.

Personal Expense Tracker Application

A PROJECT REPORT

Submitted by:

VIJAY S (210419205059)

DHINAKARAN V (210419205010)

PUGALARASAN G (210419205036)

NAVEEN K M (210419205033)

Index

1 INTRODUCTION

Who doesn't have a goal for the new year that involves money? Many of us have goals that involve making more money or managing the money we already have — but, no matter exactly what goal you might have for your money, you'll probably need some baseline information about it. While understanding your expenses is basic, they make up some of the most important information you can gather about where your money goes. Tracking expenses can be a relatively simple matter and can provide you so much information about your spending habits.

You might get a little more information from other expense tracking systems (listing them in a spreadsheet, using money management software or even choosing an online application), but all methods have one thing in common: you have to get in the habit of thinking about your expenses.

It's very easy to misplace a receipt or forget about any cash you spent. You may even think that a cup of coffee or a trip to the vending machine isn't worth tracking — although those little expenses can add up amazingly fast.

1.1 Project Overview

Whether you're working on creating a budget or you are trying to simplify the bookkeeping for a small business, tracking your expenses should be a first step. If most of your spending is done electronically (using a debit card or a credit card), you may be able to get away with just tracking your cash spending. Most money management software can automatically import those electronic expenses, further simplifying matters. You can also choose to use your own system, from the ground up, including setting up a spreadsheet and entering information by hand.

1.2 Purpose

It's also worthwhile to track your income in the same system that you track your expenses. This may seem like a no-brainer, because many people think that they only have one source of income: the salary that they receive from their job. In truth, however, most of us have additional sources of money, whether we hold a yearly garage sale, freelance or receive rebates. If you choose an application specifically created to track expenses, you'll find that most have some sort of tool for inputting information about your income as well. If you decide to use a system of your own devising, such as a list of expenses in a spreadsheet, you'll need to clearly separate income and expenses — place them in different columns, make one negative or denote the difference in another way.

2 LITERATURE SURVEY

S.No.	Author	Paper title	Journal / Conference Title	Page No. & Volume No.	Year of Publication	Description
1	Muskaan Sharma, Ayush Bansal, Dr.Raju Ranjan, Shivam Sethi	A Novel Expense Tracker using Statistical Analysis	INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY	Pages 154-158, Volume 8 Issue 1.	Jun 2021	Expense Tracker' to efficiently manage house-old budget. Our system will allow user to keep track of their expenses. Some statistical analysis has to be done to be able to give users correct information on their expenses and help them spend better.
2	Sabab, S. A., Islam, S. S., Rana, M. J., & Hossain,M.	A smart approach to track everyday expense.	4 th International Conference on Electrical Engineering and Information & Communication Technology	Pages 136 – 141	September 2018	Tracking regular expense is a key factor to maintain a budget. People often track expense using pen and paper method or take notes in a mobile phone or a computer. These processes of storing expense require further computations and processing for these data to be used as a trackable record. In this work, we are proposing an automated system named as eExpense to store and calculate these data.

S.No.	Author	Paper title	Journal / Conference Title	Page No. & Volume No.	Year of Publication	Description
3	Aman Garg , Mukul Goel, Sagar Mittal, Mr. Shekhar Singh.	Expense Tracker	International Journal for Research in Applied Science & Engineering Technology (IJRASET)	Volume 9	April 2021	This Expense Tracker is a web application that facilitates the users to keep track and manage their personal as well as business expenses. This application helps the users to keep a digital diary. It will keep track of a user's income and expenses on a daily basis. The user will be able to addhis/her expenditures instantly and can review them anywhere and anytime with the help of the internet. He/she can easily import transactions from his/her mobile wallets without riskinghis/her information and efficiently protecting his/her privacy
4	ATIYA KAZI , PRAPHULLA S. KHERADE,RAJ S. VILANKAR , PARAG M. SAWANT	Expense Tracker	ICONIC RESEARCH AND ENGINEERING JOURNALS	Volume 4 Issue 11	May 2021	We are building an android application named as "Expense Tracker". As the name suggests, this project is an android app which is used to track the daily expenses of the user. It is like digital record keeping which keeps the records of expenses done by a user. The application keeps the track of the Income and Expenses both of user on a day-to-day basis

2.1 Existing problem

1. In the existing system the information are less secure. The information are used and sold to third party applications.
2. In the existing system there is no cloud services so that we cannot access our data all over the world.

3. In the existing system there is no traffic control on browser so that if multiple users are using the application on the same time there is chance of malfunction and reduce the performance of the mobile or PC.

2.2 References

http://expense-manager.com/how-expensesoftware/

https://www.splitwise.com/terms

http://code.google.com/p/socialauth-android/wiki/Facebook

http://code.google.com/p/socialauth-android

Developer.android.com

http://www.appbrain.com/app/expensemanager/com.expensemanager

https://www.xpenditure.com/en?

http://expense-manager.com/how-expensesoftware/

Donn Felker, “Android Application Developmentfor Dummies”, published by For Dummies, 2010.

Lee, “Beginning Android Application Development”, Published by WroxPress, 2011.

2.3 Problem Statement Definition

The problem of current generation population is that they can’t remember where all of the money they earned have gone and ultimately have to live while sustaining the little money they have left for their essential needs. In this time there is no such perfect solution which helps a person to track their daily expenditure easily and efficiently and notify them about the money shortage they have. For doing so they have to maintain long ledger’s or computer logs to maintain such data and the calculation is done manually by the user, which may generate error leading to losses. Not having a complete tracking system, generates a regular need of entering daily data of the expenditure and total estimation till the end of month.

3 IDEATION & PROPOSED SOLUTION

The expense tracker is an application which runs on all platforms. It helps user to manage all their expenses in an efficient way and track budget. This would avoid budget handling difficulties and gives us efficient results on our savings. In everyone’s life, money plays an important role. A person who cannot manage his expenses cannot successfully lead a household and fulfill his goals. In the current world where mobile phones and laptops have become a part of living, such an app would be handy to deal with all our expenses. A person generally cannot keep track of all his expenses through the

traditional pen and paper method and might miss a few of his small expenditures and may even miss some bills. Such a situation will never arise when we use an app. We can make easy comparisons by seeing the graphs, which is impossible in the rigorous methods.

Who does the problem affect?	Investors, savers, big spenders, debtors ,shoppers, budget conscious consumers
What are the boundaries of the problem?	Expense tracker for working individuals, students, common people
What is the issue?	To be vigilant about the expense spent, increases financial stress. Being indecisive about the finances may result in less financial security and exceed the budget.
When does this issue occur?	When using wrong budgeting techniques. When not tracking the expenses doesn't help you to know the amount that is actually spent.
Where is the issue occurring?	Working individuals who find it difficult to track their expenses
Why is it important that we fix the problem?	Fixing this issue, brings accountability and helps to be intentional with the income by assign it to spending, saving and giving. This leads to financial stability.

3.1 Empathy Map Canvas

Build empathy and keep your focus on the user by putting yourself in their shoes

Figure 1: Empathy Map

3.2 Ideation & Brainstorming

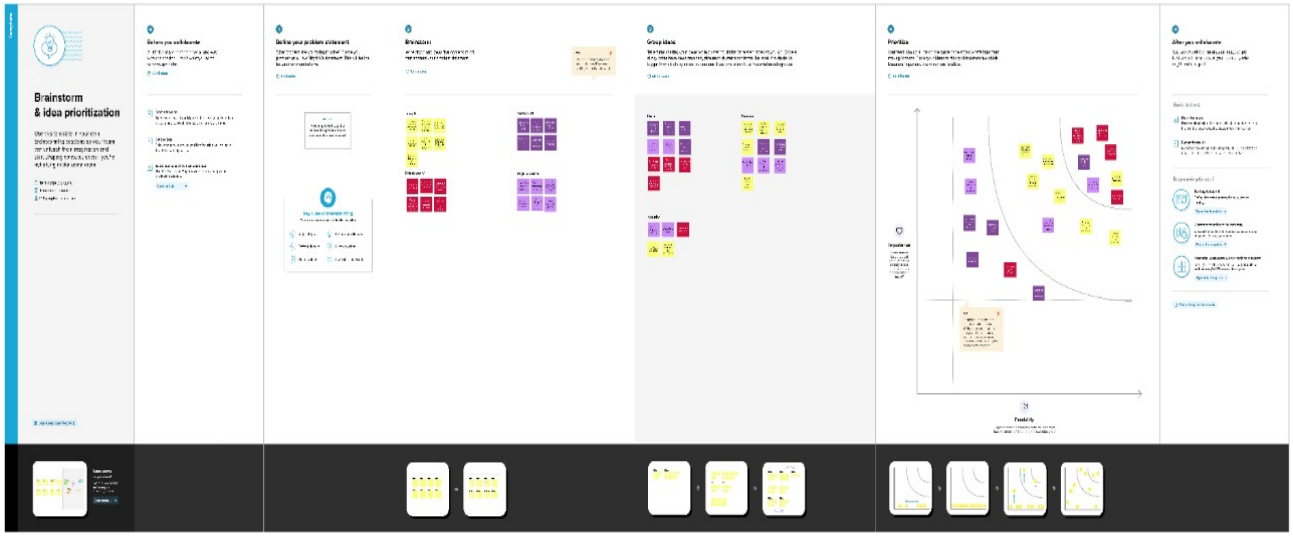


Figure 2: Brainstroming

3.3 Problem Solution fit

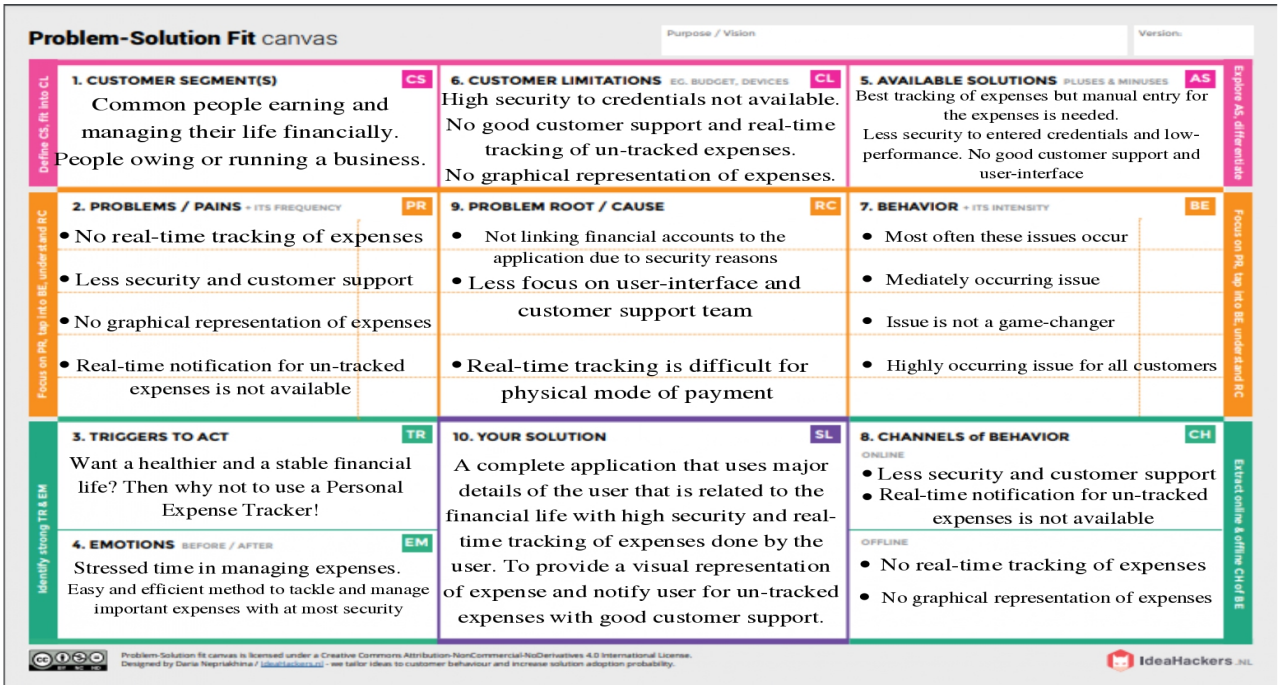


Figure 3: Problem Solution fit

3.4 Proposed Solution

SL.NO	Parameter	Description
1	Problem Statement (Problem to be solved)	The problem of current generation population is that they can't remember where all of the money they earned have gone and ultimately have to live while sustaining the little money they have left for their essential needs. In this time there is no such perfect solution which helps a person to track their daily expenditure easily and efficiently and notify them about the money shortage they have. For doing so they have to maintain computer logs to maintain such data and the calculation is done manually by the user,which may generate error leading to losses. Not having a complete tracking system,generates a regular need of entering daily data of the expenditure and total estimation till the end of month
2	Idea / Solution description	To design a web application that runs in the cloud to track user's expenses, do the calculations automatically, and produce helpful reports that are tracked with graphs and charts and visually enhanced data. It has a built-in analytical tool that can provide information about how expenses performed at a given time. Users can download the report in PDF format, business people can use the application to examine financial news and stock market activity.
3	Novelty / Uniqueness	Our cloud-based expense tracker application as the following uniqueness: • It is fast and cost-effective. • It eliminates the risk of human errors during expense calculation when manually. • It is more secure. • It offers better analytics and transparency
4	Social Impact/ Customer Satisfaction	• People can use it to track their spending and get notifications when they go over their budget's limit. • Additionally, it will provide investing features and financial news for business people and ordinary users to invest in the post office investment plan by using interest calculator. • This application can create awareness among common people about finance . This application also helps user to be financially responsible
5	Business Model (Revenue Model)	We can provide the application on a subscription and advertising basis and the cost depends on the usage.
6	Scalability of the Solution	• Instead of switching to a bigger instance size, our cloud-based application uses horizontal scalability to supply more instances, and flask micro services are utilised to improve specific functionalities. • This application can handle large number of users and data with high performance and security at any given point of time

4 REQUIREMENT ANALYSIS

Requirements analysis allows software engineers to define user needs early in the development process. It helps them deliver a system that meets customers' time, budget and quality expectations. If you are designing a system or a software program, you might want to know how to analyze its requirements properly.

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Google Registration through LinkedIn Registration through Facebook
FR-2	User Confirmation	Confirmation via Email Confirmation via Mobile OTP
FR-3	User Financial Accounts	Account Details Verification of Details Authenticate Bank Accounts
FR-4	User Dashboard	Expense Data Data Records
FR-5	User Notifications	System Access Real time Alerting Notification through Mobile and Email
FR-6	Security of User Data	Secured Database Data Security Algorithms Hashing Passwords

4.2 Non-Functional requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	By using this application, the user can keep track their expenses and can ensure that user's money is used wisely and check the bank balance and transaction history of daily usage. This Application support in all platforms and devices and Device Portability.
NFR-2	Security	Maintain user personal details in a encrypted manner by using data security algorithms . OAuth is used in this application.
NFR-3	Reliability	It helps you track all transactions like bills, refunds, payrolls, receipts, taxes, etc., on a daily, weekly, and monthly basic.
NFR-4	Performance	Its help to monitor the expense and store in the cloud. It is fast and secure with high performance Graphical interface.
NFR-5	Availability	Available 24/7 Service Support Using charts and graphs may help you monitor your Expenses, income, budgeting and assets.
NFR-6	Scalability	We can access the application without any infrastructure setup. This application is easy to install and configure in the devices. Rely on your budgeting app to track, streamline,and automate all the recurrent expenses and remind you on a timely basis. Upgrade application using customer feedback.

5 PROJECT DESIGN

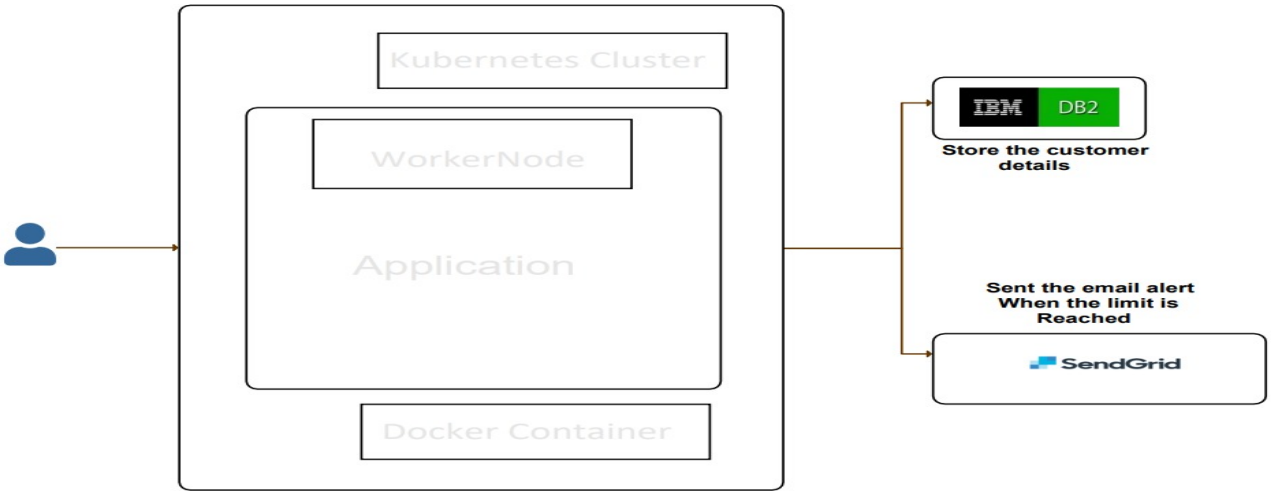


Figure 4: PROJECT DESIGN

5.1 Data Flow Diagrams

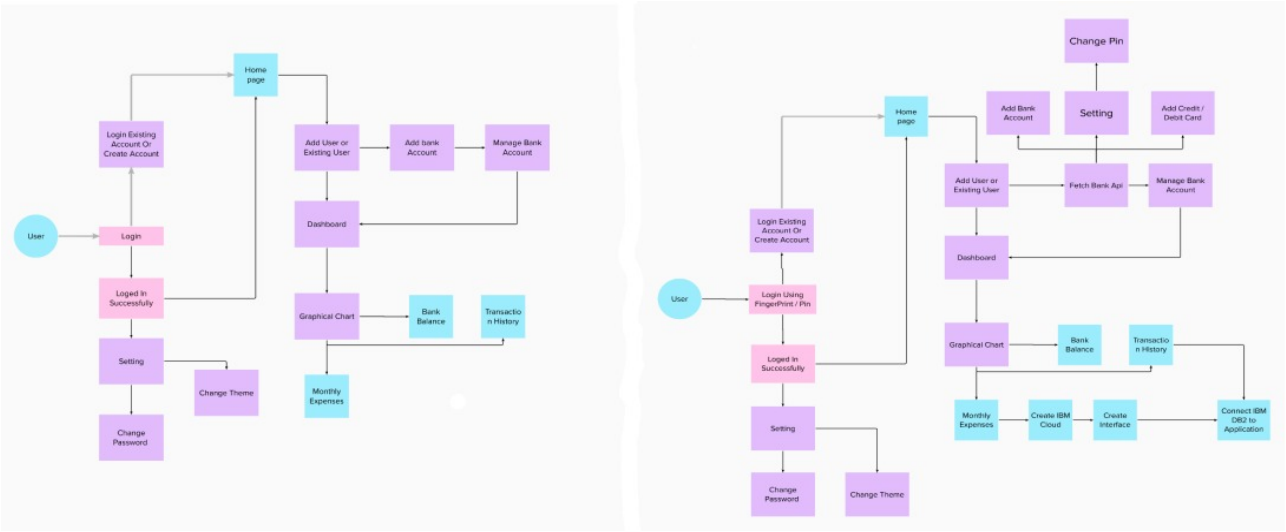


Figure 5: PROJECT DESIGN

5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

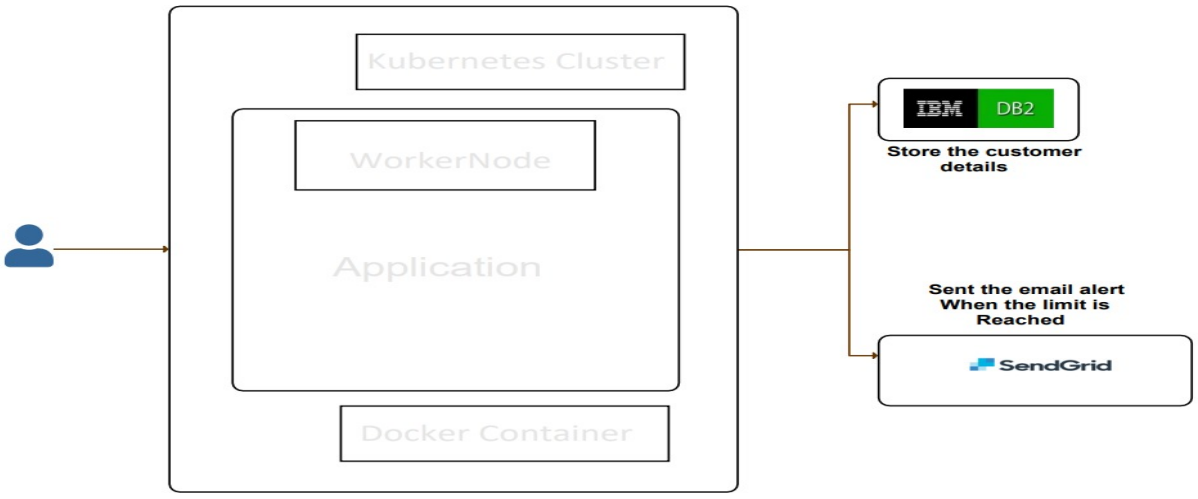


Figure 6: Solution Architecture Diagram

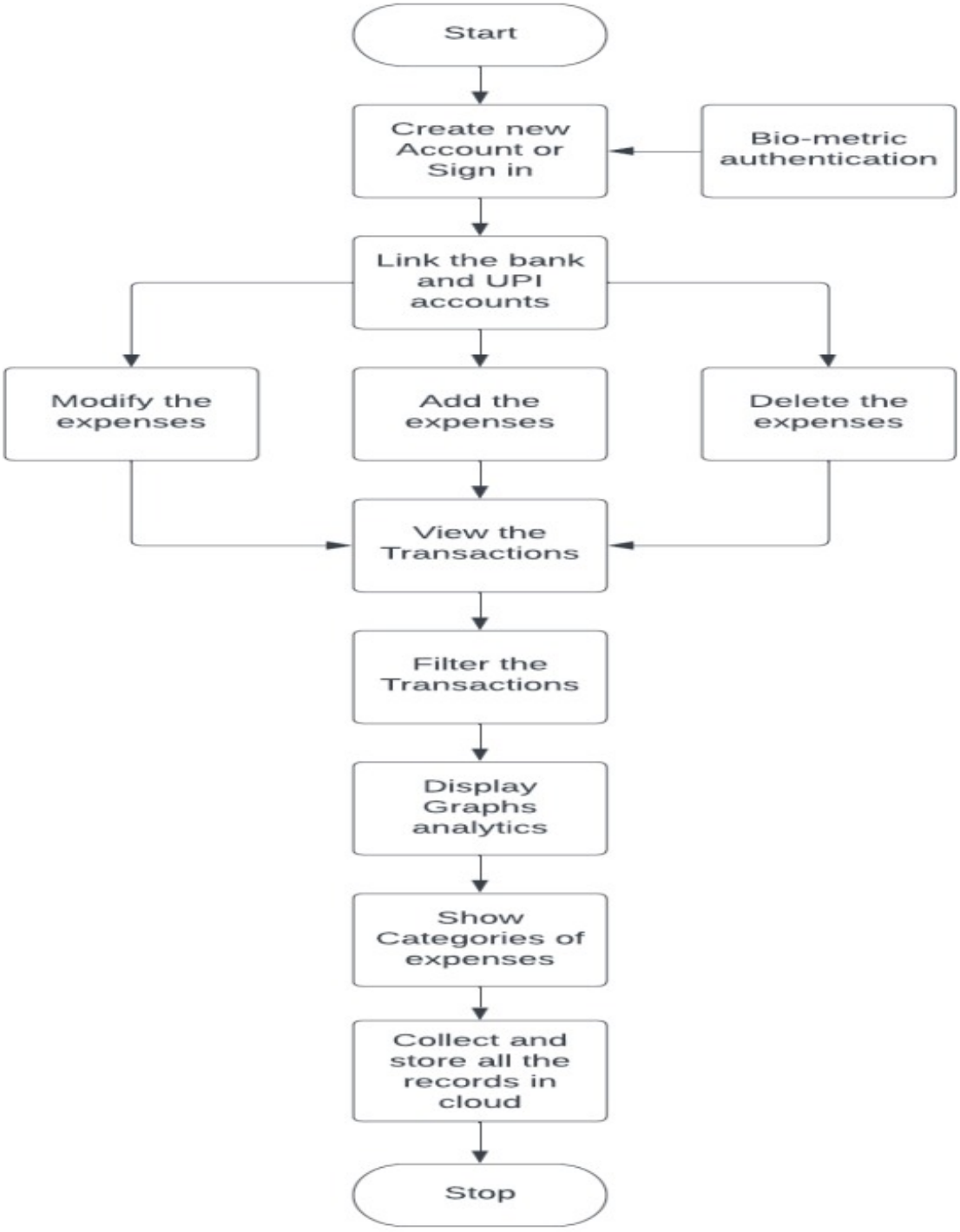


Figure 7: Flow Chart Diagram

5.3 Technical

5.3.1 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 / Angular Js / React Js etc.
2	User Enrollment	User can register themselves using their email id orany social media accounts	Python
3	Application Logic 1	Analyse the Expense of each customer.	Flask
4	Application Logic 2	Integrating the customer feed with IBM Watson to improve user experience.	IBM Watson Assistant
5	Dashboards	User can view their expenses and investments in graphical view	HTML, CSS, JavaScript, Web UIFramework, Python, Flask
6	Accounts	User can view and manage all their financialaccounts for real-time tracking of expenses	Python, Flask and trusted bankdatabases for verification
7	Notifications	Alerts and suggestions on expenses andearning/saving money techniques	InfoSphere® MDM NotificationFramework
8	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
9	File Storage	File storage requirements	IBM Object Storage or Other StorageService or Local Filesystem.
10	External API-1	Bank accounts and user verification	Bank API
11	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Docker Cloud Server Configuration : Kubernetes	Local, Cloud Foundry, Kubernetes, etc.

5.3.2 Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	A software for which the original source code is made freely available and may be redistributed and modified according to the requirement of the user.	Flask, Docker, etc.
2	Security Implementations	Cloud Security Posture Management(CSPM), Detect cloud security and compliance configuration risk, anomalous activity,	Built-in encryption, BYOK
3	Scalable Architecture	Python is one of the pioneers of programming languages that developers can use to do all the scaling work. To improve scalability, you can enable or disable services run by the dispatcher on individual servers to balance the load for a given computer by request type.	Technology used in the architecture is that with the Python and the IBM cloud.
4	Availability	Availability is the ability of a system to withstand or recover from exceptional situations, such as a computer failure. IBM Cloud is on-demand access, via the internet, to computing resources applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more hosted at a remote data centre managed by a cloud services provider (or CSP)	Technology used are the IBM cloud and the database.
5	Performance	DB2 is a database product from IBM. It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyse and retrieve the data efficiently. DB2 product is extended with the support of Object-Oriented features and non-relational structures with XML.	Technology used are the python, cloud and IBM db2.

6 PROJECT PLANNING & SCHEDULING

It is necessary in project management, an early process that makes your project come to life. It's known as a starting phase for the project, because without it, you really don't have the tools or the blueprint to get the project going.

In this phase, you plan all of a project's key features, its elements for success, major deliverables, structure, and more. Usually in this phase, you'll come up with one or more designs that will showcase and later be used to achieve your projected goals. Thanks to project design, stakeholders are then able to choose the best design for the fulfillment of the project, making the whole process smoother and easier for the entire team.

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	Vijay S Dhinakaran V
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	2	High	Pugalarasan G Naveen K M
Sprint-2		USN-3	As a user, I can register for the application through Social media accounts	2	Low	Vijay S Dhinakaran V
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	2	Medium	Vijay S Dhinakaran V
Sprint-2	Dashboard	USN-5	Once logged in , based on user's expenses and data records , graphical representation is achieved	4	High	Vijay S Dhinakaran V Pugalarasan G Naveen KM

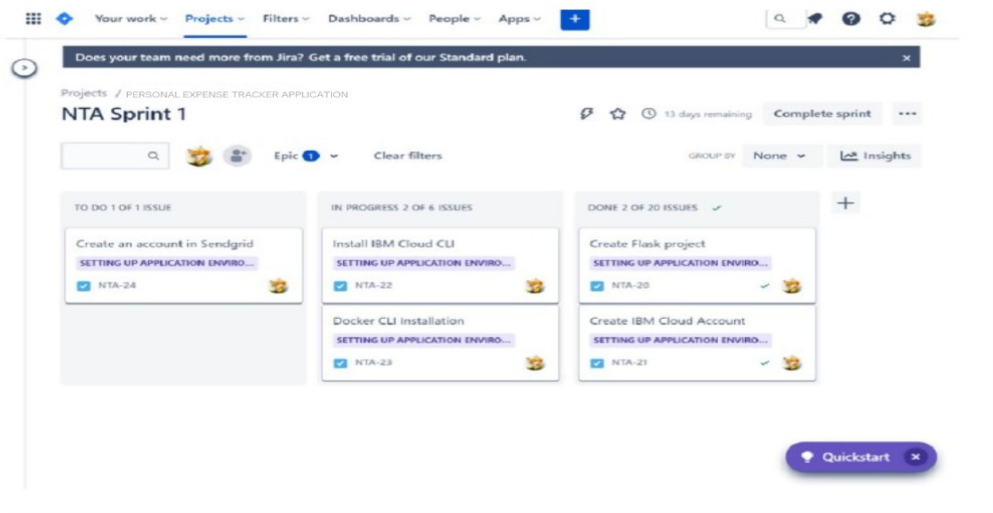
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Financial account	USN-6	As a user, I can add and remove any financial accounts	2	High	Vijay S Dhinakaran V
Sprint-3	Notifications	USN-7	As a user, I can receive alerting notifications on untracked expenses	2	High	Vijay S Pugalarasan G Naveen K M
Sprint-3		USN-8	As a user, I can receive suggesting notifications for saving and earning money	2	Medium	Pugalarasan G Naveen KM
Sprint-4	Security	USN-9	As a user, I am assured for linking my financial accounts securely	4	High	Vijay S Dhinakaran V
Sprint-4	Customer care	USN-10	As a user, I can access the customer care for any queries and issues regarding the applications	2	Medium	Pugalarasan G Naveen KM

6.2 Sprint Delivery Schedule

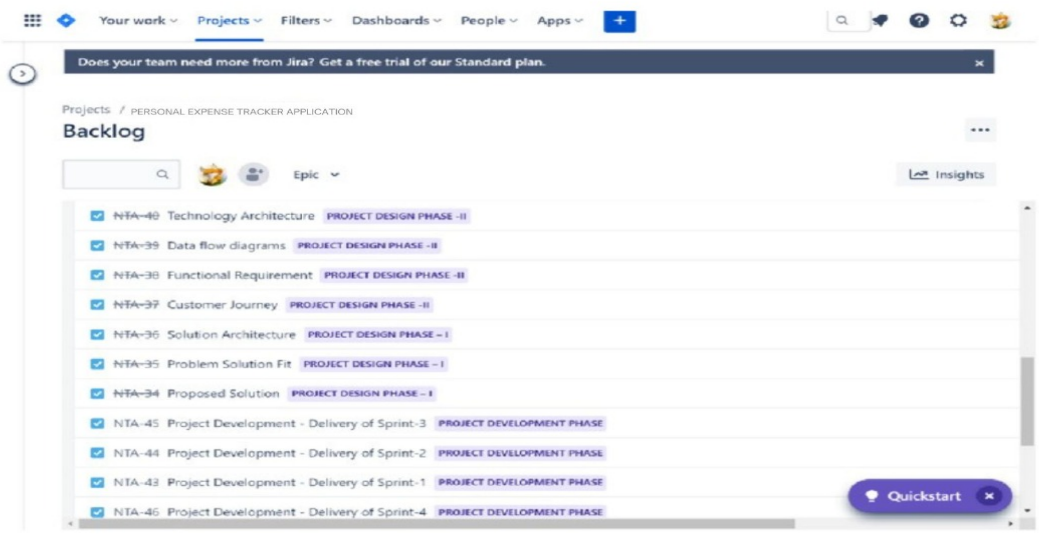
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	3	6 Days	24 Oct 2022	29 Oct 2022	3	29 Oct 2022
Sprint-2	2	6 Days	31 Oct 2022	05 Nov 2022	2	05 Nov 2022
Sprint-3	3	6 Days	07 Nov 2022	12 Nov 2022	3	12 Nov 2022
Sprint-4	2	6 Days	14 Nov 2022	19 Nov 2022	2	19 Nov 2022

6.3 Reports from JIRA

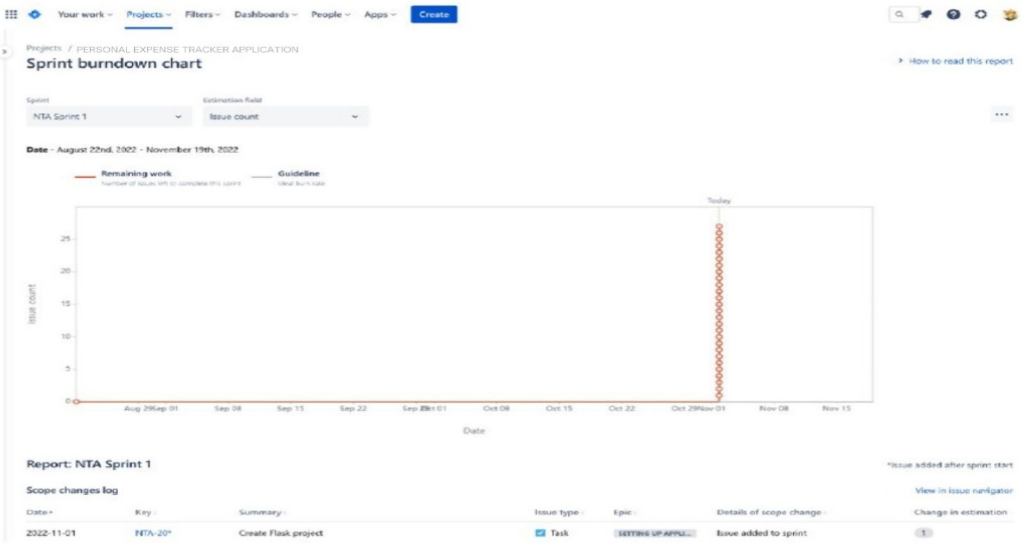
6.3.1 JIRA Board



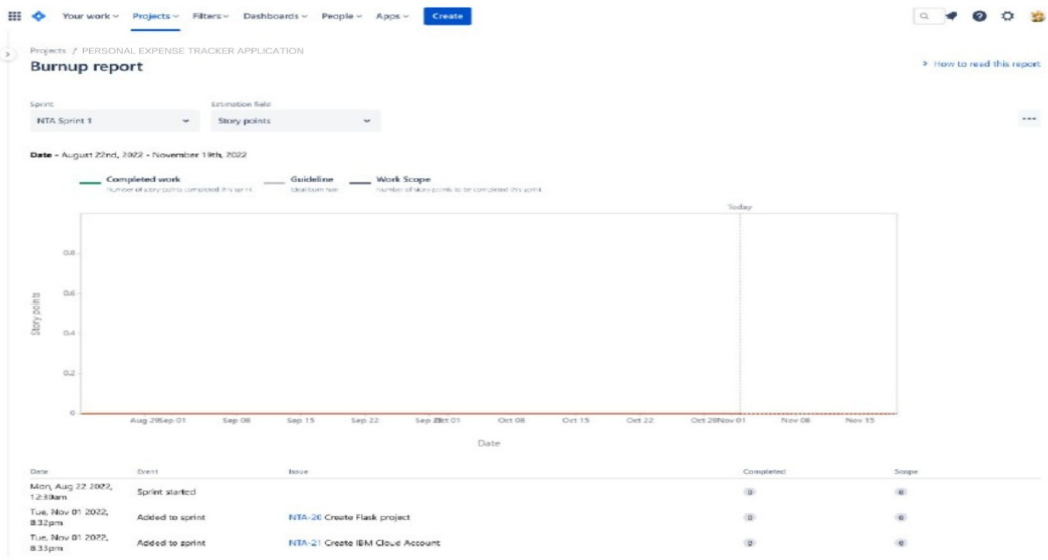
6.3.2 JIRA Backlog Report



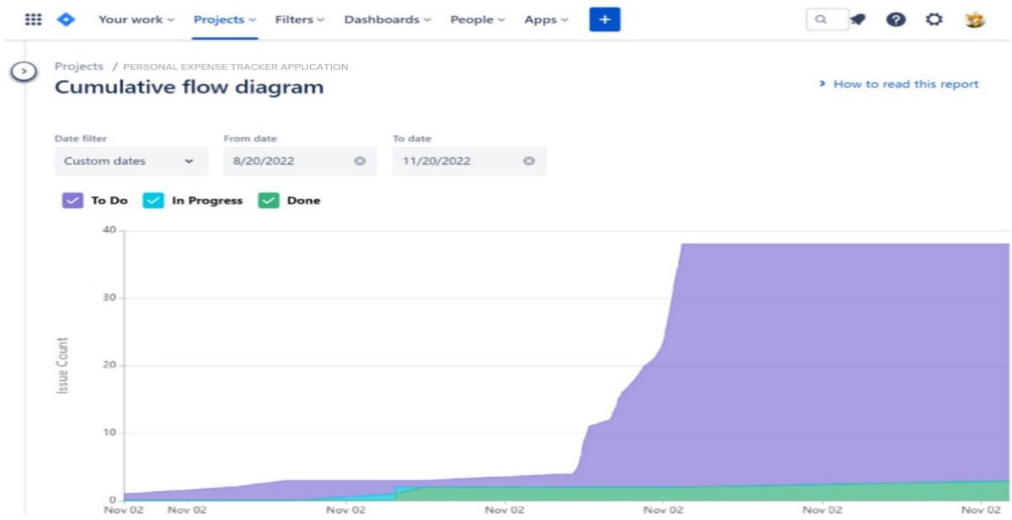
6.3.3 Burndown Chart



6.3.4 Burnup Chart



6.3.5 Cumulative Flow Diagram



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

7.1 Feature 1

Code Listing 1: Balance.html

```
<!doctype html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.
min.css" rel="stylesheet" integrity="sha384
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
```

```
<title>Update Balance</title>
</head>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.
bundle.min.js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

<div class="container-fluid" >
<div class="row flex-nowrap">
<div class="col-auto col-md-3 col-xl-2 px-sm-2 px-0" style="background-
color: #B2D3C2">
<div class="d-flex flex-column align-items-center align-items-sm-start
px-3 pt-2 min-vh-100" style="color: black">
<p class="d-flex align-items-center pb-3 mb-md-0 me-md-auto text-white
text-decoration-none">
<span class="fs-5 d-none d-sm-inline" style="color: black; font-weight:
bold;">Personal Expense Tracker</span>
<img src="bank.png" style="width:50px; height: 50px;">
</p>
<ul class="nav nav-pills flex-column mb-sm-auto mb-0 align-items-center
align-items-sm-start" id="menu">
<li class="nav-item mt-2">
<a href="dashboard" class="nav-link align-middle px-0"
style="color: black;">
<span class="ms-1 d-none d-sm-inline">Home</span><img src="house-
outline.svg" style="width:20px; height:20px; margin-left: 5px;">
</a>
</li>
<li class="nav-item mt-2" >
<a href="addexpense" class="nav-link px-0 align-middle"
style="color: black;">
<span class="ms-1 d-none d-sm-inline">Add Expense</span><img
src="pay.png" style="width:20px; height:20px; margin-left: 5px;">
</a>
```

```
</li>
<li class="nav-item mt-2">
<a href="modifyexpense" class="nav-link px-0 align-middle"
style="color:black;">
<span class="ms-1 d-none d-sm-inline">Modify Expense</span><img
src="edit_icon.svg" style="width:20px;height:20px;margin-left: 5px;">
</a>
</li>
<li class="nav-item mt-2">
<a href="analysis" class="nav-link px-0 align-middle"
style="color:black;">
<span class="ms-1 d-none d-sm-inline">View Analysis</span><img
src="graph.png" style="width:20px;height:20px;margin-left: 5px;">
</a>
</li>
<li class="nav-item mt-2">
<a href="rewards" class="nav-link px-0 align-middle" style="
color:black;">
<span class="ms-1 d-none d-sm-inline">Rewards & Goals</span><img
src="reward.png" style="width:20px;height:20px;margin-left: 5px;">
</a>
</li>
<li class="nav-item mt-2">
<a href="addcategory" class="nav-link px-0 align-middle"
style="color:black;">
<span class="ms-1 d-none d-sm-inline">Create Category
</span><img src="category_add.webp" style="width:20px;height:
20px;margin-left: 5px;">
</a>
</li>
</ul>
</div>
</div>
<div class="col py-3" style="background-color:#00AD83">
<h3 style="color:white; text-align: center;">Update Balance</h3>
```

```
<div class="container mt-3" style="width: 600px;">
<div class="card shadow-lg bg-white rounded">
<form action="/updatebalance" method="POST">
<div class="card-header" style="text-align: center;">
<span style="display: inline-flex"><h4>Wallet Balance</h4><img
src="wallet_money.webp" style=" margin-left: 10px; width: 30px;
height: 30px"></span>
</div>
<div class="card-body">
<div class="mb-3">
<label for="category" class="form-label">Current Balance: </label>
<input type="text" value={{wallet}} readonly>
</div>
<div class="mb-3">
<label for="description" class="form-label">New Balance: </label>
<input type="text" class="form-control" name="balanceupdated"
id="balanceupdated"></input>
</div>
</div>
<div class="card-footer text-muted" style="text-align: center">
<button type="submit" style="background-color: #00AD83; border-
color: #00AD83; border-radius: 5px;">Update Balance</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>

</html>
```

7.2 Feature 2

Code Listing 2: HTML exampl

```
<!doctype html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap
.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

<title>AddExpense</title>
</head>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.
bundle.min.js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

<div class="container-fluid" >
<div class="row flex-nowrap">
<div class="col-auto col-md-3 col-xl-2 px-sm-2 px-0" style="background-
color: #B2D3C2">
<div class="d-flex flex-column align-items-center align-items-sm-start
px-3 pt-2 min-vh-100" style="color: black">
<p class="d-flex align-items-center pb-3 mb-md-0 me-md-auto text-white
text-decoration-none">
<span class="fs-5 d-none d-sm-inline" style="color: black; font-weight:
bold;">Personal Expense Tracker</span>
<img src="bank.png" style="width:50px; height: 50px;">
</p>
```

```
<ul class="nav nav-pills flex-column mb-sm-auto mb-0 align-items-center
align-items-sm-start" id="menu">
<li class="nav-item mt-2">
<a href="Dashboard.html" class="nav-link align-middle px-0"
style="color:black;">
<span class="ms-1 d-none d-sm-inline">Home</span><img src="house-
outline.svg" style="width:20px;height:20px;margin-left: 5px;">
</a>
</li>
<li class="nav-item mt-2" style="background-color:#00AD83; height: 50px;
width: 150px; border-radius: 5px;" >
<a href="Expense.html" class="nav-link px-0 align-middle"
style="color:black;">
<span class="ms-1 d-none d-sm-inline">Add Expense</span><img
src="pay.png" style="width:20px;height:20px;margin-left: 5px;">
</a>
</li>
<li class="nav-item mt-2">
<a href="modifyexpense.html" class="nav-link px-0 align-middle"
style="color:black;">
<span class="ms-1 d-none d-sm-inline">Modify Expense</span><img
src="edit_icon.svg" style="width:20px;height:20px;margin-left: 5px;">
</a>
</li>
<li class="nav-item mt-2">
<a href="analysis.html" class="nav-link px-0 align-middle"
style="color:black;">
<span class="ms-1 d-none d-sm-inline">View Analysis</span><img
src="graph.png" style="width:20px;height:20px;margin-left: 5px;">
</a>
</li>
<li class="nav-item mt-2">
<a href="rewards.html" class="nav-link px-0 align-middle"
style="color:black;">
<span class="ms-1 d-none d-sm-inline">Rewards & Goals</span><img
```

```
src="reward.png" style="width:20px;height:20px;margin-left:5px;">
</a>
</li>
</ul>
</div>
</div>
<div class="col py-3" style="background-color:black">
<h3 style="color:white; text-align: center;">Add expense</h3>
<div class="container mt-3" style="width: 600px;">
<div class="card shadow-lg bg-white rounded">
<div class="card-header" style="text-align: center;">
<span style="display:inline-flex"><h4>Expense Made</h4><img src=
"pay.png" style="margin-left:10px; width:30px; height:30px"></span>
</div>
<div class="card-body">
<form>
<div class="mb-3">
<label for="amountspent" class="form-label">Amount Spent: (Rs) </label>
<input type="number" class="form-control" name="amountspent"
id="amountspent" placeholder="100.00">
</div>
<div class="mb-3">
<label for="expensecategory" class="form-label">Expense Category:
</label><input type="text" class="form-control" name="expensecategory"
id="expensecategory"></input>
</div>
<div class="mb-3">
<label for="date" class="form-label">Date of Expense: </label>
<input type="date" class="form-control" name="date" id="date"></input>
</div>
<div class="mb-3">
<label for="description" class="form-label">Description of Expense:
</label>
<input type="text" class="form-control" name="description"
id="description"></input>
```

```
</div>
</form>
</div>
<div class="card-footer text-muted" style="text-align:center">
<button type="submit" style="background-color:#00AD83; border-
color:#00AD83; border-radius:5px;">Submit Expense</button>
</div>
</div>
</div>
</div>
</div>

</html>
```

7.3 Database Schema (if Applicable)

A database schema defines how data is organized within a relational database; this is inclusive of logical constraints such as, table names, fields, data types, and the relationships between these entities. Schemas commonly use visual representations to communicate the architecture of the database, becoming the foundation for an organization's data management discipline. This process of database schema design is also known as data modeling.

These data models serve a variety of roles, such as database users, database administrators, and programmers. For example, it can help database administrators manage normalization processes to avoid data duplication. Alternatively, it can enable analysts to navigate these data structures to conduct reporting or other valuable business analyses. These diagrams act as valuable documentation within the database management system (DBMS), ensuring alignment across various stakeholders.

8 TESTING

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

8.1 Test Cases

A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly. The purpose of a test case is to determine if different features within a system are performing as expected and to confirm that the system satisfies all related standards, guidelines and customer requirements. The process of writing a test case can also help reveal errors or defects within the system.

- Verify if user can login.
- Verify if the UI elements are getting displayed properly.
- Verify if the user can select the Expenses displayed.
- Verify if the user can register.
- Verify if the application displays Expenditures.
- Verify if the user can logout.
- Verify if the user is getting redirected to the destination
- Verify if the user can update the Expenses
- • Verify if the user login details are stored in the database

8.2 User Acceptance Testing

8.2.1 Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Personal Expense Tracker Application project at the time of the release to User Acceptance Testing (UAT).

8.2.2 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	3	2	2	1	8
Duplicate	1	0	2	0	3
External	2	0	0	1	3
Fixed	6	3	4	6	19
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	1	1
Won't Fix	0	3	1	1	5
Totals	12	8	10	10	40

8.2.3 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	6	0	0	6
Client Application	5	0	0	5
Security	2	0	0	2
Outsource Shipping	7	0	0	7
Exception Reporting	5	0	0	5
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9 RESULTS

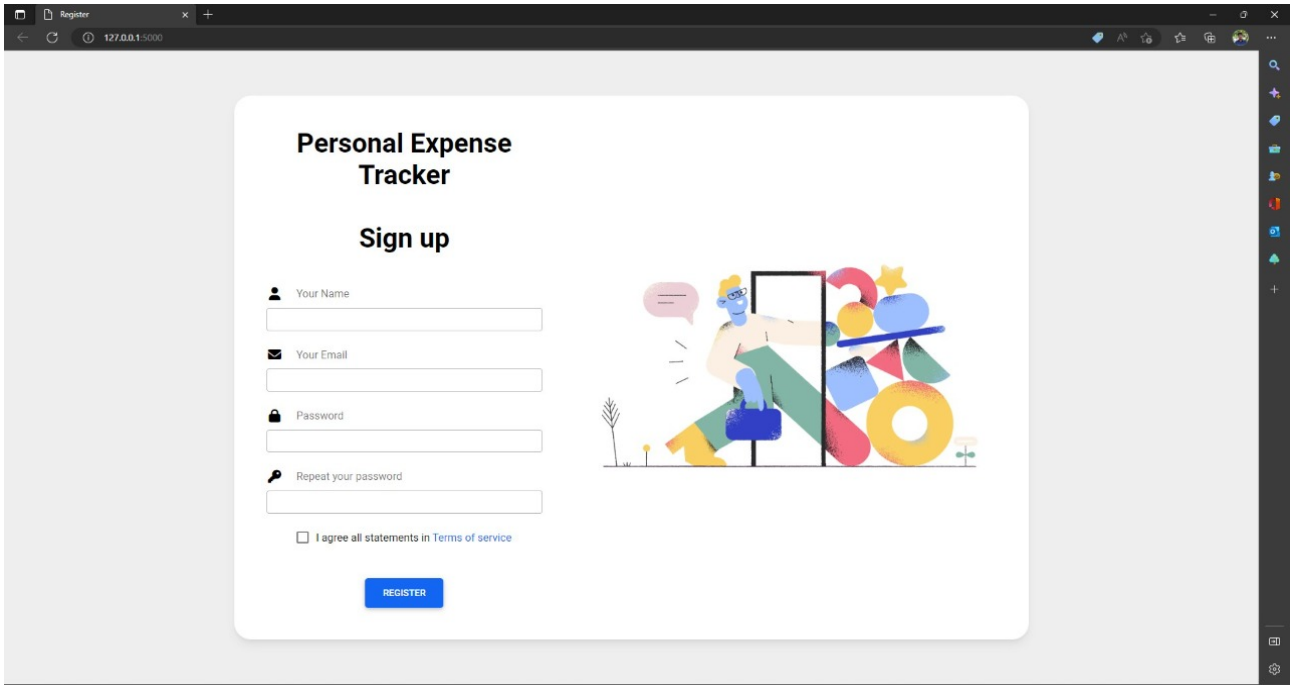


Figure 8: Sign Up:

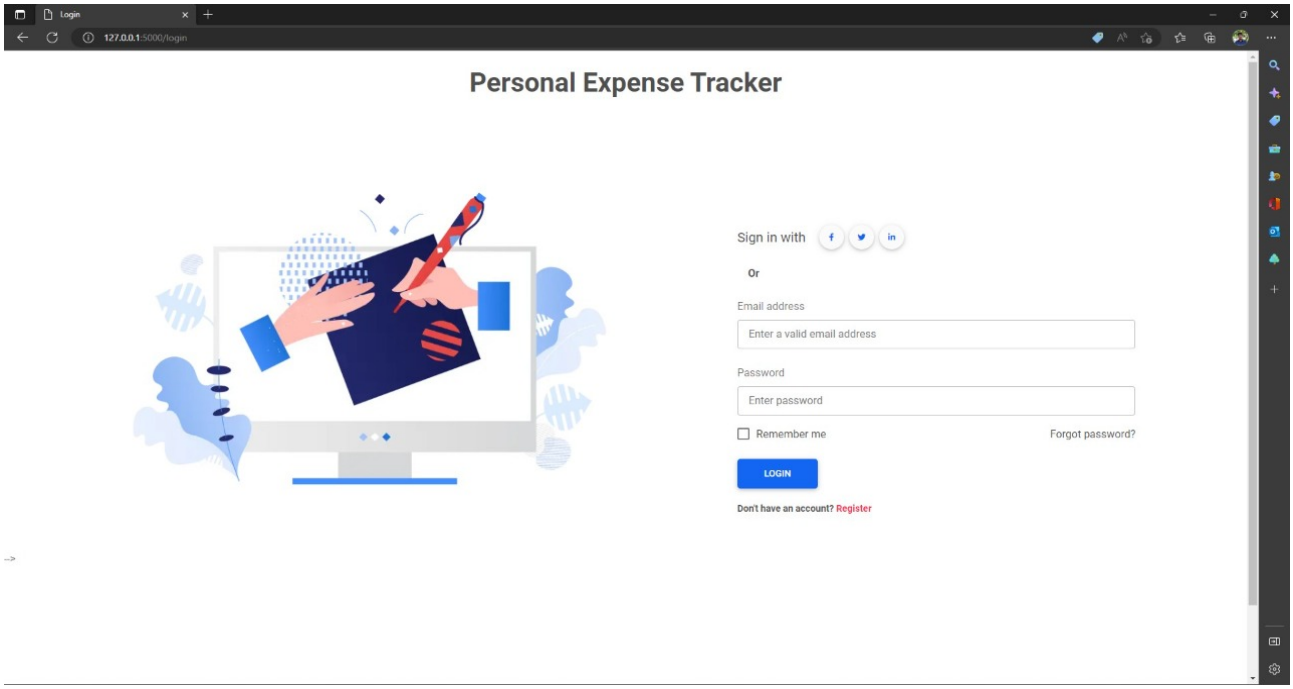


Figure 9: Login:

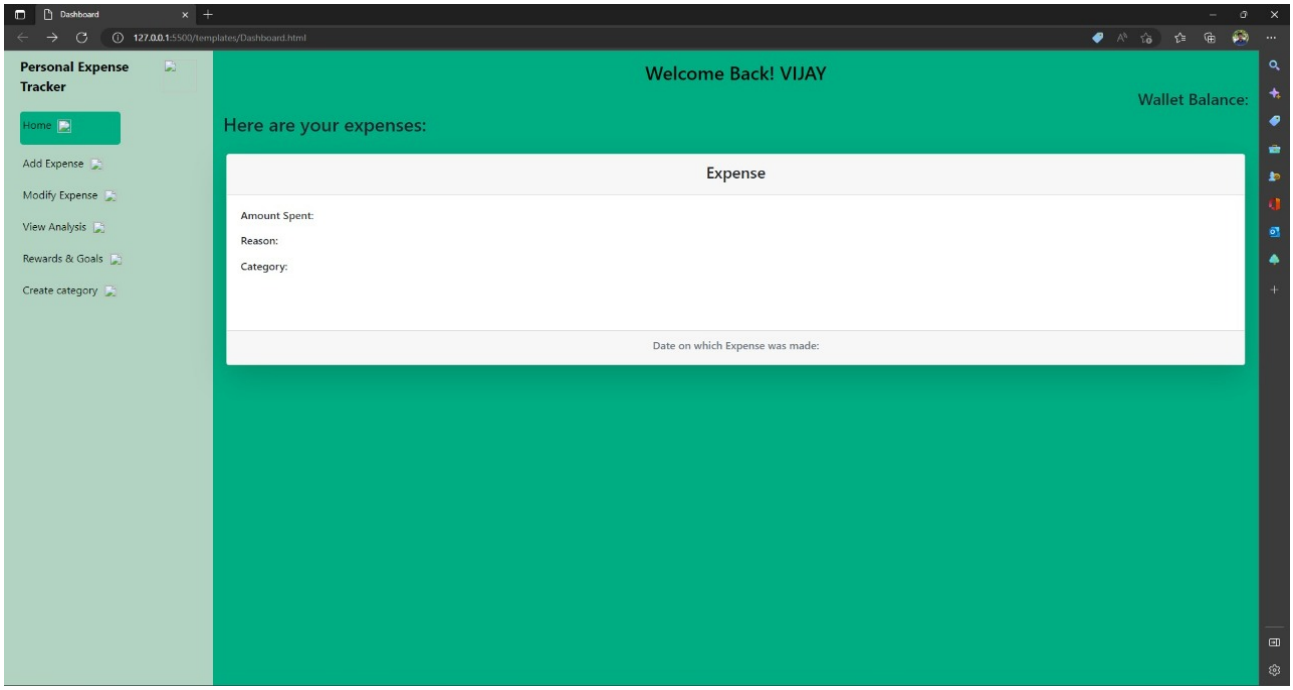


Figure 10: Dashboard:

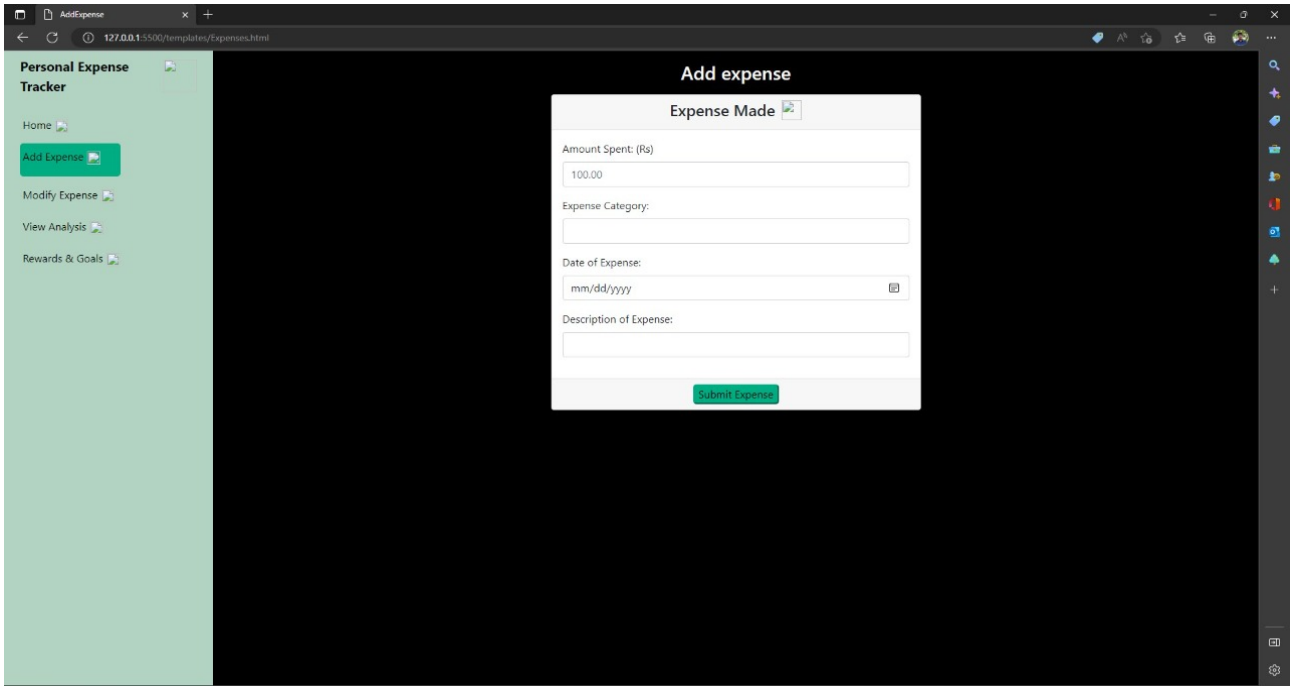


Figure 11: Add Expenses:

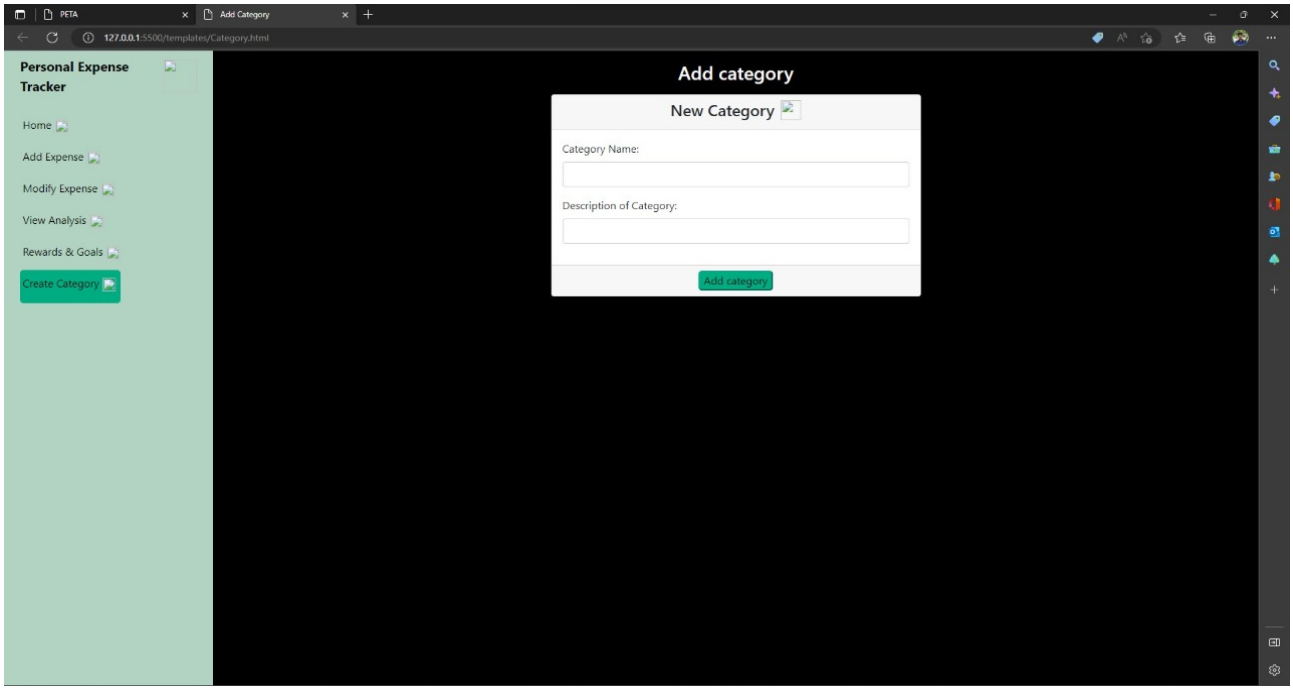


Figure 12: Add Category:

10 ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Easy user interaction.
- Faster loading of Expenses
- Easy login process.
- Multiple users can access at same time.
- Easy to calculate and maintain the expenses.
- Easy to retrieve the Information.
- Helps to control unnecessary spending and make a budget better.

DISADVANTAGES

- No advance security layer for data protection.
- Loading time maybe increase If the Server is low.
- Application doesn’t work in older browser.

11 CONCLUSION

Tracking the daily expenses can not only help in saving money but also help in setting financial goals for the future. If we know where our money is being spent every day, it is easy to set some cutbacks and such to help reduce expenditure. This project is developed to work more efficiently in comparison to other trackers and avoid manual calculation. It is developed to be efficient and look attractive at the same time.

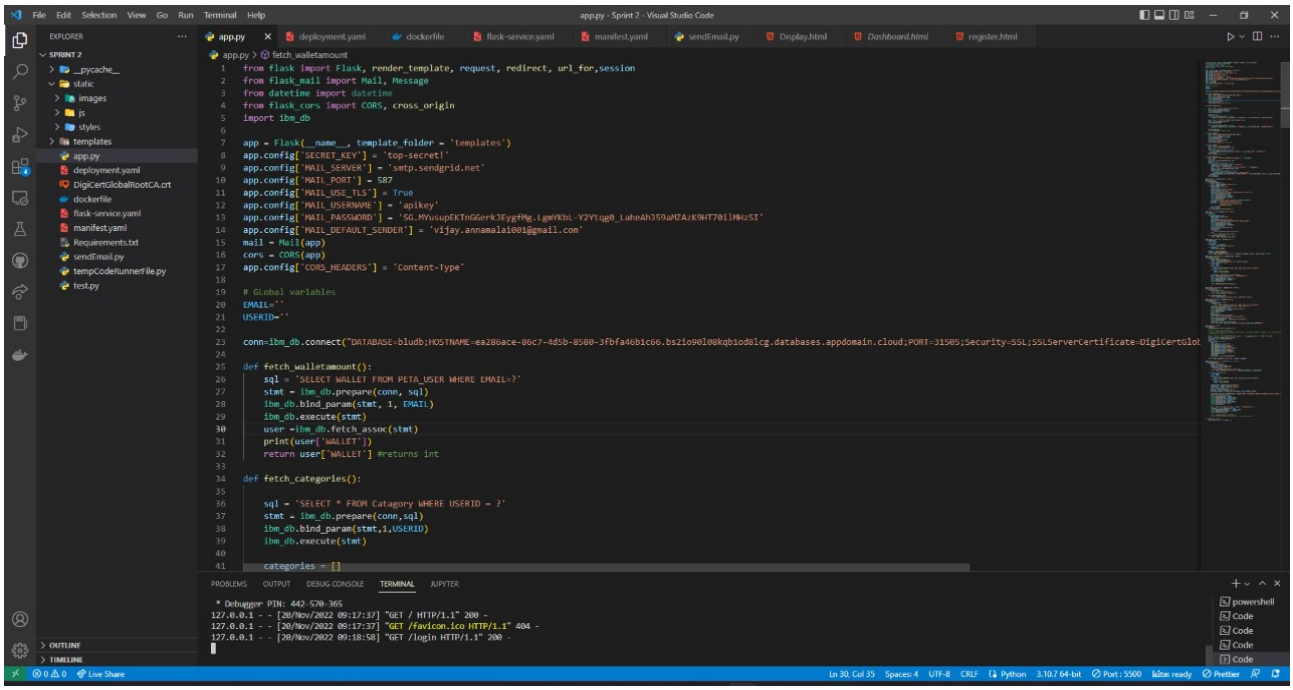
Monitoring your everyday expenses can set aside you cash, yet it can likewise help you set your monetary objectives for what's to come. On the off chance that you know precisely where your sum is going much of a stretch see where a few reductions and bargains can be made. Expense Tracker project is for keeping our day-to-day expenditures will helps us to keep record of our money daily. The project what we have created is work more proficient than the other income and expense tracker. The project effectively keeps away from the manual figuring for trying not to ascertain the pay and cost each month. It's a user-friendly application.

12 FUTURE SCOPE

- It will have various options to keep record (forexample Food, Travelling Fuel, Salary etc.).
- Automatically it will keep on sending notifications for our daily expenditure.
- 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.
- 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.

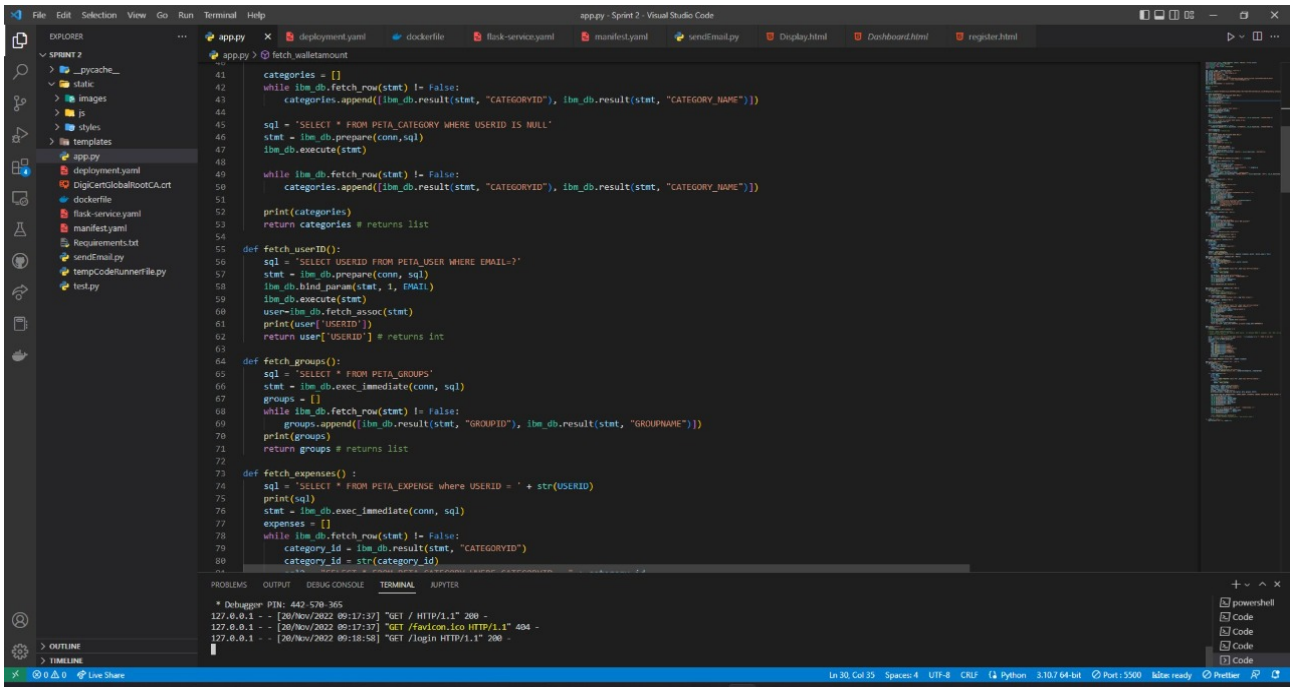
13 APPENDIX

13.1 Source Code



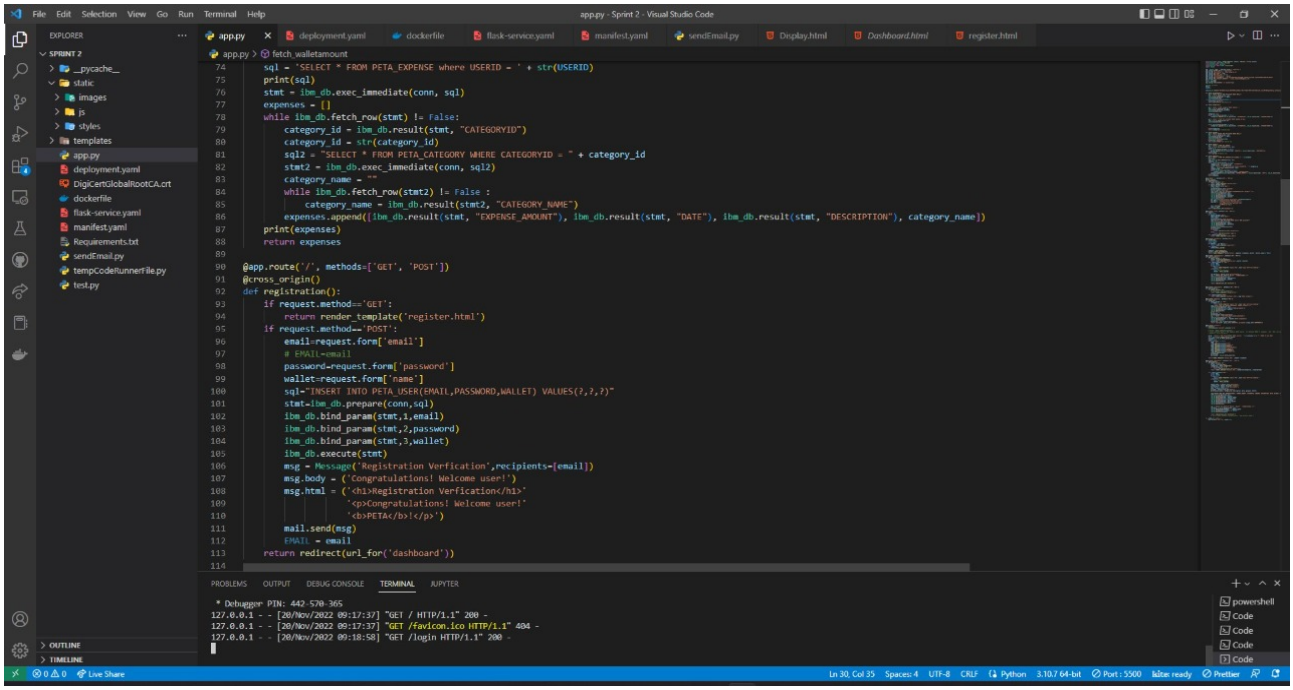
```
from flask import Flask, render_template, request, redirect, url_for,session
from flask_mail import Mail, Message
from datetime import datetime
from flask_cors import CORS, cross_origin
import ibm_db

app = Flask(__name__, template_folder = 'templates')
app.config['SECRET_KEY'] = 'top-secret!'
app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USERNAME'] = 'apikey'
app.config['MAIL_PASSWORD'] = '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'
app.config['MAIL_DEFAULT_SENDER'] = 'vijay.annamalai001@gmail.com'
mail = Mail(app)
cors = CORS(app)
app.config['CORS_HEADERS'] = 'Content-Type'

# Global variables
EMAIL=''
USERID=''

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;Security=SSL;SSLServerCertificate=DigiCertGlot

def fetch_walletamount():
    sql = 'SELECT WALLET FROM PETA_USER WHERE EMAIL=?'
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, EMAIL)
    ibm_db.execute(stmt)
    user =ibm_db.fetch_assoc(stmt)
    print(user['WALLET'])
    return user['WALLET'] #returns int

def fetch_categories():

    sql = 'SELECT * FROM Catagory WHERE USERID = ?'
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,USERID)
    ibm_db.execute(stmt)

    categories = []
```

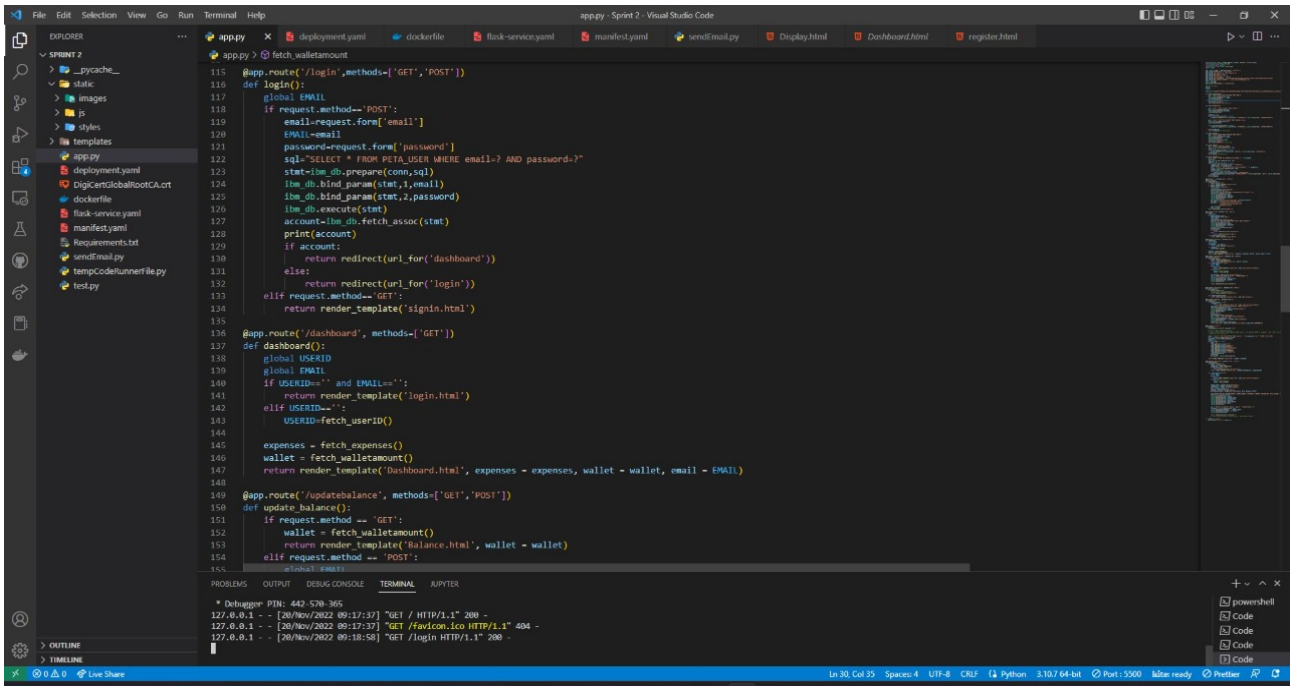
```
* Debugger PIN: 442-570-365
127.0.0.1 - - [20/Nov/2022 09:17:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [20/Nov/2022 09:17:37] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [20/Nov/2022 09:18:58] "GET /login HTTP/1.1" 200 -
```

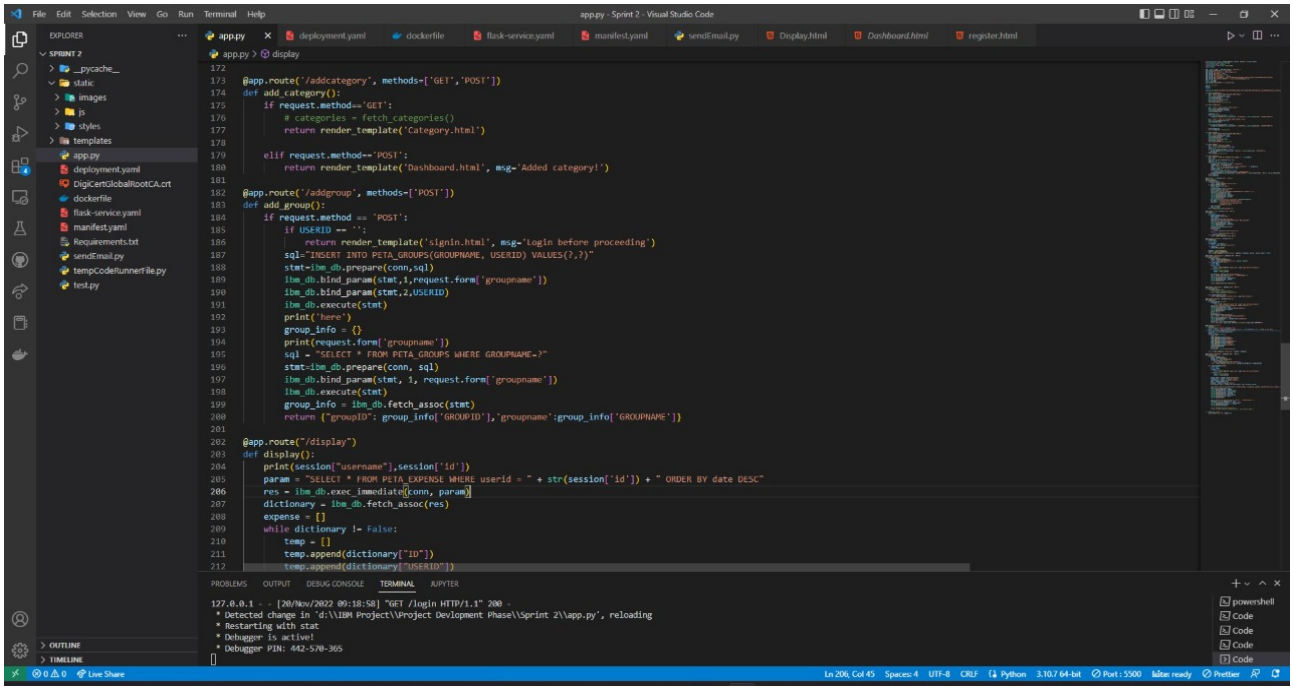


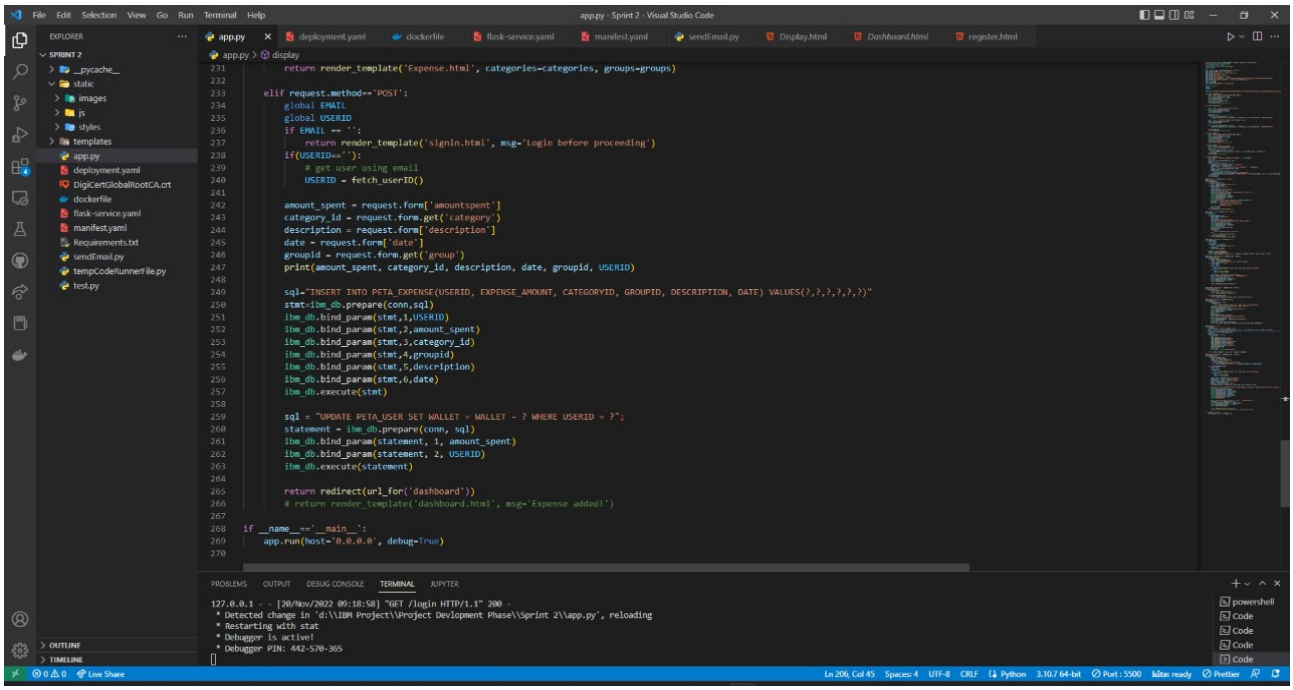
```
    categories = []
    while ibm_db.fetch_row(stmt) != False:
        categories.append([ibm_db.result(stmt, "CATEGORYID"), ibm_db.result(stmt, "CATEGORY_NAME")])

    sql = 'SELECT * FROM PETA_CATEGORY WHERE USERID IS NULL'
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.execute(stmt)

    while ibm_db.fetch_row(stmt) != False:
        categories.append([ibm_db.result(stmt, "CATEGORYID"), ibm_db.result(stmt, "CATEGORY_NAME")])

    print(categories)
    return categories # returns list

def fetch_userID():
    sql = 'SELECT USERID FROM PETA_USER WHERE EMAIL=?'
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, EMAIL)
    ibm_db.execute(stmt)
    user=ibm_db.fetch_assoc(stmt)
    print(user['USERID'])
    return user['USERID'] # returns int

def fetch_groups():
    sql = 'SELECT * FROM PETA_GROUPS'
    stmt = ibm_db.exec_immediate(conn, sql)
    groups = []
    while ibm_db.fetch_row(stmt) != False:
        groups.append([ibm_db.result(stmt, "GROUPID"), ibm_db.result(stmt, "GROUPNAME")])
    print(groups)
    return groups # returns list

def fetch_expenses() :
    sql = 'SELECT * FROM PETA_EXPENSE where USERID = ' + str(USERID)
    print(sql)
    stmt = ibm_db.exec_immediate(conn, sql)
    expenses = []
    while ibm_db.fetch_row(stmt) != False:
        category_id = ibm_db.result(stmt, "CATEGORYID")
        category_id = str(category_id)
```

```
* Debugger PIN: 442-570-365
127.0.0.1 - - [20/Nov/2022 09:17:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [20/Nov/2022 09:17:37] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [20/Nov/2022 09:18:58] "GET /login HTTP/1.1" 200 -
```


app.py - Sprint 2 - Visual Studio Code
app.py
deployment.yaml
dockerfile
flask-service.yaml
manifest.yaml
sendEmail.py
Display.html
Dashboard.html
register.html
app.py > fetch_walletamount
sql = 'SELECT * FROM PETA_EXPENSE where USERID = ' + str(USERID)
print(sql)
stmt = ibm_db.exec_immediate(conn, sql)
expenses = []
while ibm_db.fetch_row(stmt) != False:
category_id = ibm_db.result(stmt, "CATEGORYID")
category_id = str(category_id)
sql2 = "SELECT * FROM PETA_CATEGORY WHERE CATEGORYID = " + category_id
stmt2 = ibm_db.exec_immediate(conn, sql2)
category_name = ""
while ibm_db.fetch_row(stmt2) != False :
category_name = ibm_db.result(stmt2, "CATEGORY_NAME")
expenses.append([ibm_db.result(stmt, "EXPENSE_AMOUNT"), ibm_db.result(stmt, "DATE"), ibm_db.result(stmt, "DESCRIPTION"), category_name])
print(expenses)
return expenses
@app.route('/', methods=['GET', 'POST'])
@cross_origin()
def registration():
if request.method=='GET':
return render_template('register.html')
if request.method=='POST':
email=request.form['email']
# EMAIL=email
password=request.form['password']
wallet=request.form['name']
sql="INSERT INTO PETA_USER(EMAIL,PASSWORD,WALLET) VALUES(?,?,?)"
stmt=ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,email)
ibm_db.bind_param(stmt,2,password)
ibm_db.bind_param(stmt,3,wallet)
ibm_db.execute(stmt)
msg = Message('Registration Verfication',recipients=[email])
msg.body = ('Congratulations! Welcome user!')
msg.html = ('<h1>Registration Verfication</h1>'
' Congratulations! Welcome user!'
'PETA! ')
mail.send(msg)
EMAIL = email
return redirect(url_for('dashboard'))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
* Debugger PIN: 442-570-365
127.0.0.1 - - [20/Nov/2022 09:17:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [20/Nov/2022 09:17:37] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [20/Nov/2022 09:18:58] "GET /login HTTP/1.1" 200 -
powershell
Code
Code
Code
Code
Ln 30, Col 35 Spaces: 4 UTF-8 CRLF Python 3.10.7 64-bit Port : 5500 Ikiter ready Prettier


app.py - Sprint 2 - Visual Studio Code
app.py
deployment.yaml
dockerfile
flask-service.yaml
manifest.yaml
sendEmail.py
Display.html
Dashboard.html
register.html
app.py > fetch_walletamount
@app.route('/login',methods=['GET','POST'])
def login():
global EMAIL
if request.method=='POST':
email=request.form['email']
EMAIL=email
password=request.form['password']
sql="SELECT * FROM PETA_USER WHERE email=? AND password=?"
stmt=ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,email)
ibm_db.bind_param(stmt,2,password)
ibm_db.execute(stmt)
account=ibm_db.fetch_assoc(stmt)
print(account)
if account:
return redirect(url_for('dashboard'))
else:
return redirect(url_for('login'))
elif request.method=='GET':
return render_template('signin.html')
@app.route("/dashboard", methods=['GET'])
def dashboard():
global USERID
global EMAIL
if USERID=='' and EMAIL=='':
return render_template('login.html")
elif USERID=='':
USERID=fetch_userID()
expenses = fetch_expenses()
wallet = fetch_walletamount()
return render_template('Dashboard.html', expenses = expenses, wallet = wallet, email = EMAIL)
@app.route('/updatebalance', methods=['GET','POST'])
def update_balance():
if request.method == 'GET':
wallet = fetch_walletamount()
return render_template('Balance.html', wallet = wallet)
elif request.method == 'POST':
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
* Debugger PIN: 442-570-365
127.0.0.1 - - [20/Nov/2022 09:17:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [20/Nov/2022 09:17:37] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [20/Nov/2022 09:18:58] "GET /login HTTP/1.1" 200 -
powershell
Code
Code
Code
Code
Ln 30, Col 35 Spaces: 4 UTF-8 CRLF Python 3.10.7 64-bit Port : 5500 Ikiter ready Prettier





13.2 Github:

https://github.com/IBM-EPBL/IBM-Project-50496-1660912705