

PROJECT REPORT

PERSONAL EXPENSE TRACKER

Team ID	PNT2022TMID26352
Project Name	PERSONAL EXPENSE TRACKER
Team Members	1) HARISH KUMAR.R 2) JAYARAJ.S 3) KAARTHIK.S 4) DHANUSH.S 5) ADITI.S

Table Of Contents

SI No	Title	Page No
1	INTRODUCTION 1.1 Project Overview 1.2 Purpose	4
2	LITERATURE SURVEY 2.1 Existing problem 2.2 References 2.3 Problem Statement Definition	5
3	IDEATION & PROPOSED SOLUTION 3.1 Empathy Map Canvas 3.2 Ideation & Brainstorming 3.3 Proposed Solution 3.4 Problem Solution fit	7
4	REQUIREMENT ANALYSIS 4.1 Functional requirement 4.2 Non-Functional requirements	13
5	PROJECT DESIGN 5.1 Data Flow Diagrams 5.2 Solution & Technical Architecture 5.3 User Stories	14

6	PROJECT PLANNING & SCHEDULING 6.1 Sprint Planning & Estimation 6.2 Sprint Delivery Schedule 6.3 Reports from JIRA	17
7	CODING & SOLUTIONING 7.1 Feature 1 7.2 Feature 2 7.3 Database Schema (if Applicable)	19
8	TESTING 8.1 Test Cases 8.2 User Acceptance Testing	23
9	RESULTS 9.1 Performance Metrics	25
10	ADVANTAGES & DISADVANTAGES	31
11	CONCLUSION	32
12	FUTURE SCOPE	32
13	APPENDIX	32

1. INTRODUCTION

1.1 PROJECT OVERVIEW:

Personal Expense Tracker is a daily expense management system which is specially designed for non- salaried and salaried personnel for keeping track of their daily expenditure with easy and effective way through computerized system which tends to eliminate manual paper works. It will also manage records in systematic way and user can access the stored data conveniently. We have tried to design the project in such way that user may not have any difficulty in using this application without much effort. This software can be really used by end user who have Android running devices with them. The language that we use to develop this system is and MYSQL for database.

1.2 PURPOSE:

The idea of developing this project for user convenience. Because whenever they make expenses immediately, they add in the application. Some of the concerns maintaining a personal expense is a BIG problem, in daily expenses many times we don't know where the money goes. Some of the conventional methods used to tackle this problem in normal circumstances are like making use of a sticky notes by common users, Proficient people deals with this kind of problems by using spreadsheets to record expense and using a ledger to maintain the large amounts data by especially by expert people. We believe a handy design and a handy mobile application which handles these troubles. Such that app is capable of recording the expenditure and giving broad view with easy to use the user interface and this application is intelligent enough to shows the history of expenses.

2. LITERATURE SURVEY

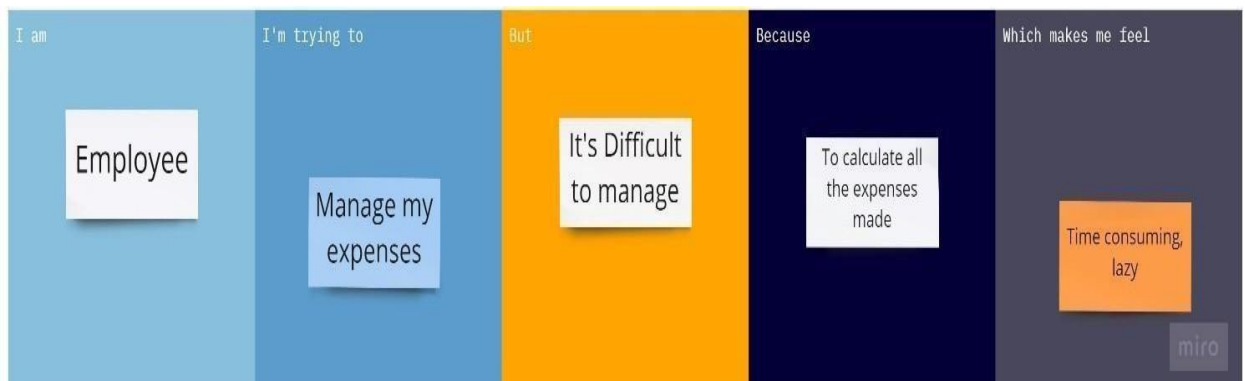
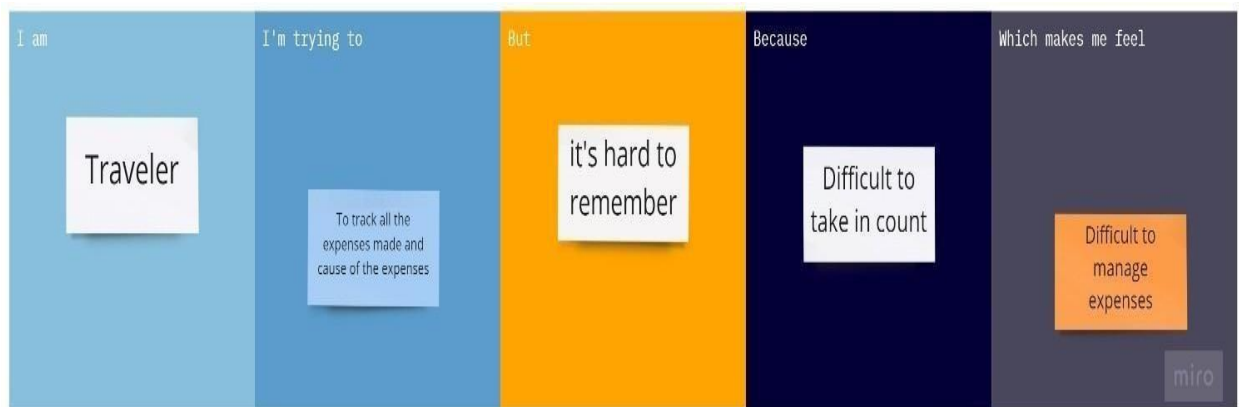
2.1 EXISTING PROBLEM:

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 has to keep a login 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 remind a person in a specific date, so that is the only drawback in which the reminder 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 be used to perform calculation on income and expenses to overcome this problem we propose the new project.

2.2 REFERENCES:

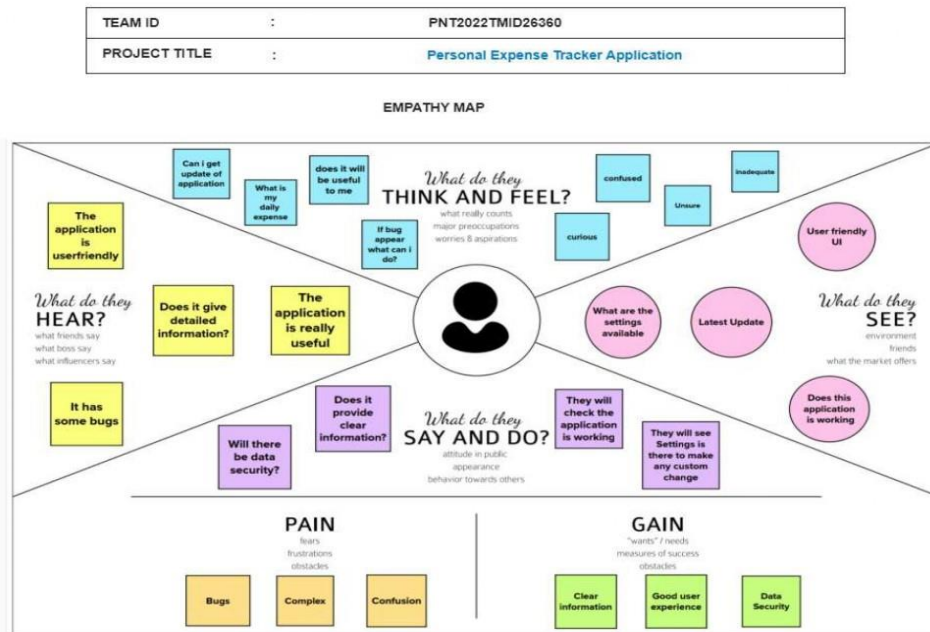
- [1] <http://expense-manager.com/how-expense software/>
- [2] <https://www.splitwise.com/terms>
- [3] <http://code.google.com/p/socialauth-android/wiki/Facebook>
- [4] <http://code.google.com/p/socialauth-android>
- [5] Developer.android.com
- [6] <http://www.appbrain.com/app/expensemanager/com.expensemanager>

2.3 PROBLEM STATEMENT:



3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION & BRAINSTORMING:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash the imagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Conducting a brainstorm

Executing a brainstorm isn't unique; holding a productive brainstorm is. Great brainstorms are ones that set the stage for fresh and generative thinking through simple guidelines and an open and collaborative environment. Use this when you're just kicking-off a new project and want to hit the ground running with big ideas that will move your team forward.

- 15 minutes to prepare
- 30-60 minutes to collaborate
- 3-6 people recommended

Meta Meta

Share template feedback

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.

15 minutes

- Choose your best "How Might We" Questions**
Create 5 HMTL statements before the activity to propose them to the team.
- Set the stage for creativity and inclusivity**
Go over the brainstorming rules and keep them in front of your team while brainstorming to encourage collaboration, optimism, and creativity.
 1. **Encourage wild ideas** (If none of the ideas sound a bit ridiculous, then you are thinking yourself too much.)
 2. **Defer judgement** (This can be as direct as harsh words or as subtle as a condescending tone or taking over one another.)
 3. **Build on the ideas of others** ("I want to build on that idea" or the use of "yes, and...")
 4. **Stay focused on the topic at hand**
 5. **Have one conversation at a time**
 6. **Be visual** (Draw and/or repeat to show ideas, whenever possible.)
 7. **Go for quantity**
- Interested in learning more?**
Check out the Meta Team Kit website for additional tools and resources to help your team collaborate, innovate and move ideas forward with confidence.

[Open the website →](#)

Choose your best "How Might We" Questions

Share the top 5 brainstorm questions that you created and let the group determine where to begin by selecting one question to move forward with based on what seems to be the most promising for idea generation in the areas you are trying to impact.

12 minutes

QUESTION

How might we... (insert problem statement here)?

QUESTION

How might we... (insert problem statement here)?

QUESTION

How might we... (insert problem statement here)?

QUESTION

How might we... (insert problem statement here)?

QUESTION

How might we... (insert problem statement here)?

Step-2: Brainstorm, Idea Listing and Grouping

2

Brainstorm solo

Have each participant begin in the "solo brainstorm space" by silently brainstorming ideas and placing them into the template. This "silent-storming" avoids group-think and creates an inclusive environment for introverts and extroverts alike. Set a time limit. Encourage people to go for quantity.

🕒 10 minutes

Person 1	Person 2	Person 3	Person 4
Creating limited way of tracking	More financial features	Basics of the enterprise	Remembering the user in business
Finding the best way to implement	Avoiding unwanted checkboxes of visibility	Adding weekly reports	User can use this for any purpose
More control with the resources	Providing security updates	Identifying the particular business needs	User can use this for their regular job
Can be used by anyone	Identifying the requirements and fix it	Users who monitor expenses and give their reports	Getting full access to the data about the expenses
No age limits	Sending email for verification	Performance tracker	Int and start according to budget
Works for single person as well as for business	Encrypting information	Adding the commented expense section	Summarized through email if the expense is out of control
Simple thoughtful designs about expenses	Handling passwords for expenses login and fix	Determining the budget	Easy adaptation for add-on expenses
Limited access from the user	Securing valuable information given by users	Determining the reports according to budget	Setting up average limit
Allowing and restricting the user to control expenses	Analytic stage of add		Limiting the expenses

3

Brainstorm as a group

Have everyone move their ideas into the "group sharing space" within the template and have the team silently read through them. As a team, sort and group them by thematic topics or similarities. Discuss and answer any questions that arise. Encourage "Yes, and..." and build on the ideas of other people along the way.

🕒 15 minutes

TIP

You can use the **Voting session** tool above to focus on the strongest ideas.



Unassigned areas					
Average Limit					
Setting up average limit	Notify the user when exceeding	Real time notification	Provide analytics	Weekly report	Notifying the bills
Features					
Weekly reports	Real time expenses notification	Getting the full report for a user	Can link bills through this app	Can replace the expenses	Summary
Validation					
Can stream for any users	No age limit	Handling users expenses	To create where security is important	Supporting multiple users	Report password

Step-3: Voting Section



4

Decide your focus
Give each person two icons to vote which idea should your team focus on.
⌚ 5 minutes



Person 1





Person 2



Person 3



Person 4



3.2 PROPOSED SOLUTION:

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	There are many budgeting tools online, but not all of them are effective in assisting users in actually creating and adhering to a budget. The ongoing maintenance, the consolidation of all user financial accounts and activity into a single dashboard are some of the negatives. However, a lot of this current software includes convoluted features that are difficult to use. The major problem is to take count paper receipts and calculate the expense statistics.

2.	Idea/Solution description	People tend to neglect their budgets due to their busy and chaotic lifestyles, which results in them spending more than they planned. Future costs cannot be foreseen by the user. Their carelessness with money management will be a concern even though they can record their expense.
3.	Novelty/ Uniqueness	Including all the expense including money spend using cash, Bank-cheques, etc. This application keeps track of all of your spending. To enter your expense, simply click. To decrease human error, prevent data loss, and expedite settlements. In this program to display the pie chart or graph lines.
4.	Social Impact / Customer Satisfaction	One can keep track of their own costs and create a monthly or annual budget with this tool. The application will display the statistics and sent alert message, if your spending exceeds the specified limit.
5.	Business Model(Revenue Model)	The subscription/ premium to access extra features of this application can be used by business people, and also adding advertisement to generate revenue.
6.	Scalability of the Solution	Using IBM-cloud statistics can be shown to the user with high scalability, and with high accuracy.

3.3 PROPOSED SOLUTION FIT:

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS <ul style="list-style-type: none"> Customers those who spend money unwontedly and to track their expenses. Customer those who can't remember their expense. Those who expecting to track their expense via statistics. 	6. CUSTOMER CONSTRAINTS CC <ul style="list-style-type: none"> Customer should use UPI or Net-Banking to track the expense. If the money is spend through cash customer must add the expense in the application. 	5. AVAILABLE SOLUTIONS AS <ul style="list-style-type: none"> SPENDEE Application available both android and the ios.
	2. JOBS-TO-BE-DONE / PROBLEMS J&P <ul style="list-style-type: none"> The main Intention of the application is to track the expense and provide statistics of expenses It provides statistics based on categories of expenses. To include money spend through cash, bank cheque's etc. 	9. PROBLEM ROOT CAUSE RC <ul style="list-style-type: none"> The Main problem is gathering the data from the UPI apps or Nat-Banking application. This will act as the main problem of the application. Laziness of the customer to add the expense done through cash in the application. 	7. BEHAVIOUR BE <ul style="list-style-type: none"> Customer should responsibly add the expenses done through off-line mode. To assure the data safety to the user.
	3. TRIGGERS TR <ul style="list-style-type: none"> Customer may think , they spend more money and no saving. 	10. YOUR SOLUTION SL <ul style="list-style-type: none"> Design a cloud based web Application of the expense tracker. Provide statistic of the expense done by the user through the graphs or charts. 	8. CHANNELS of BEHAVIOUR H <ul style="list-style-type: none"> In Online mode user don't have more work user need to set the maximum expense limit. In Off-line mode user should responsibly
Identify strong TR & EM	4. EMOTIONS: BEFORE / AFTER EM <ul style="list-style-type: none"> BEFORE: No Savings. AFTER: Few saving due to expense tracking application. 	<ul style="list-style-type: none"> Providing email alerts if the total expense exceed the limit. 	add the expenses done through cash
			Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 NON - FUNCTIONAL REQUIREMENT:

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

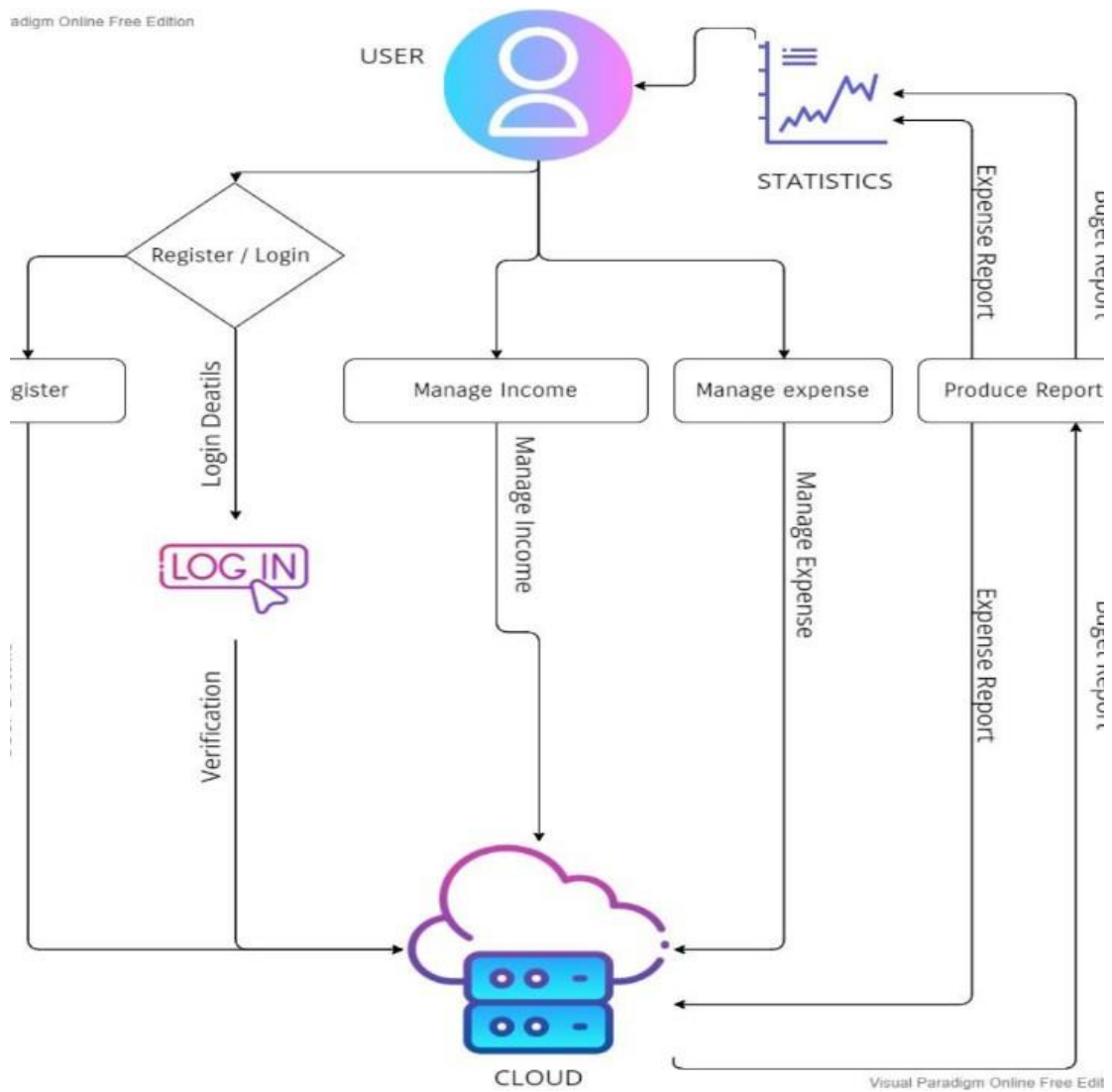
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Effectiveness, efficiency, and overall experience of the user interacting with our application should be maximised.
NFR-2	Security	Authentication, authorization, encryption of the data must be done in the application.
NFR-3	Reliability	Probability of error in the operations in a specified environment for a specified time should be minimised.
NFR-4	Performance	How the application is functioning accurately and effectively the application is to the end-users.
NFR-5	Availability	Using Cloud Storage and database, application reliability and the user satisfaction will affect the solution
NFR-6	Scalability	Capacity of the application to handle growth, especially in handling more users.

4.2 FUNCTIONAL REQUIREMENT:

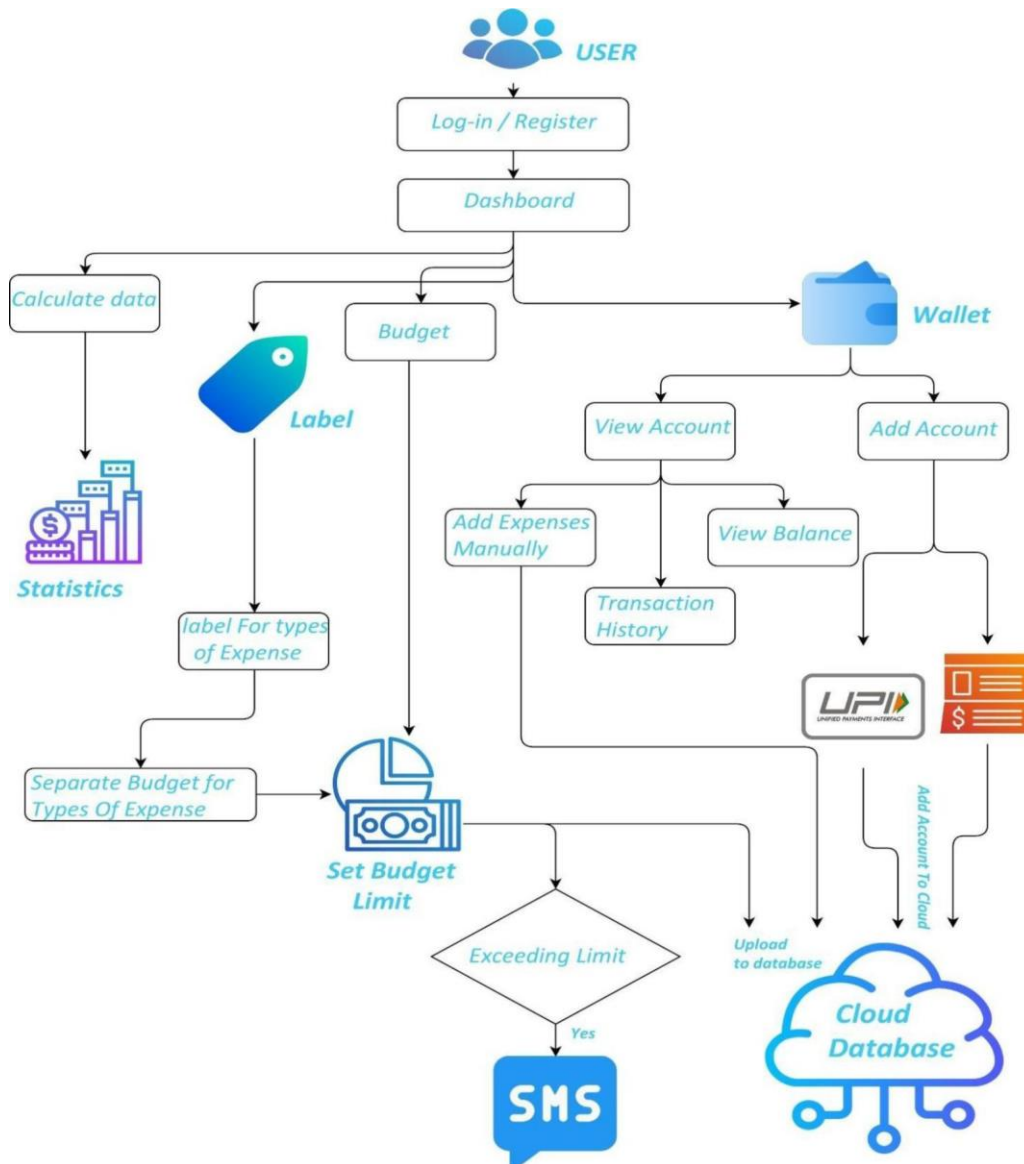
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Data	Data gathered in the application server is saved in the high security cloud server.
FR-4	Alert Notification	Alert messages through the Email or SMS.
FR-5	User Monthly Budget Plan	Setting Monthly budget to manage their expenses.
FR-6	Cloud Data Storage	To save the user valuable data high security cloud storage are used (AWS, IBM, GOOGLE, etc)

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM:



5.2 SOLUTION & TECHNICAL ARCHITECTURE:



5.3 USER STORIES:

User Stories

Use the below template to list all the user stories for the product.

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 entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-2	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard		As a user, I can access my detail, manage the expense, add budget, expense report from the app etc..		High	Sprint-1
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-3	As a user, I can access my detail, manage the expense, add budget, expense report from the app etc..		High	Sprint-1
Customer Care Executive	Email or Customer Care no		As a user, I can contact the service administration for the support.	I can solve the Issue.	High	Sprint-3
Administrator	Email or Customer Care no		As a user, I can contact the service administration for the support.	I can solve the Issue.	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

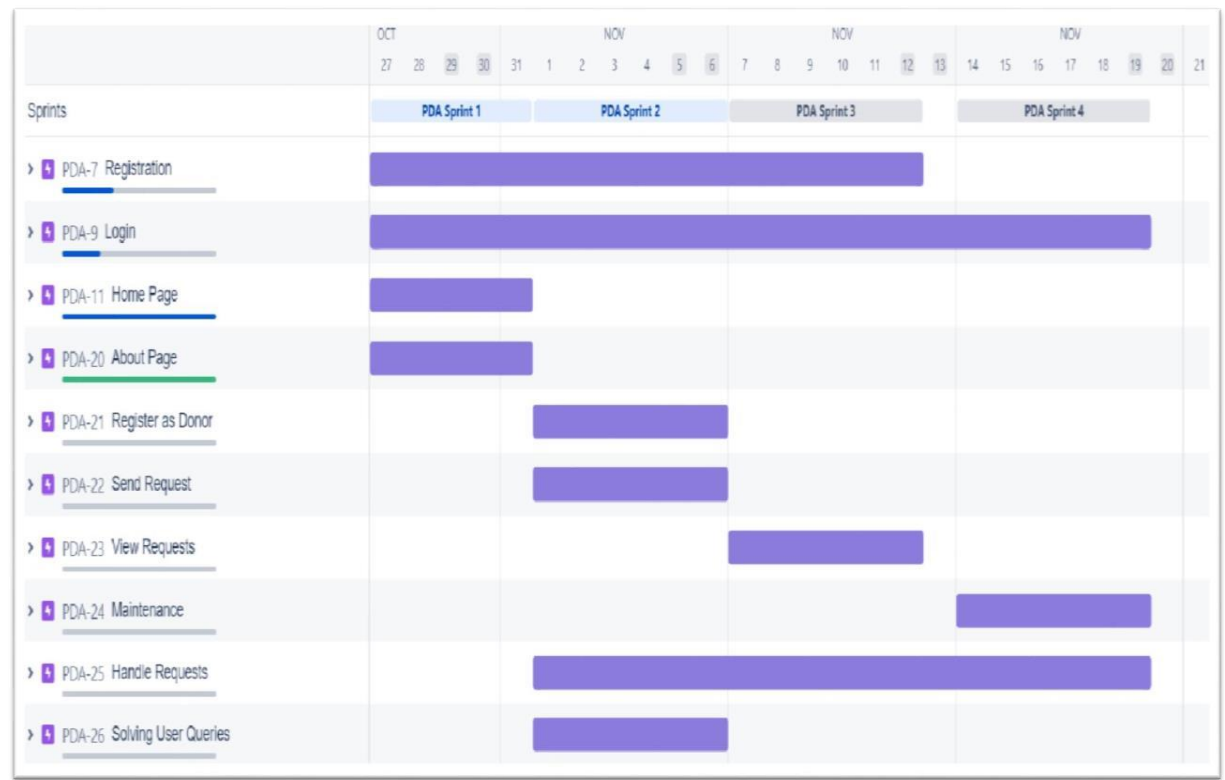
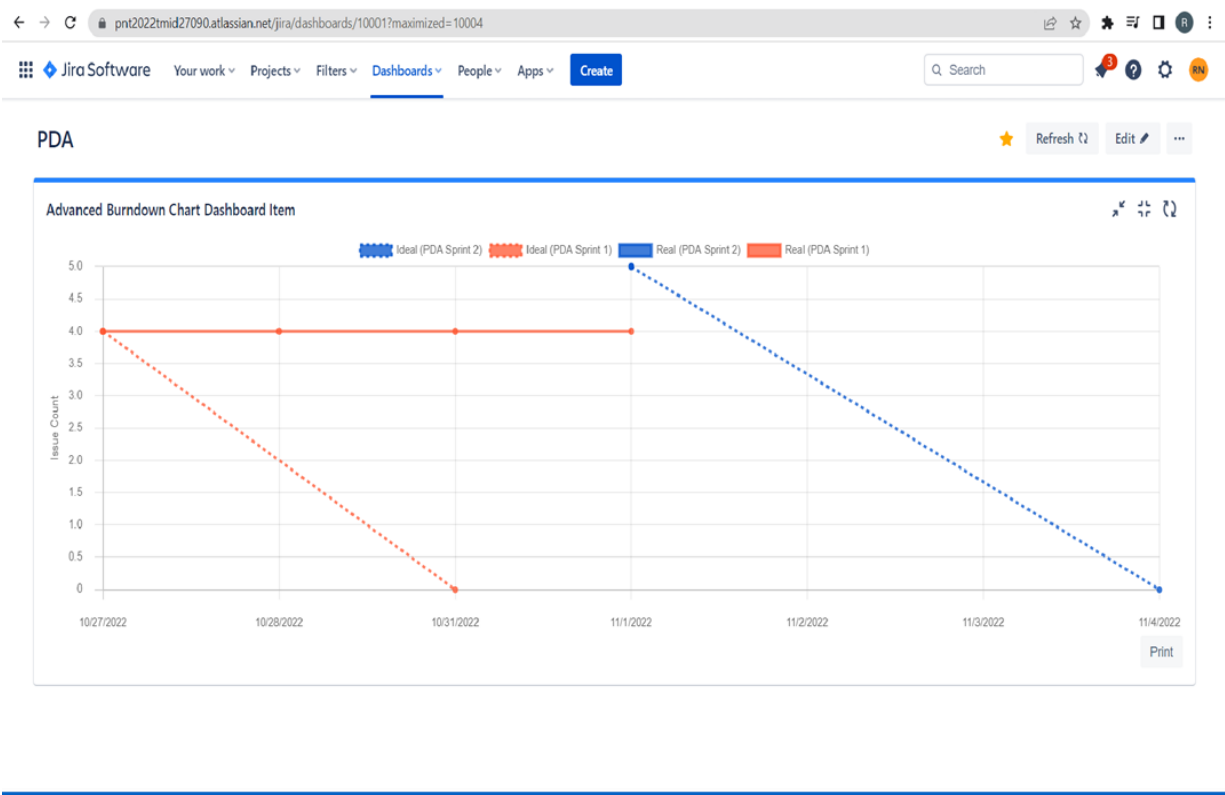
6.1 SPRINT PLANNING & ESTIMATION :

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	Harish Kumar R
Sprint-2		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Dhanush S
Sprint-3		USN-3	As a user, I can register for the application through Facebook	2	Low	Jayaraj S
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Kaarthick S
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Aditi S
	Dashboard	USN-6	As a user, I access my detail, manage my expense, add budget, expense report from the app etc..	2	High	Jayaraj S

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	25 Oct 2022	30 Oct 2022	20	30 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	18	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	15	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	19	21 Nov 2022

6.3 REPORTS FROM JIRA :



7. CODING & SOLUTIONING

7.1 FEATURE 1:

Python

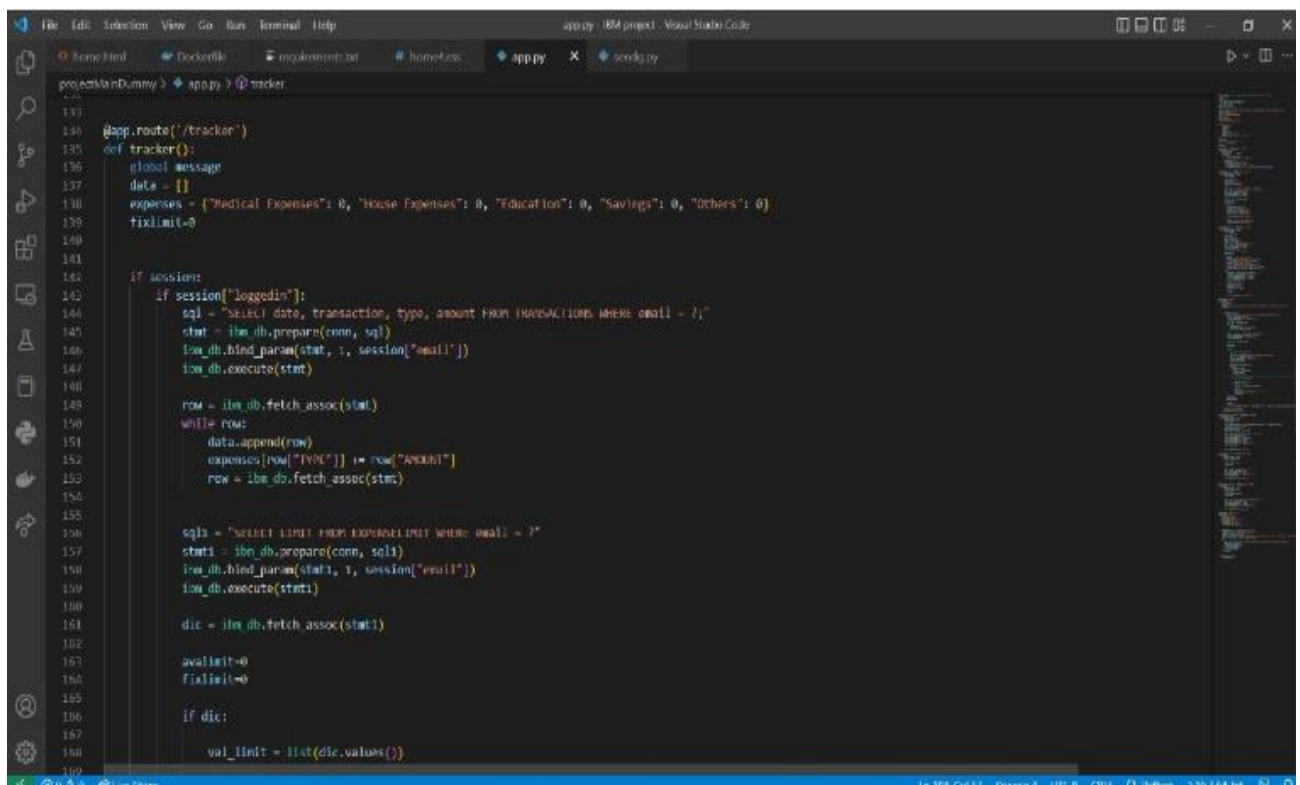
It is a **high-level**, **general-purpose programming language**. Its design philosophy emphasizes **code readability** with the use of **significant indentation**.

Python is **dynamically-typed** and **garbage-collected**.

It supports multiple, including **structured** (particularly **procedural**), **object-oriented** and **functional programming**.

It is often described as a "batteries included" language due to its comprehensive **standard library**

OUTPUT SCREEN :



```
133
134 @app.route('/tracker')
135 def tracker():
136     global message
137     data = []
138     expenses = {"Medical Expenses": 0, "House Expenses": 0, "Education": 0, "Savings": 0, "Others": 0}
139     txlimit=0
140
141
142     if session["loggedin"]:
143         if session["loggedin"]:
144             sql = "select date, transaction, type, amount from transactions where email = ?;"
145             stmt = ibm_db.prepare(conn, sql)
146             ibm_db.bind_param(stmt, 1, session["email"])
147             ibm_db.execute(stmt)
148
149             row = ibm_db.fetch_assoc(stmt)
150             while row:
151                 data.append(row)
152                 expenses[row["TYPE"]] += row["Amount"]
153                 row = ibm_db.fetch_assoc(stmt)
154
155             sql1 = "select limit from expenselimit where email = ?"
156             stmt1 = ibm_db.prepare(conn, sql1)
157             ibm_db.bind_param(stmt1, 1, session["email"])
158             ibm_db.execute(stmt1)
159
160             dic = ibm_db.fetch_assoc(stmt1)
161
162             availimit=0
163             fixlimit=0
164
165             if dic:
166                 val_limit = list(dic.values())
```

7.2 FEATURE 2:

Flask

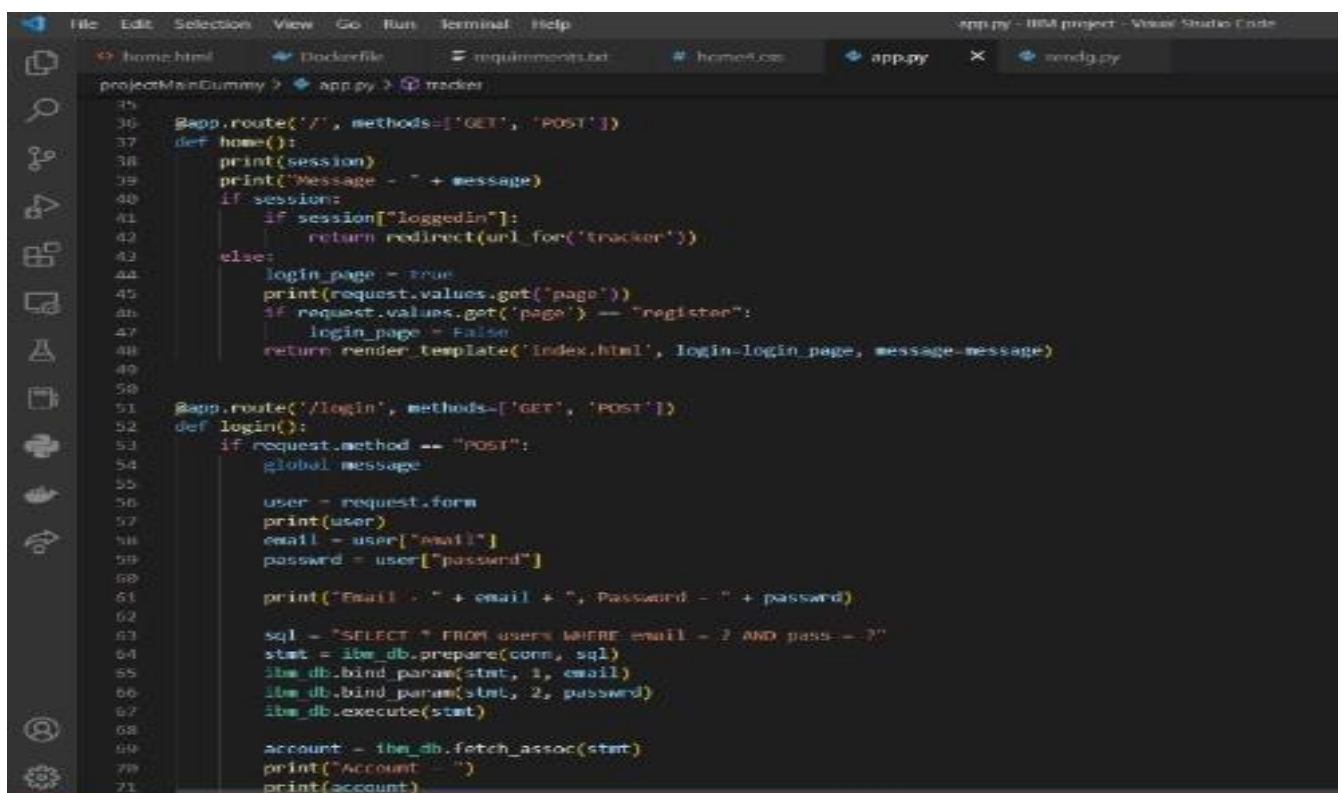
Flask is a micro [web framework](#) written in [Python](#). It is classified as a [micro_framework](#) because it does not require particular tools or libraries.^[2]

It has no [database](#) abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

Extensions exist for [object-relational mappers](#), form validation, upload handling, various open authentication technologies and several common framework related tools.

OUTPUT SCREEN :



```
15
16 @app.route('/', methods=['GET', 'POST'])
17 def home():
18     print(session)
19     print("Message - " + message)
20     if session:
21         if session["loggedin"]:
22             return redirect(url_for('tracker'))
23     else:
24         login_page = True
25         print(request.values.get('page'))
26         if request.values.get('page') == "register":
27             login_page = False
28         return render_template('index.html', login=login_page, message=message)
29
30
31 @app.route('/login', methods=['GET', 'POST'])
32 def login():
33     if request.method == "POST":
34         global message
35
36         user = request.form
37         print(user)
38         email = user["email"]
39         password = user["password"]
40
41         print("Email - " + email + ", Password - " + password)
42
43         sql = "SELECT * FROM users WHERE email = ? AND pass = ?"
44         stmt = ibm_db.prepare(conn, sql)
45         ibm_db.bind_param(stmt, 1, email)
46         ibm_db.bind_param(stmt, 2, password)
47         ibm_db.execute(stmt)
48
49         account = ibm_db.fetch_assoc(stmt)
50         print("Account - ")
51         print(account)
```

7.3 DATABASE SCHEMA:

IBM Db2 –

A Hybrid ANSI-compliant data virtualization tool for accessing, querying and summarizing data across the enterprise which:

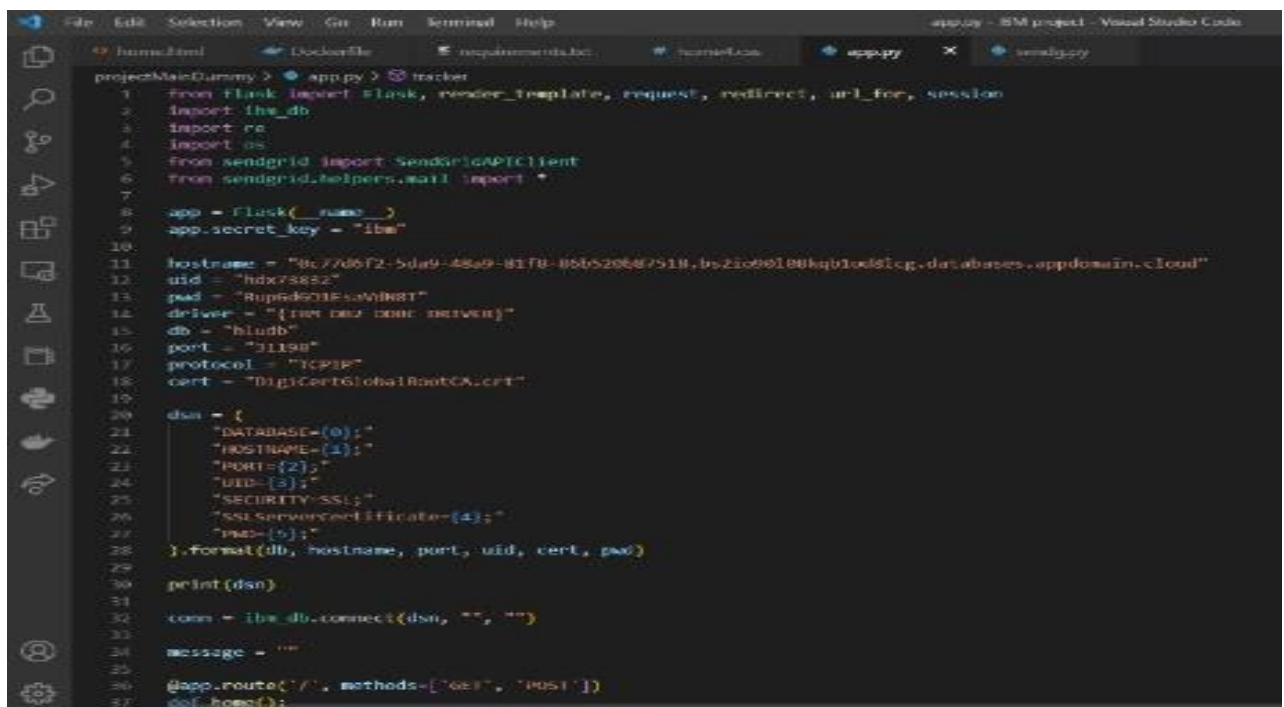
Provides a massively parallel processing (MPP) architecture Exploits Hive, HBase and Apache Spark concurrently for best-in-class analytic capabilities

Requires only a single database connection or query to connect disparate sources such as HDFS, RDMS, NoSQL databases, object stores and Web HDFS

Provides low latency support for ad-hoc and complex queries, high performance, and federation capabilities

Understands dialects from other vendors and various products from Oracle, IBM® Db2® and IBM Netezza®

OUTPUT SCREEN :



```
projectName: dummy > app.py > tracker
1  from flask import Flask, render_template, request, redirect, url_for, session
2  import ibm_db
3  import re
4  import os
5  from sendgrid import SendGridAPIClient
6  from sendgrid.helpers.mail import *
7
8  app = Flask(__name__)
9  app.secret_key = "ibm"
10
11  hostname = "8c77d6f2-5da9-48a9-81f8-85b520b87518.bs2ic06l80kqblod8lqg.databases.appdomain.cloud"
12  uid = "hdx78852"
13  pw = "Rup6d501EsaVn8RT"
14  driver = "{(IBM DB2) IBM DB2 DRIVER}"
15  db = "hludb"
16  port = "31198"
17  protocol = "TCP/IP"
18  cert = "DigitCertGlobalRootCA.crt"
19
20  dsn = {
21      "DATABASE={0};".format(db),
22      "HOSTNAME={1};".format(hostname),
23      "PORT={2};".format(port),
24      "UID={3};".format(uid),
25      "SECURITY=SSL;".format("SSL ServerCertificate={4};".format(cert)),
26      "PWD={5};".format(pw)
27  }.format(db, hostname, port, uid, cert, pw)
28
29  print(dsn)
30
31  conn = ibm_db.connect(dsn, "", "")
32
33  message = ""
34
35  @app.route('/', methods=['GET', 'POST'])
36  def home():
37
```

SENDGRID :

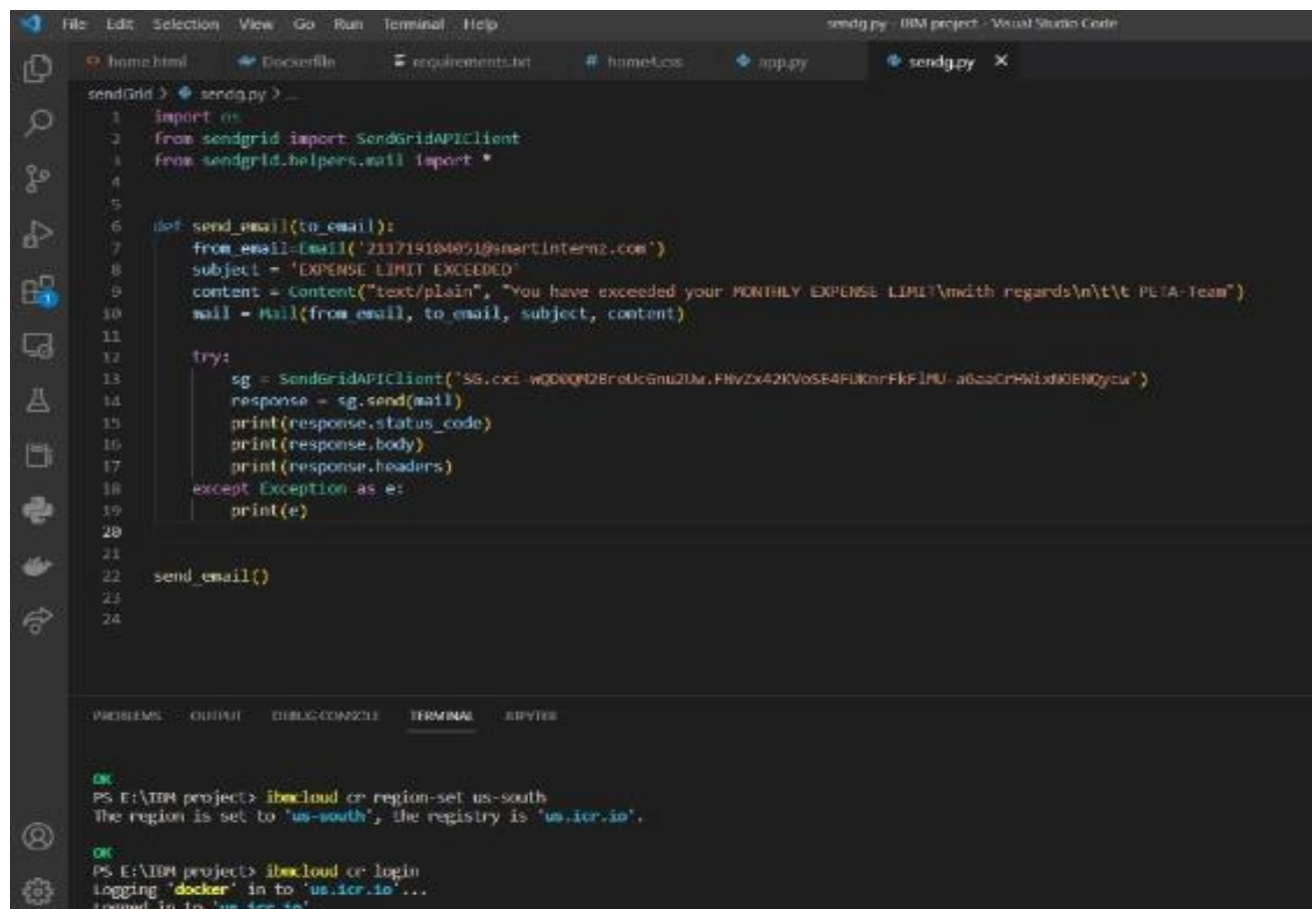
SendGrid (**also known as** Twilio SendGrid) is a [Denver, Colorado](#)-based customer communication platform for transactional and marketing [email](#).^{[3][4][5][6][7][8]}

The company was founded by Isaac Saldana, Jose Lopez, and Tim Jenkins in 2009, and incubated through the [Techstars accelerator](#) program.¹

SendGrid provides two ways to send email: through our SMTP relay or through our Web API.

SendGrid provides client libraries in many languages. This is the preferred way to integrate with SendGrid.

OUTPUT SCREEN :



```
File Edit Selection View Go Run Terminal Help
sendgrid.py - IBM project - Visual Studio Code
home.html Dockerfile requirements.txt # home-tux app.py sendgrid.py x

sendGrid > sendgrid.py > _
1 import os
2 from sendgrid import SendGridAPIClient
3 from sendgrid.helpers.mail import *
4
5
6 def send_email(to_email):
7     from_email=Email('211719164851@smartintern2.com')
8     subject = 'EXPENSE LIMIT EXCEEDED'
9     content = Content("text/plain", "You have exceeded your MONTHLY EXPENSE LIMIT\ndwith regards\n\t\t PETA-Team")
10    mail = Mail(from_email, to_email, subject, content)
11
12    try:
13        sg = SendGridAPIClient('SG.cxl-wQdQGN28reUcGnu2Uw.FHVZx42KV0SE4FkKnfFkF1MU_a6oaCRHWLxdK0EN0ycu')
14        response = sg.send(mail)
15        print(response.status_code)
16        print(response.body)
17        print(response.headers)
18    except Exception as e:
19        print(e)
20
21
22 send_email()
23
24
```

```
OK
PS E:\IBM project> ibmcloud cr region-set us-south
The region is set to 'us-south', the registry is 'us.icr.io'.

OK
PS E:\IBM project> ibmcloud cr login
Logging 'docker' in to 'us.icr.io'...
logged in to 'us.icr.io'.
```

8. TESTING

8.1 TESTING CASE:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.

It provides a way to check the functionality of your components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectation and does not fail in an unacceptable manner.

There are various types of test. Each test type addresses a specific testing requirement.

8.2 ACCEPTANCE TESTING

UAT Execution & Report Submission

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Plasma Donor Application](#) 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	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

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
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS:

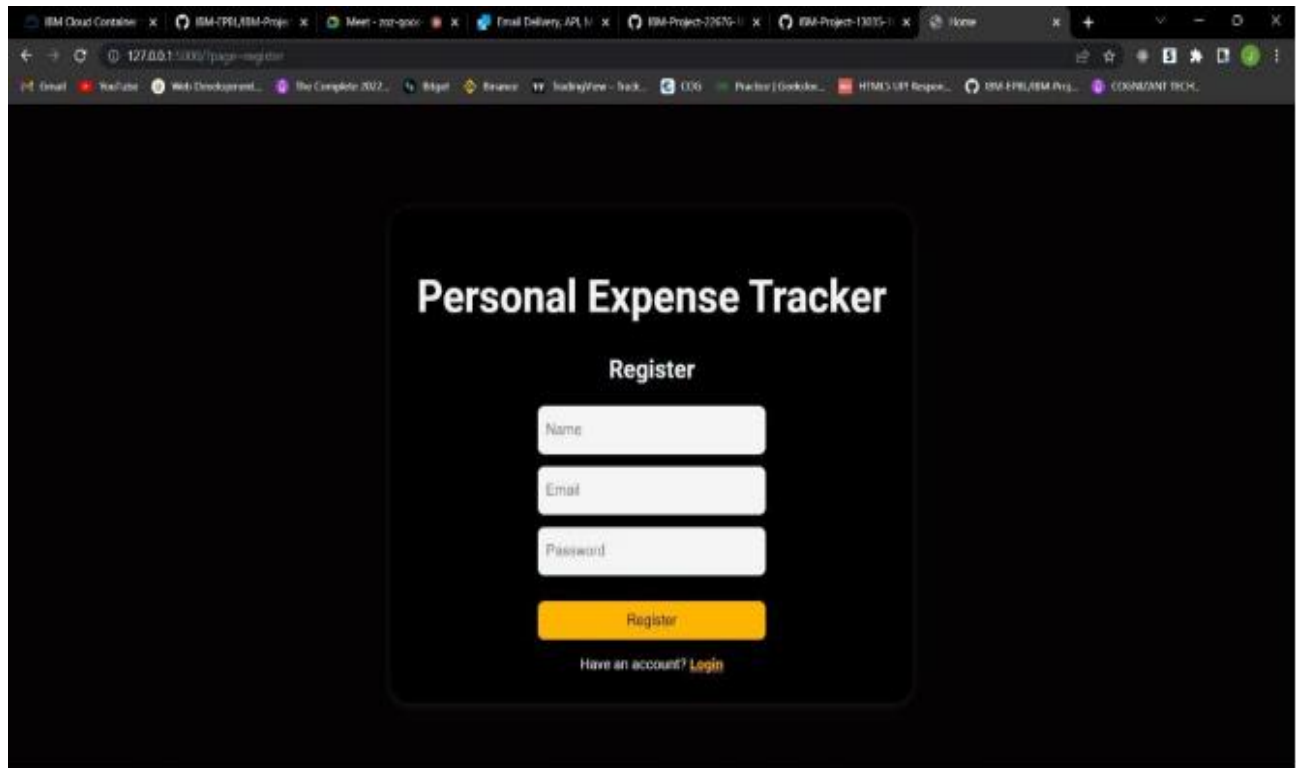
Project Metrics are used to track the progress and performance of a project.

Monitoring the parts of a project like Productivity, Scheduling and scope make it easier for team leaders to see what's on track.

As a project evolves, managers needs access to changing deadlines or budgets to meet their client expectations

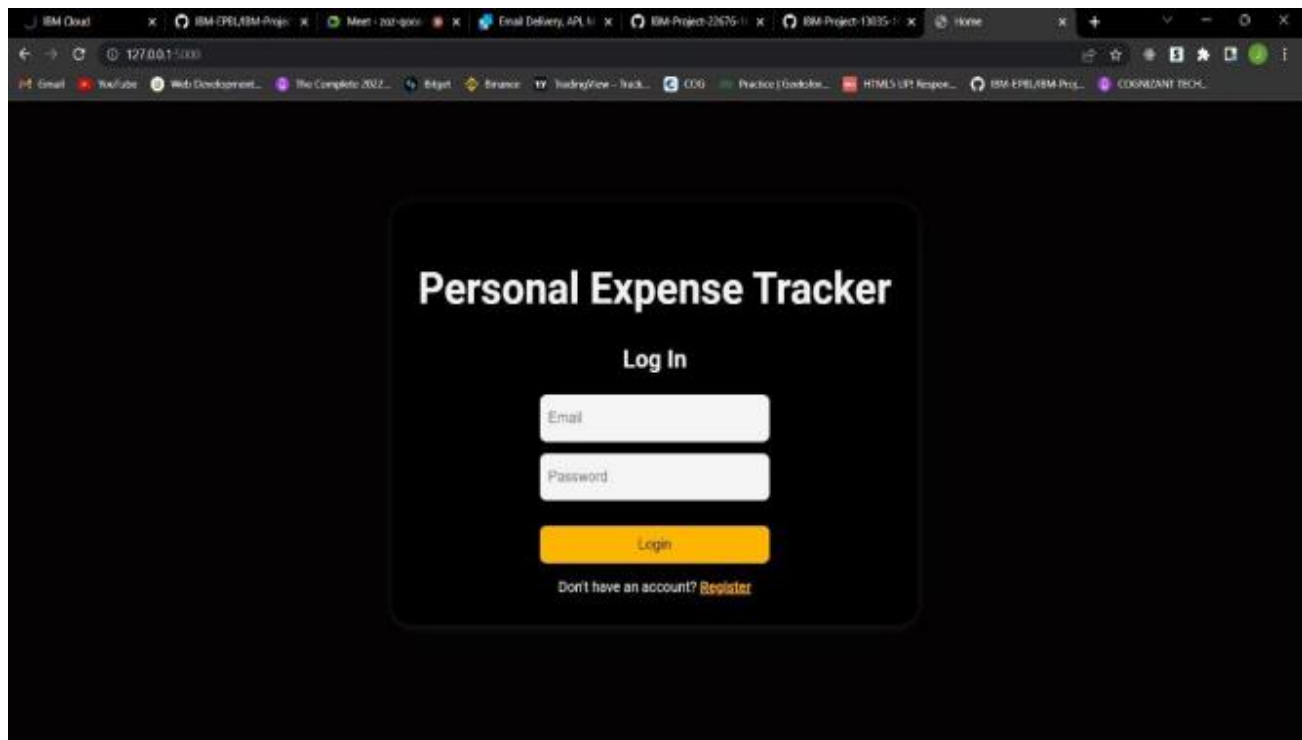
OUTPUT SCREENS:

Registration



A screenshot of a web browser displaying the registration page for a 'Personal Expense Tracker'. The browser's address bar shows the URL '127.0.0.1:5000/page=register'. The page has a dark background with a central white rounded rectangle containing the title 'Personal Expense Tracker' in bold. Below the title is the heading 'Register'. There are three white input fields labeled 'Name', 'Email', and 'Password'. Below these fields is a yellow 'Register' button. At the bottom of the form, there is a link that says 'Have an account? [Login](#)'.

Login



A screenshot of a web browser displaying the login page for a 'Personal Expense Tracker'. The browser's address bar shows the URL '127.0.0.1:5000'. The page has a dark background with a central white rounded rectangle containing the title 'Personal Expense Tracker' in bold. Below the title is the heading 'Log In'. There are two white input fields labeled 'Email' and 'Password'. Below these fields is a yellow 'Login' button. At the bottom of the form, there is a link that says 'Don't have an account? [Register](#)'.

Dashboard

The screenshot shows a web application titled "Personal Expense Tracker" running in a browser. The browser's address bar shows the URL "127.0.0.1:5000/track". The dashboard has a dark theme and includes a sidebar on the left with three categories: "Medical Expenses" (₹0), "House Expenses" (₹13500), and "Education Expenses" (₹0). The main content area is divided into three sections. The "Monthly Limit" section displays "Expense Limit : 200000" and "Available Limit : 46500", with input fields for "Set Amount" and "Change Amount", and buttons for "Set Limit" and "Change Limit". The "Add Expenditure" section contains input fields for "Transaction", "Transaction Type", "Amount", and "mm/dd/yyyy", along with an "Add" button. The top navigation bar includes links for "Dashboard", "Statistics", and "Logout".

Personal Expense Tracker

Welcome, Jayaraj

Medical Expenses
₹0

House Expenses
₹13500

Education Expenses
₹0

Monthly Limit

Expense Limit : 200000

Available Limit : 46500

Set Amount

Change Amount

Set Limit

Change Limit

Add Expenditure

Transaction

Transaction Type

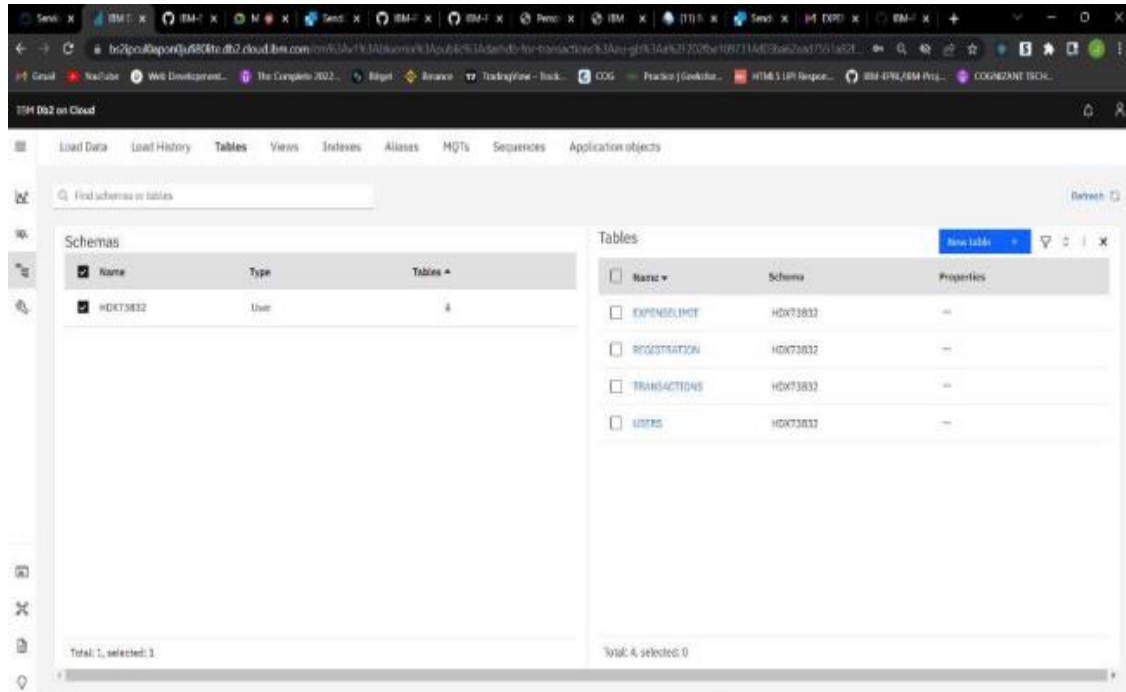
Amount

mm/dd/yyyy

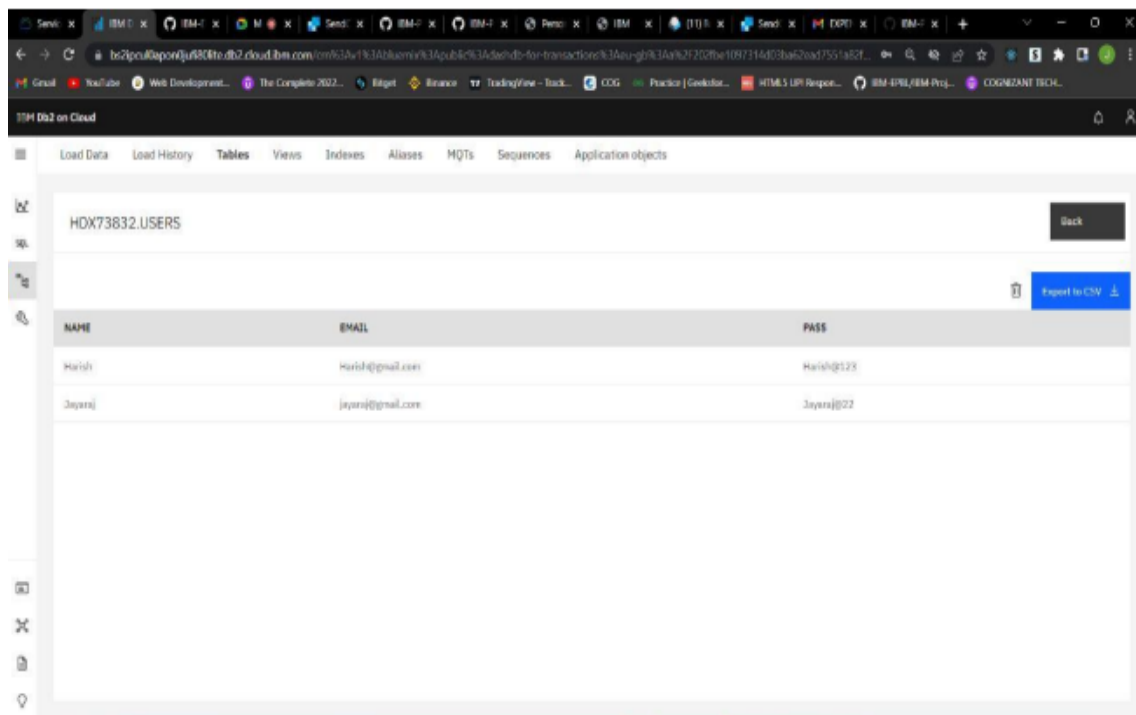
Add

Databases

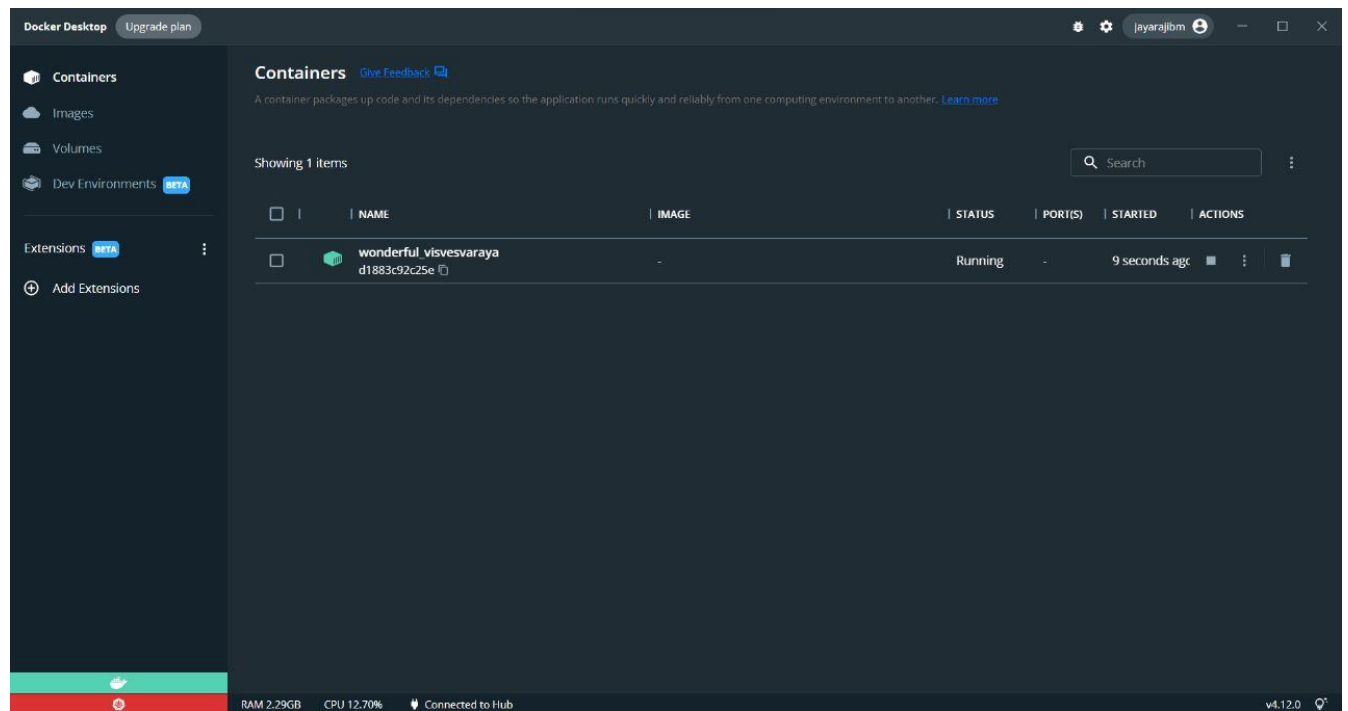
1.Tables



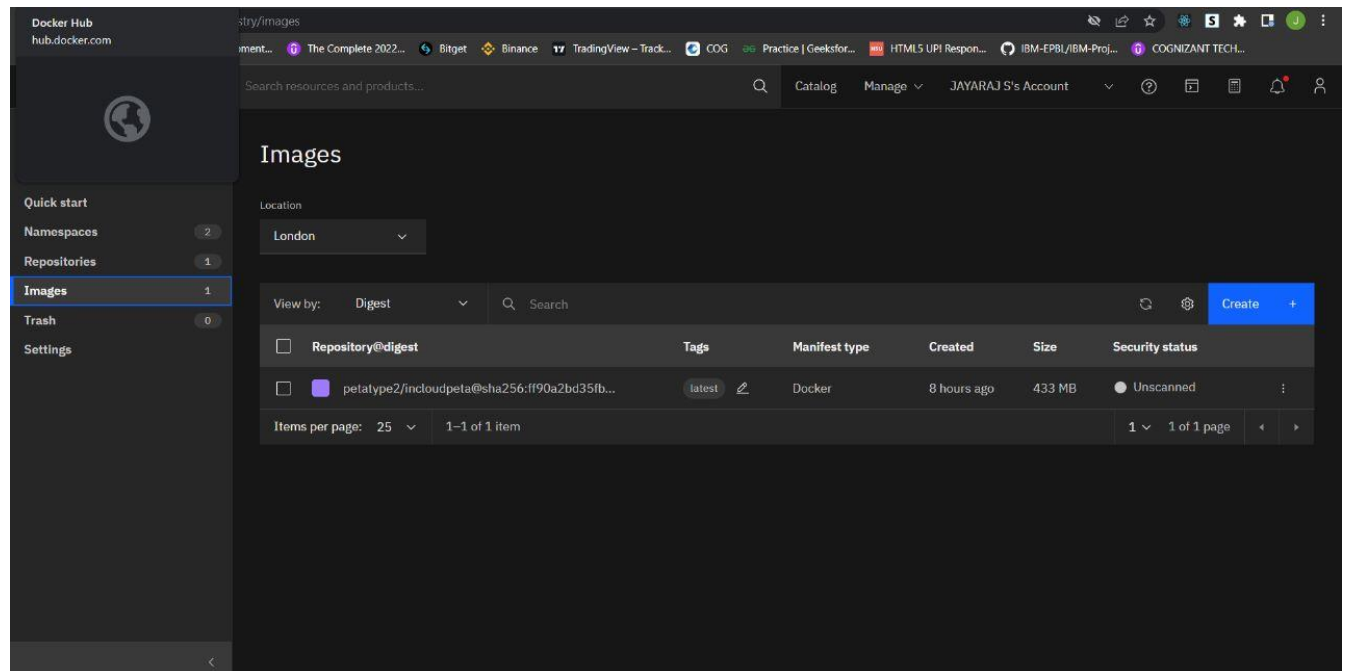
2.Data



Docker Container :



IBM Container Registry :



Cloud Registry Image Details:

The screenshot displays the IBM Cloud Container Registry interface. The browser address bar shows the URL: `cloud.ibm.com/registry/images/eyJyZX8vYjoidWsuYWNYLmV3BldGF0eXBIMi9pbmNsb3VkcGV0YSlslmxvbmddEaWdlc3QlOiZaGyNTY6ZmY5MGYyYmQzNWZlNTk1OTgyNjdiN2MzZTlmMjBIM...`. The page title is "incloudpeta@sha256:ff90a2bd35fb...". The image details section shows the following information:

Tags	Digest	Size	Created	Security status
latest	ff90a2bd35fb...	433 MB	11 மார்ச் 2022	261 issues

The "Image inspect data" section displays the following JSON output:

```
{
  "Id": "sha256:4133d5ea67b71bacdf4a81da22281ef4e7b2ab7db4faecf97b819525eaedd9",
  "Parent": "",
  "Comment": "",
  "Created": "2022-11-18T07:10:53.204078736Z",
  "Container": "",
  "ContainerConfig": {
    "Hostname": "",
    "Domainname": "",
    "User": "",
    "AttachStdin": false,
    "AttachStdout": false,
    "AttachStderr": false,
    "Tty": false,
    "OpenStdin": false,
    "StdinOnce": false,
    "Env": null,
    "Cmd": null,
    "Image": "",
    "Volumes": {},
    "WorkingDir": "",
    "Entrypoint": null,
    "OnBuild": null,
    "Labels": null
  },
  "DockerVersion": "",
  "Author": "",
  "Config": {
    "Hostname": "",
    "Domainname": "",
    "User": "",
    "AttachStdin": false,
    "AttachStdout": false,
    "AttachStderr": false,
    "Tty": false,
    "OpenStdin": false,
    "StdinOnce": false,
    "Env": null,
    "Cmd": null,
    "Image": "",
    "Volumes": {},
    "WorkingDir": "",
    "Entrypoint": null,
    "OnBuild": null,
    "Labels": null
  }
}
```

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

Speed: This website is fast and offers great accuracy as compared to manual registered keeping.

Maintenance : Less maintenance is required

User Friendly : It is very easy to use and understand. It is easily workable and accessible for everyone.

DISADVANTAGES:

Internet: It requires internet connection for the working of the website.

Auto- Verification: It cannot automatically verify the genuine Users

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.

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

13. APPENDIX

GITHUB :

<https://github.com/IBM-EPBL/IBM-Project-22676-1659856060>

PROJECT DEMO LINK :

https://drive.google.com/file/d/19413mewh27U1xuDymwHsjuubnYvwqbYo/view?usp=share_link