you're creating your own method of tracking your finances, you first have to figure out how you're going to do that. Are you going to use pen and paper, or software, or an excel spreadsheet? What are you going to track? How are you going to input that data, and how often are you going to do it? With an automated app, it tracks everything for you in real time. It has a wealth of information, so no matter what data you feel is important to track, it is all there and available for you – you just need to take a look to see it. There's an easy user interface for everything as well. Whether you want to set up budget , track a type of expense, or look over your financial history, there's a tab or an option ready and waiting for you.

It's instantaneous. The application will track all of your data for you. It doesn't do it once a week or once a month, like you might if you were doing it manually. All the information is automatically brought right into your account as soon as it is available. That means that you have a day by day way of checking to ensure that you're on track and moving in the right direction. I've had it where Mint.com actually helped me react very quickly to a fraudulent transaction from an ATM machine. Someone had gotten a hold of our bank account details through a card skimmer (most likely), and they withdrew a bunch of cash. Within a few hours, I had already been made aware of it through Mint, contacted my bank, and started the process of getting my money back, all thanks to the automated system that was already in place.

## Cons

Your information is less secure, and probably being used and sold. If the service is free, then the product is you. Mint.com, like other financial apps, is a free service. They have to pay their bills somehow, so regardless of what their privacy policy may or may not say, just assume that your spending history and trends are going to be recorded and analyzed, by someone, somewhere. Now, you shouldn't have to worry about credit card fraud or identity theft; these companies are large enough and secure enough that you'll never have to worry about something like that. Just recognize that your information, most likely anonymous, will be used and potentially even sold. Personally, I have no problem with that, but if you do, then make sure you avoid these types of services.

Automating everything to do with your finances can make you financially lazy. If your bills are paid automatically and your finances are tracked automatically, then what is there left for you to do? Not a lot, to be honest. So you might stop caring about what you're spending and where your money is going. Eventually you may look at your Mint data and realize that you've blown your budget over the last two months, but by then it is too late. So if you do choose to use this program, ensure that you are also being.

ADVANTAGES	DISADVANTAGES
LOW OPERATING COSTS	SECURITY AND PRIVACY ISSUE
HIGHER EMPLOYEE PRODUCTIVITY	LACK OF TECHNICAL KNOWLEDGE
BETTER CUSTOMERS EXPERIENCE	INTERNET CONNECTION
CONSUMER INSIGHTS	TIME CONSUMING



## **11.CONCLUSION**

The new system has overcome most of the limitations of the existing system and works according to the design specification given. The project what we have developed is work more efficient than the other income and expense tracker. The project successfully avoids the manual calculation for avoiding calculating the income and expense per month. The modules are developed with efficient and also in an attractive manner. The developed systems dispense the problem and meet the needs of by providing reliable and comprehensive information. All the requirements projected by the user have been met by the system.

The newly developed system consumes less processing time and all the details are updated and processed immediately. Since the screen provides online help messages and is very user friendly, any user will get familiarized with its usage. Module s are designed to be highly flexible so that any failure requirements can be easily added to the modules without facing many problems. 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

## **12.FUTURE SCOPE**

- 1) It will have various options to keep record (for example Food, Travelling Fuel, Salary etc.
- 2) Automatically it will keep on sending notifications for our daily expenditure.
- 3) In today's busy and expensive life, we are in a great rush to make moneys, but at the end of the month we broke off. As we are unknowingly spending money on title and unwanted things. So, we have come over with the plan to follow our profit.
- 4) Here user can define their own categories for expense type like food, clothing, rent and bills where they have to enter the money that has been spend and likewise can add some data in extra data to indicate the expense.

## 13. APPENDIX

### Source Code

```
1 from flask import Flask, render_template,
   redirect,url_for,request, flash
2 from flask_login import LoginManager,login_user, login_required,
   logout_user, current_user
3 import datetime
4 from flask_wtf import FlaskForm
5 from wtforms import StringField, DecimalField, SubmitField,
   DateField
6 from wtforms.validators import DataRequired
7 from werkzeug.security import generate_password_hash,
   check_password_hash
8 from flask_login import UserMixin
9 from wtforms import StringField, DecimalField, SubmitField,
   DateField
10 import ibm_db,ibm_db_dbi,sqlite3
11 import os
12 from flask import Flask, render_template, request, redirect,
   url_for, flash
13 # from flask_mail import Mail, Message
14 from sendgrid.helpers.mail import From, To
15 import smtplib
16 import sendgrid
17 import os
18 from sendgrid.helpers.mail import Mail, Email, To, Content
19
20
21 SUBJECT = "Budget"
22 s = smtplib.SMTP('smtp.gmail.com', 587)
23 def sendgridmail(user,TEXT):
24     sg =
25         sendgrid.SendGridAPIClient('SG.HEYycx0wQeuyh0AztQIQ1w.BBFcjQTgIW3
26         aFCgofqUtLIr6NcJi_yIT9XRW_DtnGUs')
27     from_email = Email("shaamatechnology@gmail.com") # add the
28     team senders mail id
29     to_email = To(user)
30     subject = "Budget Exceeded"
31     content = Content("text/plain",TEXT)
```

```

29     mail = Mail(from_email, to_email, subject, content)
30     mail_json = mail.get()
31     response = sg.client.mail.send.post(request_body=mail_json)
32     print(response.status_code)
33     print(response.headers)
34
35
36 ibm_db_conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-
    86c7-4d5b-8580-
    3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=
    31505;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;
    UID=cgz83208;PWD=K56N6PSUzD5ZcQ3f",'','')
37
38
39 connection = ibm_db_dbi.Connection(IBM_DB_CONN)
40 cursor = connection.cursor()
41 # connection = sqlite3.connect('database/db.sqlite',
    check_same_thread=False)
42 # cursor = connection.cursor()
43 cursor.execute(''CREATE TABLE IF NOT EXISTS expense (
44     id INTEGER NOT NULL,
45     user_id INTEGER NOT NULL,
46     title VARCHAR(50) NOT NULL,
47     category VARCHAR(50) NOT NULL,
48     amount FLOAT NOT NULL,
49     date DATE ,
50     PRIMARY KEY (id)
51 )'')
52 cursor.execute(''CREATE TABLE IF NOT EXISTS user (
53     id INTEGER NOT NULL,
54     email VARCHAR(100) NOT NULL,
55     password VARCHAR(100),
56     name VARCHAR(1000),
57     budget FLOAT,
58     PRIMARY KEY (id),
59     UNIQUE (email)
60 )'')
61 connection.commit()
62 cursor.execute("SELECT * from user where id = 0")
63 lu = cursor.fetchone()

```

```

64 if not lu:
65     cursor.execute(''INSERT INTO user VALUES
        (0,'null@null',0,'null',null);'')
66     cursor.execute(''INSERT INTO expense VALUES
        (0,0,'null','null',0,'1000-01-01')'')
67     connection.commit()
68
69 class User(UserMixin):
70     def __init__(self, id, email, password, name,active=True):
71         self.id = id
72         self.email = email
73         self.password = password
74         self.name = name
75         self.authenticated = False
76         self.active = active
77     def is_active(self):
78         return self.active
79     def is_anonymous(self):
80         return False
81     def is_authenticated(self):
82         return self.authenticated
83     def is_active(self):
84         return True
85     def get_id(self):
86         return self.id
87 class ExpenseForm(FlaskForm):
88     title = StringField('Title', validators=[DataRequired()])
89     category = StringField('Category',
        validators=[DataRequired()])
90     amount = DecimalField('Amount', validators=[DataRequired()])
91     date = DateField('Date')
92     submit = SubmitField('Submit')
93
94     def __repr__(self):
95         return f"ExpenseForm('{self.title}', '{self.category}',
        {self.amount}, {self.date})"
96 app = Flask(__name__)
97
98 app.config['SECRET_KEY'] = 'secret-key-goes-here'
99 # app.config['SECRET_KEY'] = 'top-secret!'

```

```

100 # app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
101 # app.config['MAIL_PORT'] = 587
102 # app.config['MAIL_USE_TLS'] = True
103 # app.config['MAIL_USERNAME'] = 'apikey'
104 # app.config['MAIL_PASSWORD'] =
    os.environ.get('SG.M1_C4UjaTTWxS2ijm5PtLA.q7amwH3-
        Y4Tz1jVe10adWFRGp76lVYt3qz1Xkx54VL8')
105 # app.config['MAIL_DEFAULT_SENDER'] =
    os.environ.get('shaamatechnology@gmail.com')
106 # mail = Mail(app)
107
108 login_manager = LoginManager()
109 login_manager.login_view = 'login'
110 login_manager.init_app(app)
111
112 @login_manager.user_loader
113 def load_user(user_id):
114
115     cursor.execute("SELECT * from user where id =
        (?)", (user_id,))
116     lu = cursor.fetchone()
117     if lu is None:
118         return None
119     else:
120         return User(int(lu[0]), lu[1], lu[2], lu[3])
121
122 @app.route('/login')
123 def login():
124     return render_template('login.html')
125
126 @app.route('/dashboard')
127 def dashboard():
128     return render_template('dashboard.html')
129
130 @app.route('/login', methods=['POST'])
131 def login_post():
132     # login code goes here
133     email = request.form.get('email')
134     password = request.form.get('password')
135     remember = True if request.form.get('remember') else False

```

```

136
137     cursor.execute("SELECT * from user WHERE email =
138         ?",(email,))
138     user=cursor.fetchone()
139     print()
140
141     # check if the user actually exists
142     # take the user-supplied password, hash it, and compare it
143     # to the hashed password in the database
143     if user==None or not check_password_hash(user[2], password):
144         flash('Please check your login details and try again.')
145         return redirect(url_for('login')) # if the user doesn't
146         exist or password is wrong, reload the page
146     LUser = load_user(user[:][0])
147     # if the above check passes, then we know the user has the
148     # right credentials
148     login_user(LUser, remember=remember)
149     return redirect(url_for('home'))
150
151 @app.route('/signup')
152 def signup():
153     return render_template('signup.html')
154
155 @app.route('/signup', methods=['POST'])
156 def signup_post():
157     # code to validate and add user to database goes here
158     email = request.form.get('email')
159     name = request.form.get('name')
160     password = request.form.get('password')
161
162
163     # if this returns a user, then the email already exists in
164     # database
164     cursor.execute("SELECT * from user WHERE email =
165         ?",(email,))
165     if cursor.fetchone():
166         flash('Email address already exists')
167         return redirect(url_for('signup'))
168
169     #creating user id

```

```

170     cursor.execute(''SELECT id FROM user ORDER BY id DESC;'')
171     user_id=cursor.fetchone()[0]+1
172     print('registered user id == ',user_id)
173
174
175     # create a new user with the form data. Hash the password so
    the plaintext version isn't saved.
176     print(user_id,email,generate_password_hash(password,
    method='sha256'),name,0)
177     cursor.execute('INSERT INTO user VALUES (?,?,?,?,?);',
    (user_id,email,generate_password_hash(password,
    method='sha256'),name,0,))
178     connection.commit()
179     flash('Account Created Successfully..')
180
181     return redirect(url_for('login'))
182
183 @app.route('/logout')
184 @login_required
185 def logout():
186     logout_user()
187     return redirect(url_for('home'))
188
189 @app.route('/index')
190 def index():
191     if current_user.is_authenticated:
192         connection.commit()
193         cursor.execute("SELECT id,budget from user WHERE email =
    ?",(current_user.email,))
194         k=cursor.fetchone()
195         print(k)
196         user_id=k[0]
197         totalBudgetNow=k[1]
198         cursor.execute("SELECT * from expense WHERE user_id =
    ?",(user_id,))
199         all_expenses =cursor.fetchall()
200         d=datetime.datetime.now()
201         print(d.strftime("%m"),d.strftime("%Y"))
202         current_month=d.strftime("%B")
203         print(all_expenses)

```

```

204         cursor.execute("SELECT SUM(amount) from expense WHERE
        user_id = ? AND MONTH(date) = ? AND YEAR(date) =
        ?",(user_id,d.strftime("%m"),d.strftime("%Y"),))
205         k=cursor.fetchone()[0]
206         totalExpenseNow=0
207         if k!=None:
208             totalExpenseNow=k
209
210         cursor.execute("SELECT SUM(amount) from expense WHERE
        user_id = ?",(user_id,))
211         k=cursor.fetchone()[0]
212         totalExpense=0
213         if k!=None:
214             totalExpense=k
215
216
217
218
219         if totalBudgetNow<totalExpenseNow:
220             sendgridmail(current_user.email,"You have Exceeded
        Your budget")
221
222         return render_template('index.html',
        expenses=all_expenses,email=current_user.email
        ,name=current_user.name,totalExpense=totalExpense,totalExpenseNow
        =totalExpenseNow,totalBudgetNow=totalBudgetNow,current_month=curr
        ent_month)
223
224
225         return render_template('index.html')
226
227 @app.route('/index', methods=['POST'])
228 @login_required
229 def index_post():
230     budget = request.form.get('budget')
231     print(budget)
232     cursor.execute("SELECT id from user WHERE email =
        ?",(current_user.email,))
233     user_id=cursor.fetchone()[0]
234     cursor.execute('UPDATE user SET budget=? WHERE id=?;',

```



```

        (budget, user_id,))
235     connection.commit()
236     return redirect('/index')
237
238
239
240 # assign home route and / route to home
241 @app.route('/')
242 def home():
243     if current_user.is_authenticated:
244         connection.commit()
245         return
        render_template('home.html',email=current_user.email,name=current
        _user.name)
246
247     return render_template('home.html')
248
249
250 @app.route('/add', methods=['POST', 'GET'])
251 @login_required
252 def add():
253     form = ExpenseForm()
254     #if form successfully validated
255     if form.validate_on_submit():
256
257         # we create an expense object
258         cursor.execute("SELECT id from user WHERE email =
        ?",(current_user.email,))
259         user_id=cursor.fetchone()[0]
260
261         #creating expense id
262         cursor.execute(''SELECT id FROM expense ORDER BY id
        DESC;'')
263         expense_id=cursor.fetchone()[0]+1
264         print('created expense id == ',expense_id)
265
266         cursor.execute('INSERT INTO expense VALUES (?,?,,?,,?);',
        (expense_id, user_id, form.title.data, form.category.data,
        str(form.amount.data), form.date.data,))
267         connection.commit()

```

```

268     return redirect(url_for('index'))
269
270     form.date.data = datetime.datetime.now()
271     return render_template('add.html', form=form)
272
273 @app.route("/update/<int:expense_id>", methods=['GET', 'POST'])
274 @login_required
275 def update(expense_id):
276     cursor.execute("SELECT id from user WHERE email =
        ?",(current_user.email,))
277     user_id=cursor.fetchone()[0]
278     cursor.execute("SELECT * from expense WHERE (user_id = ? AND
        id = ?)",(user_id,expense_id,))
279     expense=cursor.fetchone()
280     form = ExpenseForm()
281     print('updating expense_id=====',expense_id)
282     # if the form is validated and submitted, update the data
    of the item
283     # with the data from the field
284     if form.validate_on_submit():
285
286         cursor.execute('UPDATE expense SET
            title=?,category=?,amount=?,date=? WHERE id=?;',
            (form.title.data, form.category.data, str(form.amount.data),
            form.date.data, expense_id,))
287         connection.commit()
288         return redirect(url_for('index'))
289         # populate the field with data of the chosen expense
290     elif request.method == 'GET':
291         form.title.data = expense[2]
292         form.category.data = expense[3]
293         form.amount.data = expense[4]
294         form.date.data = expense[5]
295         return render_template('add.html', form=form, title='Edit
            Expense')
296
297 @app.route("/delete/<int:expense_id>")
298 @login_required
299 def delete(expense_id):
300     print('deleting expense_id=====',expense_id)

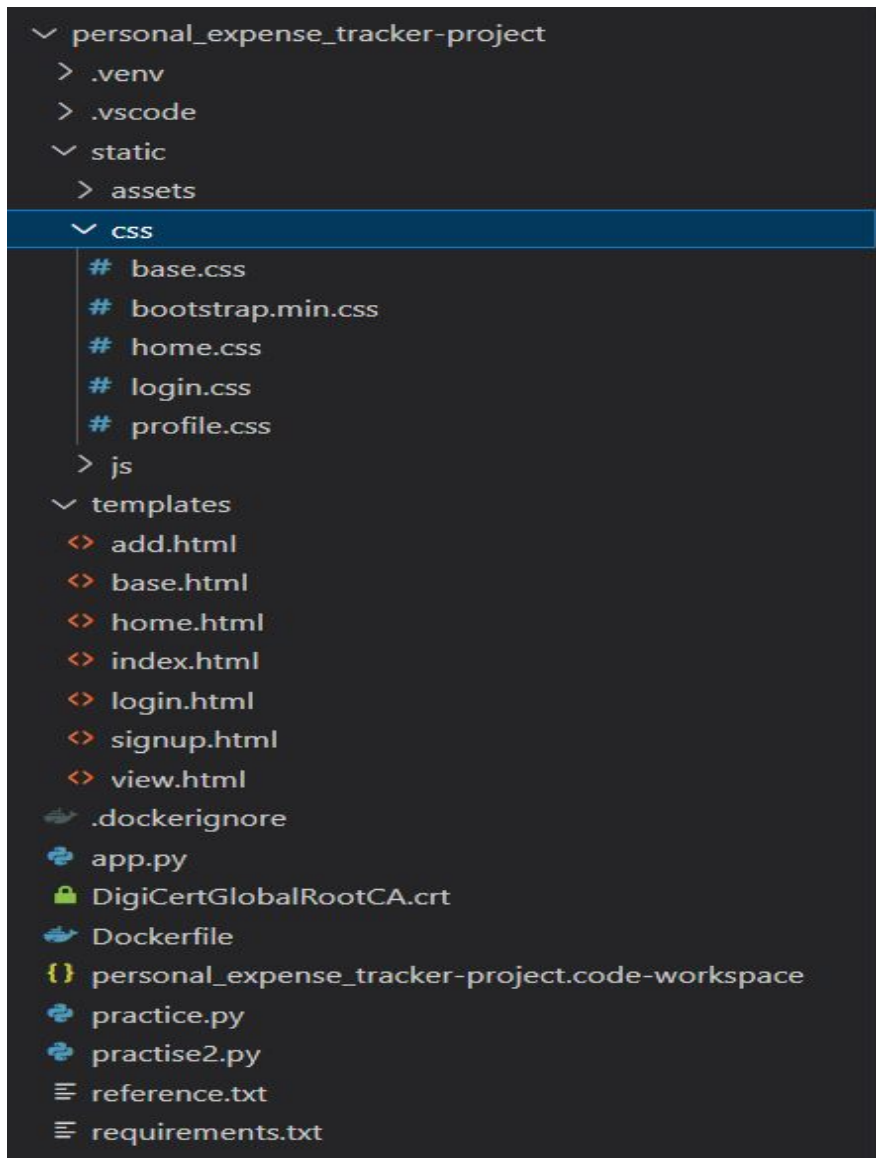
```

```

301     cursor.execute("DELETE FROM expense WHERE
        id=?",(expense_id,))
302     connection.commit()
303
304     return redirect(url_for('index'))
305
306 if __name__ == '__main__':
307     app.run(debug=True,host='0.0.0.0',port=8080,threaded=True)

```

#### File order



### Python Libraries that are being used:-

```
click==8.1.3  
colorama==0.4.6  
Flask==2.2.2  
Flask-Login==0.6.2  
Flask-WTF==1.0.1  
ibm-db==3.1.3  
itsdangerous==2.1.2  
Jinja2==3.1.2  
MarkupSafe==2.1.1  
python-http-client==3.3.7  
sendgrid==6.9.7  
starkbank-ecdsa==2.2.0  
Werkzeug==2.2.2  
WTForms==3.0.1
```

### GitHub & Project Demo Link

#### **GIT HUB LINK:**

GIT HUB (PERSONAL EXPENSER TRACKER)

<https://github.com/IBM-EPBL/IBM-Project-6778-1658837124>

#### **DEMO VIDEO LINK:**

DEMO VIDEO LINK (PERSONAL EXPENSE TRACKER)

[https://drive.google.com/drive/folders/1budcrx64bGZg8CsZnVQXvAJON0tYXvIV?usp=share\\_link](https://drive.google.com/drive/folders/1budcrx64bGZg8CsZnVQXvAJON0tYXvIV?usp=share_link)

**DEMO WEBSITE LINK:**

DEMO WEBSITE LINK (PERSONAL EXPENSE TRACKER-deployed in kubernetes)

<http://169.51.203.77:30948/>