



NALAIYATHIRAN



PROJECT BASED EXPERIENTIAL LEARNING

PERSONAL EXPENSE TRACKER APPLICATION

BATCH ID

B11 – 5A1E



PERSONAL EXPENSE TRACKER APPLICATION

A PROJECT REPORT

Submitted by

TEAM ID - PNT2022TMID17124

NAME OF THE CANDIDATE(S)

NANDHA KUMAR M	6113191041065
NAVEEN R	6113191041067
SABARI K	6113191041087
THAMODHARAN S	6113191041109

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

MAHENDRA ENGINEERING COLLEGE

NOVEMBER 2022

MAHENDRA ENGINEERING COLLEGE

(Autonomous)

Mahendhirapuri, Mallasamudram, Namakkal Dt. - 637 503

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

Certified that this Project Based Experiential Learning Report titled **“PERSONAL EXPENSE TRACKER APPLICATION”** is the bonafide work of **NANDHA KUMAR M (6113191041065), NAVEEN R (6113191041067), SABARI K(6113191041087) and THAMODHARAN S (6113191041109)** who carried out the project work under my supervision.

Mrs.Dr.S.PONLATHA

Assistant Professor,

Department of ECE,

Mahendra Engineering College,

Mahendhirapuri,Mallasamudram,

Namakkal Dt. - 637 503.

Mrs.P.BALASHANMUGA VADIVU

Assistant Professor,

Department of ECE,

Mahendra Engineering College,

Mahendhirapuri,Mallasamudram,

Namakkal Dt. - 637 503.

TABLE OF CONTENTS

1.	INTRODUCTION	6
1.1	Project Overview	6
1.2	Purpose	6
2.	LITERATURE SURVEY	7
2.1	Existing problem	8
2.2	References	9
2.3	Problem Statement Definition	9
3.	IDEATION & PROPOSED SOLUTION	10
3.1	Empathy Map Canvas	10
3.2	Ideation & Brainstorming	11
3.3	Proposed Solution	12
3.4	Problem Solution fit	14
4.	REQUIREMENT ANALYSIS	15
4.1	Functional requirement	15
4.2	Non-Functional requirements	16
5.	PROJECT DESIGN	17
5.1	Data Flow Diagrams	17
5.2	Solution & Technical Architecture	18

5.3	User Stories	19
6.	PROJECT PLANNING & SCHEDULING	20
6.1	Sprint Planning & Estimation	20
6.2	Sprint Delivery Schedule	21
6.3	Reports from JIRA	23
7.	CODING & SOLUTIONING	24
	(Explain the features added in the project along with code)	
7.1	Feature 1	24
7.2	Feature 2	26
7.3	Database Schema (if Applicable)	28
8.	TESTING	30
8.1	Test Cases	30
8.2	User Acceptance Testing	30
9.	RESULTS	31
10.	ADVANTAGES	31
11.	CONCLUSION	32
12.	FUTURE SCOPE	32
13.	APPENDIX	33

Source Code & GitHub & Project Demo Link

PROJECT REPORT

1. INTRODUCTION

With the launch and increase in sales of smartphones over the last few years, people are using mobile applications to get their work done, which makes their lives easier. Mobile applications comprise various different categories such as Entertainment, Sports, Lifestyle, Education, Games, Food and Drink, Health and Fitness, Finance, etc. This Expense Tracker application falls in the Finance Category and serves the important purpose of managing finances which is a very important part of one's life. The software product went through the design, development, and the testing phase as a part of the Software Development Lifecycle.

1.1 PROJECT OVERVIEW

In today's busy and expensive life we are in a great rush to make money. But at the end of the month, we broke off. As we are unknowingly spending money on little and unwanted things. So, we have come over with the idea to track our earnings. The daily expense tracker aims to help everyone who is planning to know their expenses and save from it. An Personal expense tracker is a app that the user's can update their daily expenses so that they are well known for their expenses. Here users can choose categories for expense types like food, clothing, rent, and bills where they have to enter the money that has been spent and also can add some information in additional information to specify the expense. Users will be able to see graphs of expenses. Users can save those graphs for their own reference.

1.2 PURPOSE

In proposed system user has more number of added features to the existing features like

- Weekly Budget Planner, Automated message Alert, UPI linkup, Weekly and Monthly Analysis, Weekly Budget Planner to track their expenses. Automated message Alert is generated when they cross their budget. UPI linkup to track their online transactions. Weekly and Monthly Analysis are generated in the form of pie chart. App Authentication for security of the user. Income, Expenses, and Wish List are the three data entry choices available to the use.

2. LITERATURE SURVEY

S.no	Title	Author	Journal	Year	Methodology
1.	Security and privacy challenges in mobile cloud computing : survey and way ahead	Muhammad Baqer Mollah et al	Journal of Network and Computer Applications	2017	Computational offloading, Virtualization
2.	Exploring infrastructure support for app based services on cloud platforms	Hai Nguyen et al	Journal of Cloud Computing Advances Systems and Applications	2017	Virtualization, Introspection and Security
3.	Mobile Financial Management Application using Google Cloud Vision AP	Kurniawan Dwi Saputra et al	International Conference on Computer Science and Computaional Intelligence	2019	Google Vision Cloud API, Optical Character Recognition
4.	Cloud Based Development Issues	Sukhpal Singh, Inderveer Chana	International Journal of Cloud Computing and Services Science	2020	A Methodical Analysis
5.	Expense Tracker	Prof Miriam Thomas, Lekshmi P, and	International Journal of Advanced Research in	2020	Least Square Algorithm

		Dr.Mahalekshmi T	Science, Communication and Technology		
6.	Cloud based Expense Tracker	Asthha Wahal and Muskan Aggarwal	International Journal of Innovative Research in Technology	2021	Clustering, Apriori algorithm, Virtualization
7.	A Novel Expense Tracker	Muskaan Sharma, Ayush Bansal, Dr. Raju Ranjan, Shivam Sethi	International Journal of Innovative Research in Technology	2021	Statistical Analysis, Data Mining
8.	Student Expense Tracking Application	Saumya Dubey, Pragya Dubey, Rigved Rishabh Kumar, Aaisha Khatoon	International Journal of Advance Research and Innovative Ideas in Education	2022	Clustering, Classification and Association

2.1 EXISTING PROBLEM

An expense tracker to prevent having to calculate income and expenses, as well as to remind someone to keep their expenses in track and also to add some details on how much money comes from other people and what expenses or payments the user have to make on a given date or month, User have categories in the expenditure tracker such as add expense, monthly expenses, add new expense, see categories of spending, export expenses in a date range, remove export files, and view expenses by category.

2.2 REFERENCES

[1] D. GRAZIANO, Gartner: Apple leads smartphone sales to new heights. BGR Media, <http://bgr.com/2012/02/15/gartner-apple-leads-smartphone-sales-to-new-heights/>, accessed October 2012, February 2012.

[2] M. BROWNLOW, Smartphone statistics and market share. Email-Marketing-Reports, <http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm>, accessed October 2012, October 2012.

[3] C. HEATH AND J. B. SOLL, Mental budgeting and consumer decisions, J. Consumer Res., 23 (1996), pp. 40-52.

[4] J. HASTINGS AND J. M. SHAPIRO, Mental accounting and consumer choice: Evidence from commodity price shocks. Unpublished report, 2012.

[5] MINT, Homepage. Mint, <https://www.mint.com>, accessed October 2012, n.d

[6] APPLE, iOS technology overview. Apple, <http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>, accessed October 2012, n.d.

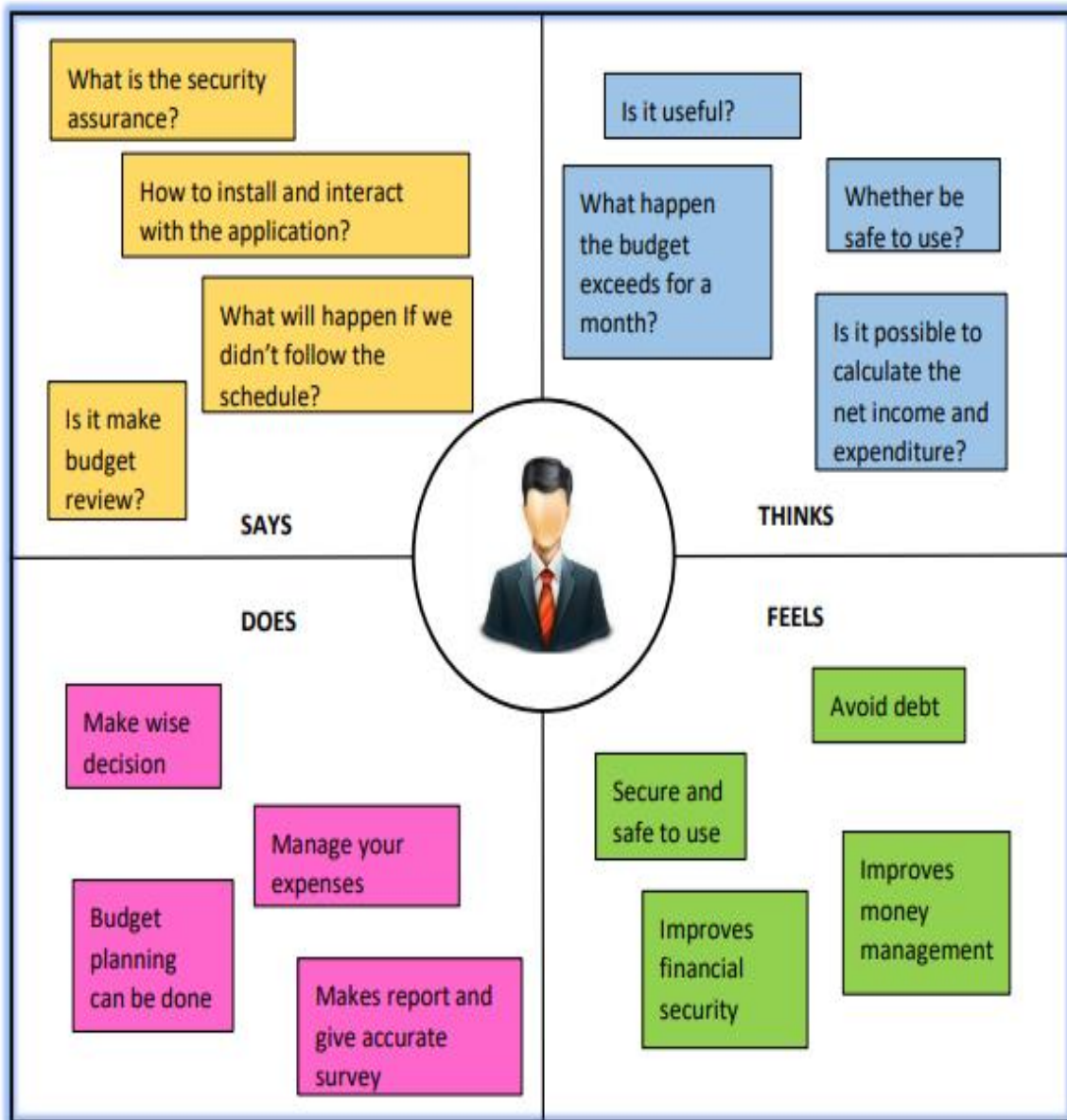
[7] APPLE, Key objects of an iOS app. Apple, http://developer.apple.com/library/IOs/#documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/AppArchitecture/AppArchitecture.html#//apple_ref/doc/uid/TP40007072-CH3-SW2, accessed October 2012, n.d.

2.3 PROBLEM STATEMENT DEFINITION

- At the end of the month we start to have money crisis.
- Lack of proper planning of our income.
- Person has to keep a log in a diary or in a computer.
- All the calculations need to be done by the user.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING

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

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](#)

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

FOCUS

How to allow the users a simple way to track all their expenses and how we define a reminder system for the same

Brainstorm

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

10 minutes

THAMO

Navigate to the dashboard

Add income and expenses

Edit User Profile

Add reminder and get notify

Visualize the expenses

Set budget

NANDHA

Filter the expenses graphically

Edit income and expenses

Create a additional stream of income

Share cash flow

Keep accurate records

Generate Monthly report

NAVEEN

Organize budget or create new budget

We need for conditional Social-views

Feedback System

Get monthly report in pdf or excel view

Categorize your expenses

Increasing a new category of items

SABARI

To record and manage the savings

Categorize the expenses

Filter the expenses periodically

Add multiple stream of income

Limitations for budget

Generate report and upload it as a new income source

Key rules of brainstorming

To run a smooth and productive session

Stay in topic. Encourage wild ideas.

Defer judgment. Listen to others.

Go for volume. If possible, be visual.

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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.

10 minutes

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.

10 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Keep accurate records

Generate Monthly report

Add income and expenses

Filter the expenses periodically

Add multiple stream of income

Set budget

Set smart budget to help you did not spend money in a chosen category

To record user to enter the savings

Categorize your expenses

Add reminder and get notify

Help you to stick on your budget and not to impulse spending

Edit User Profile

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

- A. Share the mural
Share a new link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
- B. Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
[Open the template](#)

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
[Open the template](#)

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
[Open the template](#)

[Share template feedback](#)

3.3 PROPOSED SOLUTION

S.no.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>Tracking expenses is one of the key factors in making budget work. At the instant, there is no as such complete solution present easily which enables a person to keep a track of its daily expenditure easily.</p> <p>To do so a person has to keep a log in a diary or in a computer, also all the calculations need to be done by the user which may sometimes results in errors leading to losses.</p> <p>Due to lack of a complete tracking system, there is a constant overload to rely on the daily entry of the expenditure.</p>
2.	Idea / Solution description	<p>In this project, we developed a cloud-based web application which keeps track of user's personal expenses.</p> <p>This system attempts to free the user with as much as possible the burden of manual calculation and to keep the track of the expenditure. This system also eliminates sticky notes, bills.</p>

3.	Novelty / Uniqueness	<p>This personal expense tracker Application has features that enables the user to have an option to set a limit for the amount to be used.</p> <p>If the limit is exceeded the user will be notified with an Email and SMS alert. This tracker doesn't have annoying ads.</p>
4.	Social Impact / Customer Satisfaction	<p>The user will be able to Stick to their Spending Limits. They can able to scan their bills anytime thus data loss is avoided.</p> <p>Users can also able to get an analysis of their expenditure in graphical forms.</p>
5.	Business Model (Revenue Model)	<p>This application will generate revenue by offering premium features to the user.</p> <p>Advertising through app is the easy way to earn money. Users may pay to remove the app advertisements.</p> <p>Through subscription the users can able to connect their bank account.</p>
6.	Scalability of the Solution	<p>Since this application is deployed on IBM Cloud, it can handle multiple users at a time.</p> <p>With our application, the users can be able to manage their expenses more effectively and know about their budget Vs income.</p>

3.4 PROBLEM SOLUTION FIT

Project Design Phase-I - Solution Fit Template

Project Title: Personal Expense Tracker Application

Team ID: PNT2022TMID17124

<p>Define CS, fit into CC</p> <p>Focus on ABT, tap into BE, understand RC</p>	<p>1. CUSTOMER SEGMENT(S)</p> <ul style="list-style-type: none"> ❖ Working peoples ❖ Organizations ❖ Students and families ❖ Common people with all ages can able to track their expenses. 	<p>5. CUSTOMER CONSTRAINTS</p> <ul style="list-style-type: none"> ❖ Network Issues ❖ Data Privacy ❖ Spending power ❖ Available devices 	<p>8. AVAILABLE SOLUTIONS</p> <p>People makes use of sticky notes or diary for calculating their expenditure.</p> <p>Pros:</p> <ol style="list-style-type: none"> 1. Didn't need any devices for calculations. <p>Cons:</p> <ol style="list-style-type: none"> 1. Time consuming. 2. Manual errors occur sometimes. 	<p>Explore AS, differentiate</p> <p>Focus on ABT, tap into BE, understand RC</p>
	<p>2. JOBS-TO-BE-DONE / PROBLEMS</p> <ul style="list-style-type: none"> ❖ People have to track their expenses regularly. ❖ They need to keep their receipts and bills which shows their amount they spent. ❖ Also they need to manually add or remove the desired categories. 	<p>6. PROBLEM ROOT CAUSE</p> <ul style="list-style-type: none"> ❖ The root cause for this problem is the delay in the budget. ❖ There may be a chance of getting errors in human calculations. ❖ No one alerts if their spending exceeds particular limit. ❖ They do not have enough time for calculating their expenditure. 	<p>9. BEHAVIOUR</p> <ul style="list-style-type: none"> ❖ People should know their budget for each month and set appropriate saving goals. ❖ Collect receipts regularly without fail. 	

<p>3. TRIGGERS</p> <ul style="list-style-type: none"> ❖ Realizing that excessive spending leading to lack of money in case of emergencies. ❖ Lack of Budgeting knowledge. 	<p>7. YOUR SOLUTION</p> <ul style="list-style-type: none"> ❖ A cloud-based web application which keeps track of user's personal expenses. This system attempts to free the user with as much as possible the burden of manual calculation and to keep the track of the expenditure. ❖ User just need to enter their day-to-day expenses. They also have an option to set the limit. If their expenditure exceeds that limit, notification will be sent through mail. ❖ This system also eliminates sticky notes, bills. 	<p>10. CHANNELS OF BEHAVIOUR</p> <p>ONLINE</p> <ul style="list-style-type: none"> ❖ Provide the details of day-to-day expenses. ❖ Select the area where customers use. ❖ Maintain the expenses for budgeting. <p>OFFLINE</p> <ul style="list-style-type: none"> ❖ Maintain the required documents regularly. ❖ Inspect the expenses for budgeting.
<p>4. EMOTIONS: BEFORE / AFTER</p> <p>Before</p> <ul style="list-style-type: none"> ❖ Excessive expenditure ❖ Afraid of spending <p>After</p> <ul style="list-style-type: none"> ❖ Being aware of what they are spending. ❖ Satisfied and happy with their budget expenditure. ❖ There will not be any frustrations any more since the process is quick and flexible. 		

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

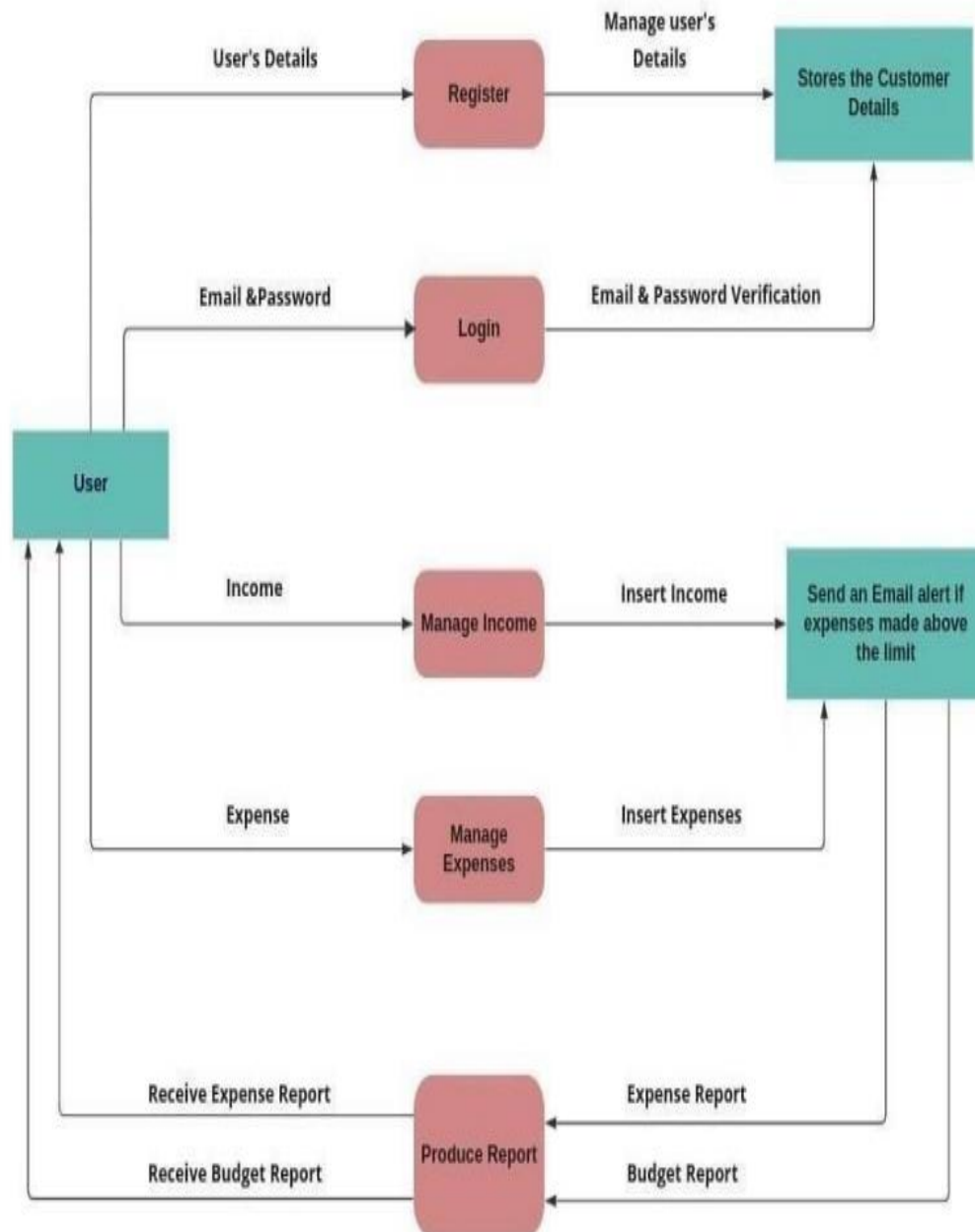
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Financial Accounts	Account Details Verification of Details
FR-4	User Dashboard	Expense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

4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	By using this application, the user can keep track of their expenses and can ensure that user's money is used wisely.
NFR-2	Security	Maintain user personal details in a encrypted manner by using data security algorithms .
NFR-3	Reliability	It will maintain a proper tracking of day-to-day expenses in an efficient manner.
NFR-4	Performance	By enter our incoming and departing cash, and the software can help you keep and monitor it with at-most quality and security with high performance.
NFR-5	Availability	Using charts and graphs may help you monitor your budgeting and assets.
NFR-6	Scalability	Rely on your budgeting app to track, streamline, and automate all the recurrent expenses and remind you on a timely basis.

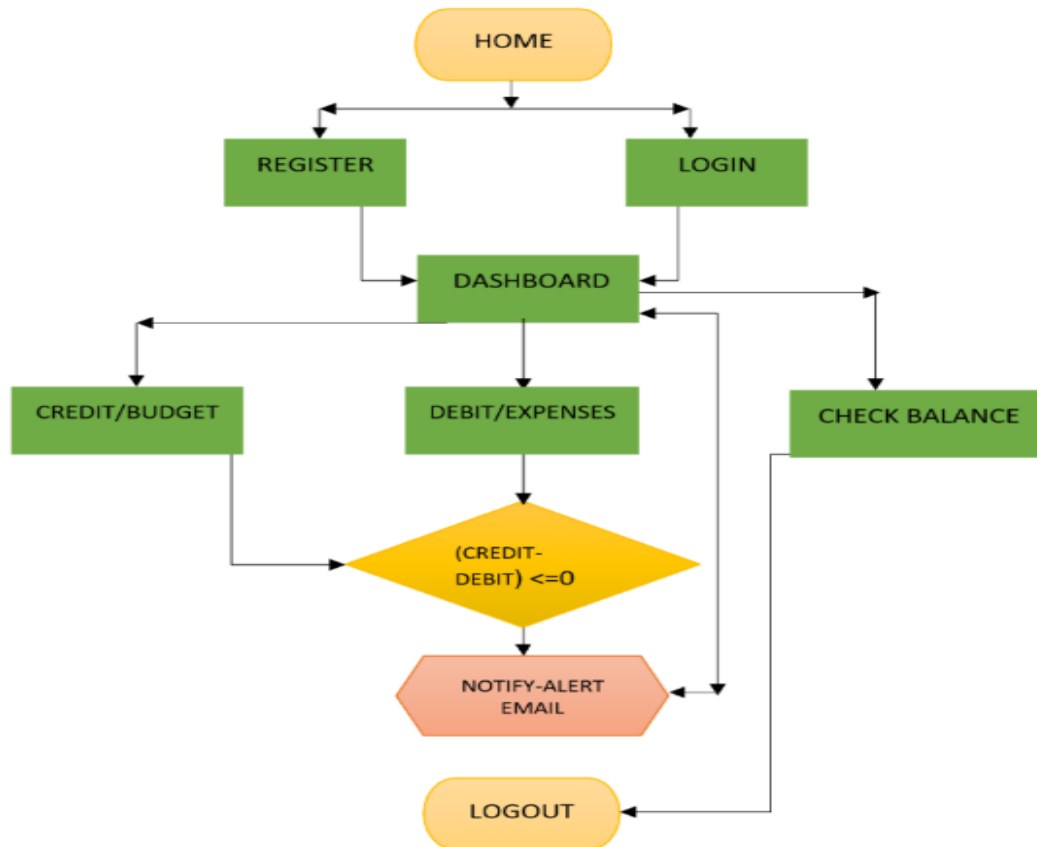
5. PROJECT DESIGN

5.1 Data Flow Diagrams



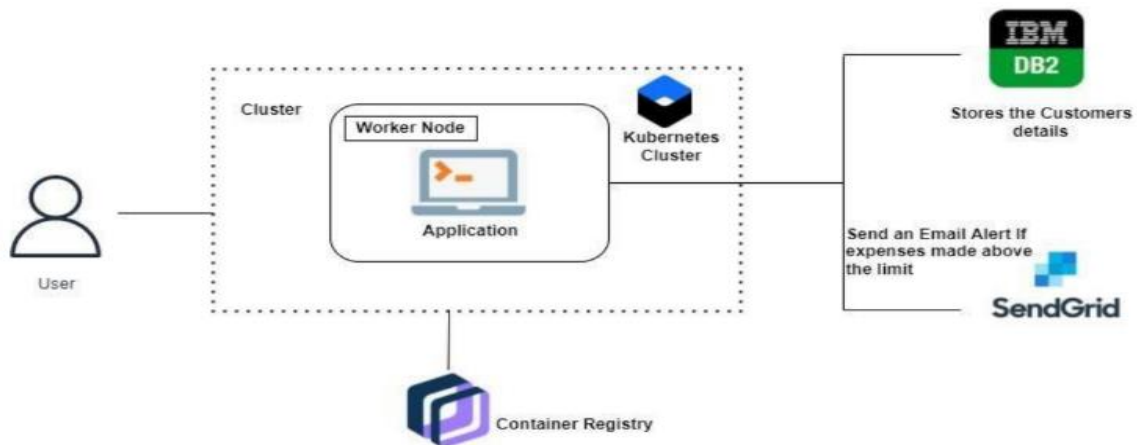
5.2 Solution & Technical Architecture

Solution Architecture:



Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table 1 & table 2



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
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
	Login	USN -2	As a user, I can log into the application by entering email & password	I can access Theapplication	High
	Dashboard	USN -3	As a user I can enter my income and expenditure details.	I can view my Dailyexpenses	High
Customer Care Executive		USN -4	As a customer care executive, I can solve the log in issues and	I can provide support or solution at any time 24*7	Medium

			other issues of the application.		
Administrator	Application	USN-5	As an administrator I can upgrade or update the application.	I can fix the bug which arises for the customers and users of the application	Medium

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

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	23 Oct 2022	28 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	30 Oct 2022	04 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	06 Nov 2022	11 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	18 Nov 2022	20	19 Nov 2022

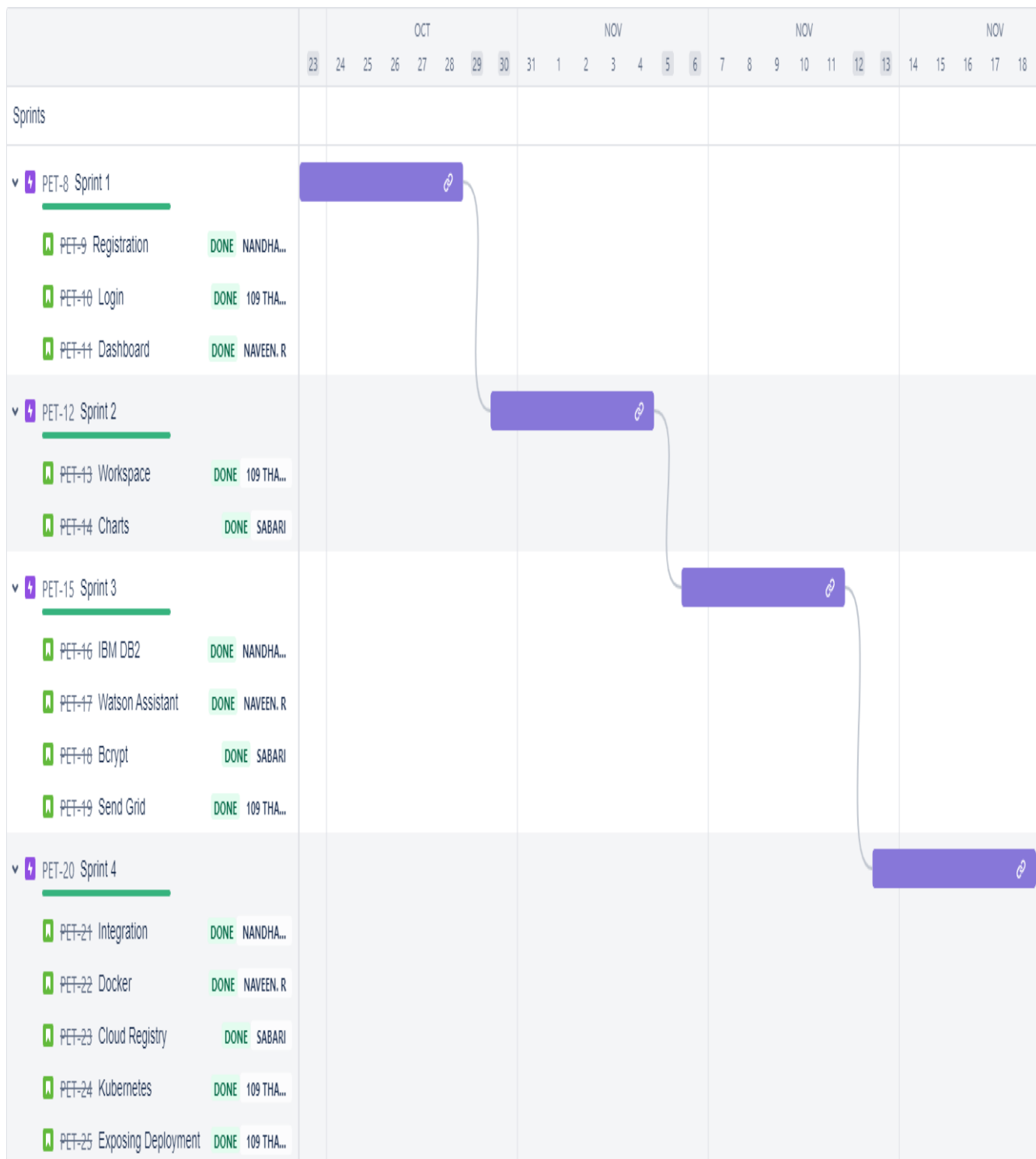
6.2 Sprint Delivery Schedule

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 , new password and confirming the same password.	2	High	Thamodharan S Nandha Kumar M Naveen R
		USN -2	As a user , I will receive confirmation email once I have registered for the application.	1	Low	Sabari K
	Login	USN -3	As a user , I can log into the application by entering email and password / Google OAuth.	2	High	Thamodharan S Nandha Kumar M
	Dashboard	USN -4	Logging in takes the user to their dashboard.	1	Low	Naveen R Sabari K
Sprint - 2		USN -5	As a user ,I will update my salary at the start of each month.	1	Medium	Nandha Kumar
		USN -6	As a user , I will set a target/limit to keep track of my expenditure.	1	Medium	Naveen
	Workspace	USN -7	Workplace for personal expense tracking	1	Medium	Thamodharan S
	Charts	USN -8	Graphs to show weekly and everyday expenditure	2	High	Naveen

		USN -9	As a user , I can export raw data as csv file.	1	Medium	Nandha Kumar
--	--	--------	--	---	--------	--------------

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint - 3	IBM DB2	USN -10	Linking database with dashboard	2	High	Nandha Kumar
		USN -11	Making dashboard interactive with JS	2	High	Thamodharan
	Watson Assistant	USN -12	Embedding Chatbot to clarify user's queries.	1	Low	Naveen
	BCrypt	USN -13	Using BCrypt to store passwords securely.	1	Medium	Sabari
	SendGrid	USN -14	Using SendGrid to send mail to the user. (To alert or remind)	1	Medium	Nandha Kumar
Sprint - 4	Integration	USN -15	Integrating frontend and backend.	2	High	Thamodharan
	Docker	USN -16	Creating Docker image of web app.	2	High	Naveen
	Cloud Registry	USN -17	Uploading docker image to IBM cloud registry.	2	High	Sabari
	Kubernetes	USN -18	Creating container using docker and hosting the webapp.	2	High	Naveen
	Exposing Deployment	USN -19	Exposing IP/Ports for the site.	1	Medium	Nandha Kumar

6.3 Reports from JIRA



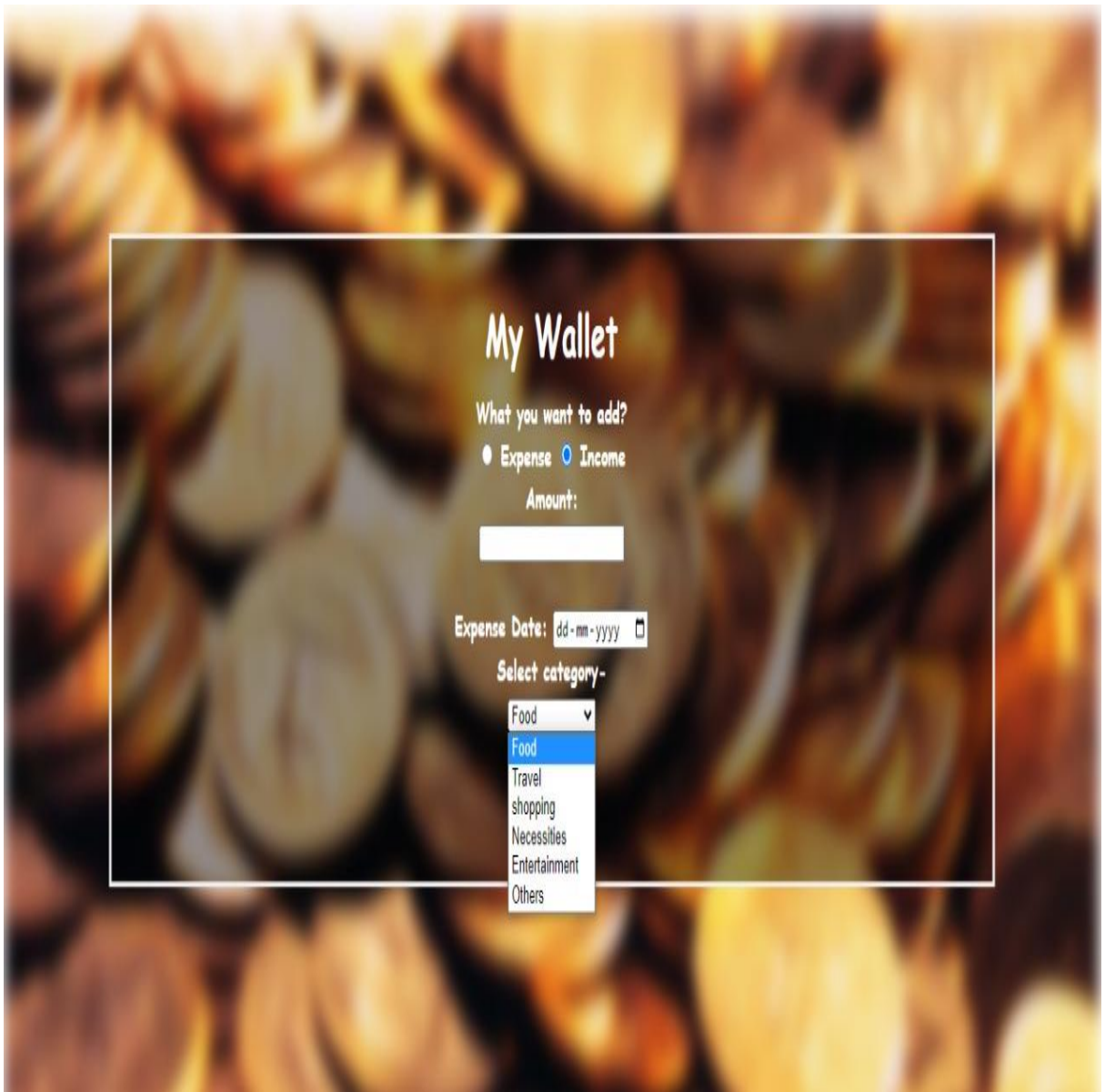
7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

```
templates > home > login.html > div.bg-text > body.bg-primary > div#layoutAuthentication > div#layoutAuthentication_content > main > div.container
1 {% extends 'base.html' %}
2 {% block body %}
3 <div class="bg-image"></div>
4
5 <div class="bg-text">
6
7     <body class="bg-primary">
8         <div id="layoutAuthentication">
9             <div id="layoutAuthentication_content">
10                 <main>
11                     <div class="container">
12                         <div class="row justify-content-center">
13                             <div class="col-lg-5">
14                                 <div class="card shadow-lg border-0 rounded-lg mt-5">
15                                     <div class="card-header">
16                                         <h3 class="text-center font-weight-light my-4">Expense Tracker by
17                                             DataFlair-Login</h3>
18                                     </div>
19                                     <div class="card-body">
20                                         <form action="/handlelogin/" method="post">
21                                             {% csrf_token %}
22                                             <div class="form-group">
23                                                 <label class="small mb-1" for="loginusername">Username</label>
24                                                 <input class="form-control py-4" id="loginusername" name="loginusername"
25                                                     type="text" placeholder="Enter user name" required />
26                                             </div>
27                                             <div class="form-group">
28                                                 <label class="small mb-1" for="loginpassword1">Password</label>
29                                                 <input class="form-control py-4" id="loginpassword1"
30                                                     name="loginpassword1" type="password" placeholder="Enter password"
31                                                     required />
32                                             </div>
33                                         </form>
34                                     </div>
35                                 </div>
36                             </div>
37                         </div>
38                     </div>
39                 </main>
40             </div>
41         </div>
42     </body>
43 </div>
44 {% endblock body %}
```

```

33                                         </form>
34                                     </div>
35                                 </div>
36                             </div>
37                         </div>
38                     </div>
39                 </main>
40             </div>
41         </div>
42     </body>
43 </div>
44 {% endblock body %}
45
46 <div class="form-group">
47     <div class="custom-control custom-checkbox">
48         <input class="custom-control-input" id="rememberPasswordCheck"
49             type="checkbox" required />
50         <label class="custom-control-label"
51             for="rememberPasswordCheck">Remember password</label>
52     </div>
53 </div>
54 <div class="form-group d-flex align-items-center justify-content-between mt-4 mb-0">
55     <div class="btnn">
56         <button type="submit" class="btn">Login</button><br>
57     </div>
58     <div class="link">
59         <a class="small" href="/reset_password">Forgot Password?</a><br>
60
61         <div class="card-footer text-center">
62             <div class="small"><a href="/register">Need an account? Sign
63                 up!</a></div>
64         </div>
65     </div>
66 </div>
67 </form>
68 </div>
69 </main>
70 </div>
71 </body>
72 {% endblock body %}
```

My Wallet

What you want to add?

☒ Expense ☐ Income

Amount:

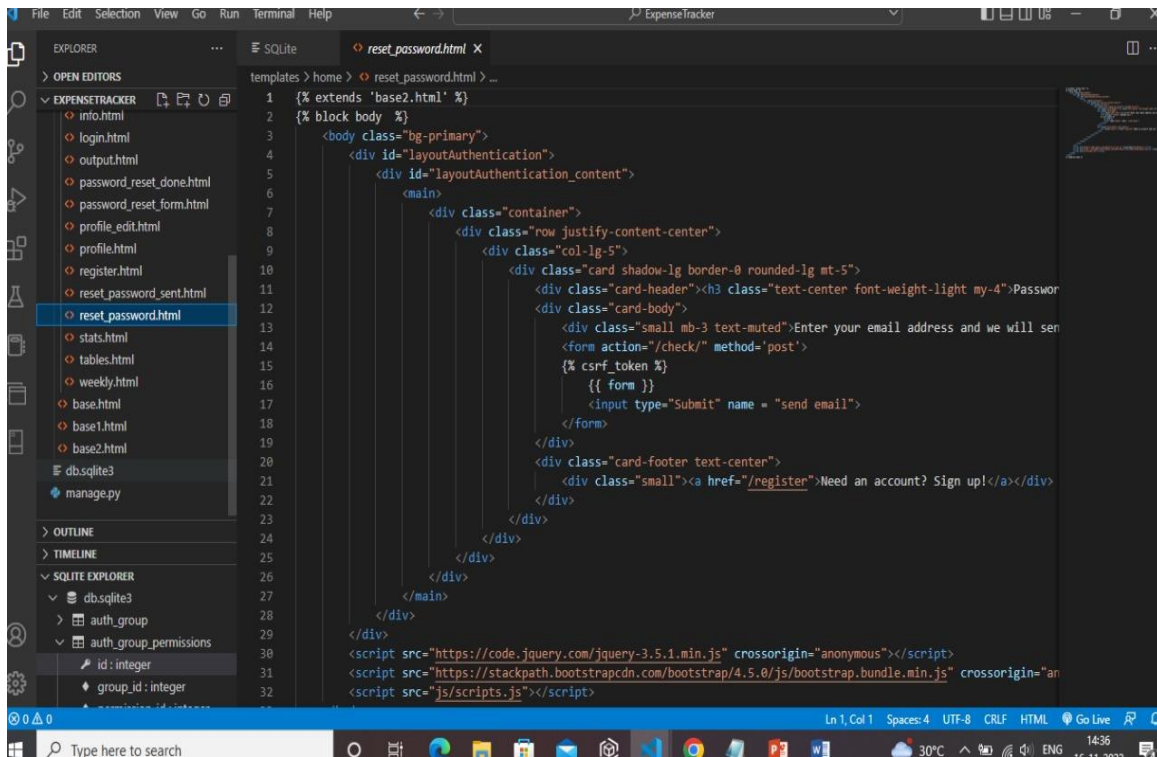
Expense Date:

Select category-

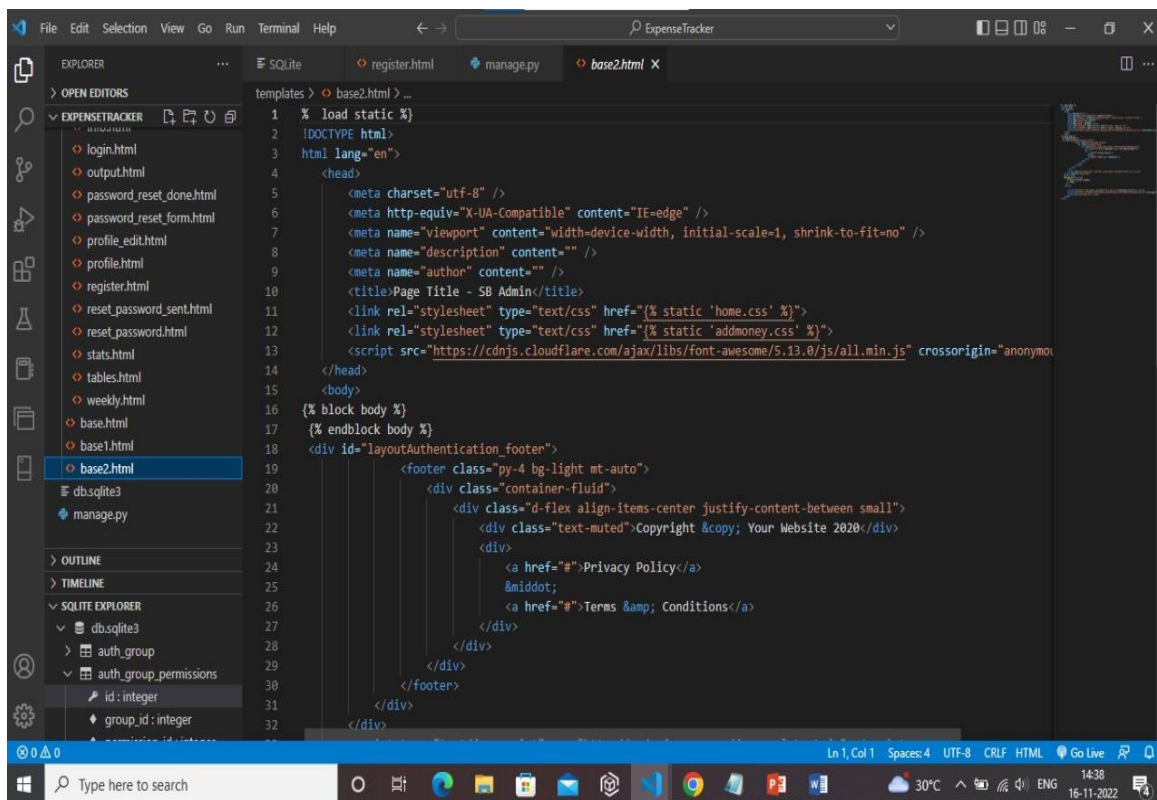
- Food
- Food
- Travel
- shopping
- Necessities
- Entertainment
- Others

7.2 Feature

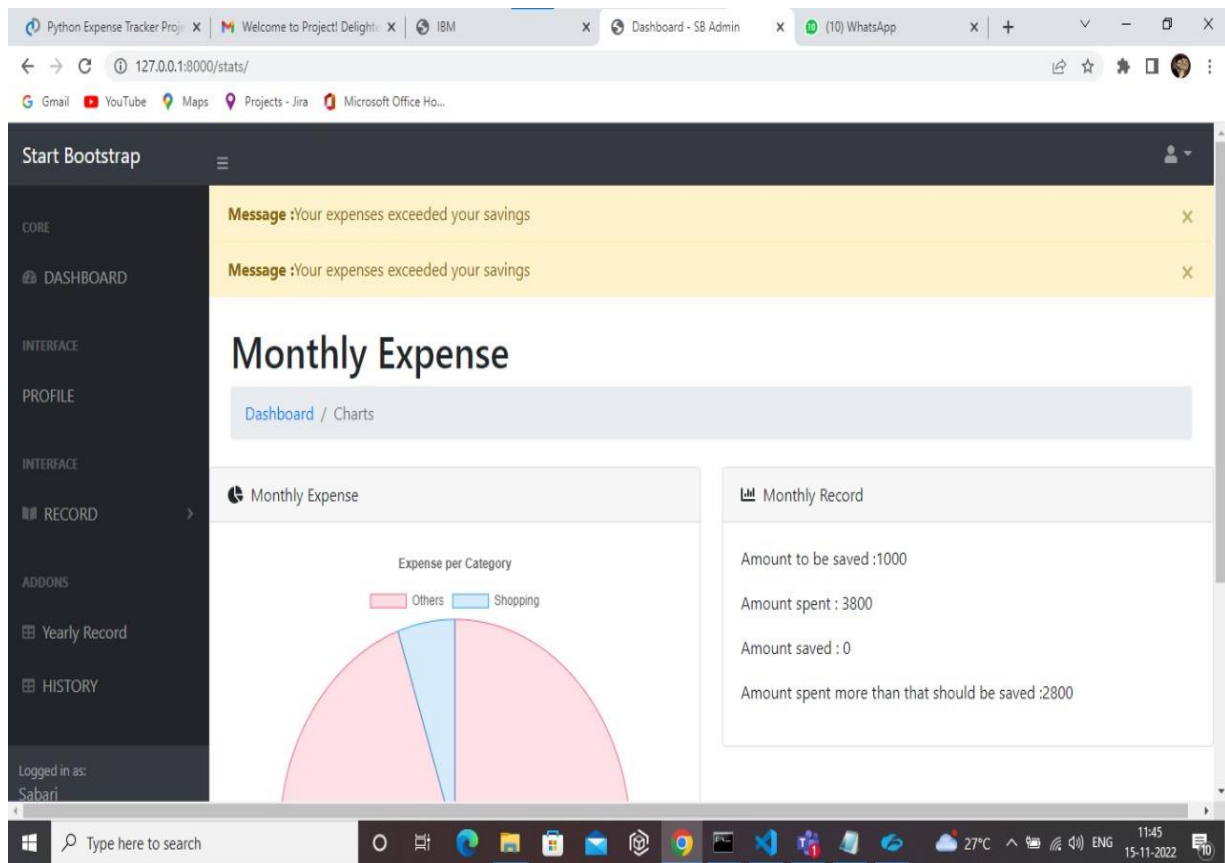
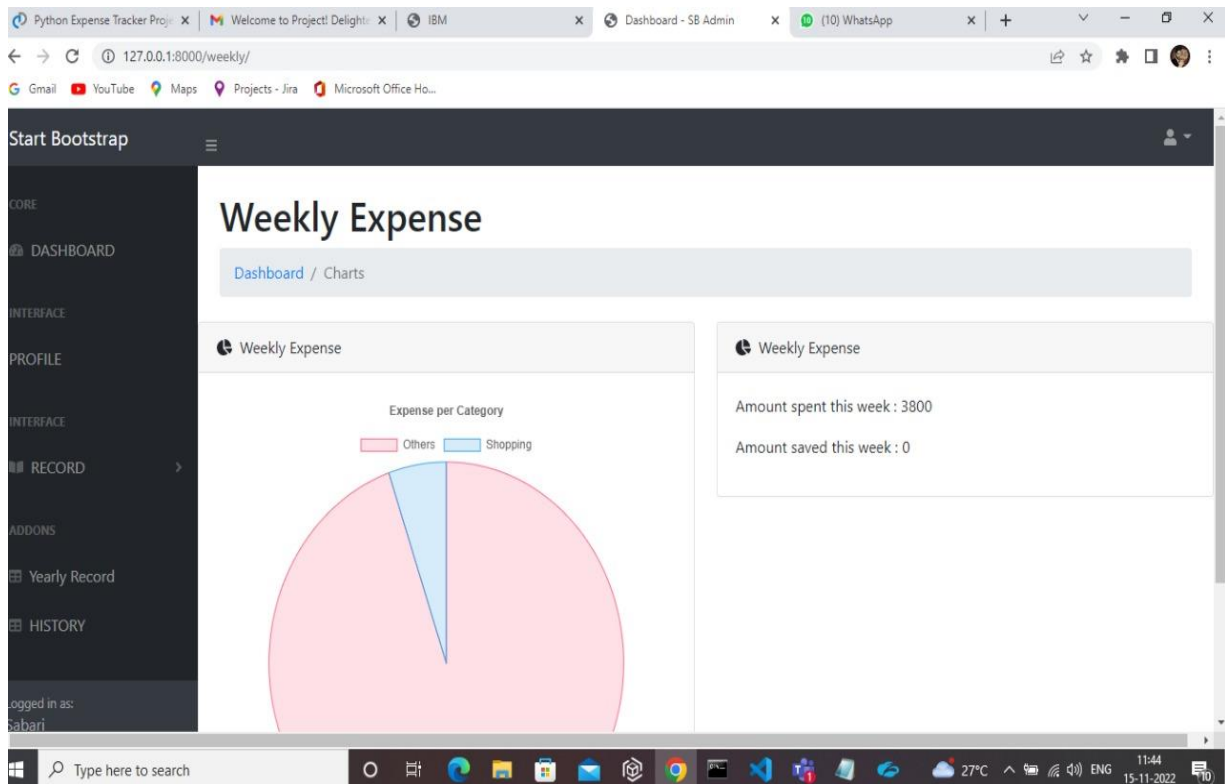
2



```
1 {% extends 'base2.html' %}
2 {% block body %}
3 <body class="bg-primary">
4   <div id="layoutAuthentication">
5     <div id="layoutAuthentication_content">
6       <main>
7         <div class="container">
8           <div class="row justify-content-center">
9             <div class="col-lg-5">
10               <div class="card shadow-lg border-0 rounded-lg mt-5">
11                 <div class="card-header"><h3 class="text-center font-weight-light my-4">Password
12                 <div class="card-body">
13                   <div class="small mb-3 text-muted">Enter your email address and we will send
14                   <form action="/check/" method="post">
15                     {{ csrf.token %}}
16                     {{ form }}
17                     <input type="Submit" name = "send email">
18                   </form>
19                 </div>
20                 <div class="card-footer text-center">
21                   <div class="small"><a href="/register">Need an account? Sign up!</a></div>
22                 </div>
23               </div>
24             </div>
25           </div>
26         </div>
27       </main>
28     </div>
29   </div>
30   <script src="https://code.jquery.com/jquery-3.5.1.min.js" crossorigin="anonymous"></script>
31   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js" crossorigin="an
32   <script src="js/scripts.js"></script>
```



```
1 % load static %}
2 <!DOCTYPE html>
3 <html lang="en">
4   <head>
5     <meta charset="utf-8" />
6     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
8     <meta name="description" content="" />
9     <meta name="author" content="" />
10    <title>Page Title - SB Admin</title>
11    <link rel="stylesheet" type="text/css" href="{% static 'home.css' %}">
12    <link rel="stylesheet" type="text/css" href="{% static 'addmoney.css' %}">
13    <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/js/all.min.js" crossorigin="anonymou
14  </head>
15  <body>
16    {% block body %}
17    {% endblock body %}
18    <div id="layoutAuthentication_footer">
19      <footer class="py-4 bg-light mt-auto">
20        <div class="container-fluid">
21          <div class="d-flex align-items-center justify-content-between small">
22            <div class="text-muted">Copyright &copy; Your Website 2020</div>
23            <div>
24              <a href="#">Privacy Policy</a>
25              &mdot;
26              <a href="#">Terms & Conditions</a>
27            </div>
28          </div>
29        </div>
30      </div>
31    </div>
32  </div>
```



7.3 Database Schema (if Applicable)

The screenshot displays a database management interface with two tables. The top table, titled 'SQL', shows a list of users with columns: last_login, is_superuser, username, first_name, email, is_staff, is_active, date_joined, and last_name. The bottom table, also titled 'SQL', shows a list of permissions with columns: id, content_type_id, codename, and name. The interface includes a sidebar with a tree view of the database schema, a search bar, and a status bar at the bottom.

	last_login	is_superuser	username	first_name	email	is_staff	is_active	date_joined	last_name
ixA=	NULL	1	ishika			1	1	2021-07-19 07:49:43.793715	
XM=	2021-07-19 08:35:56.300129	0	DataFlair	Data	dataflair@gmail.com	0	1	2021-07-19 07:53:27.010905	Flair
Wp3wW5sUCz78=	2022-11-16 08:53:26.471540	0	Sabari3		sabarikishore0630@gmail.com	0	1	2022-11-15 09:52:10.026341	

id	content_type_id	codename	name
1	1	add_logentry	Can add log entry
2	1	change_logentry	Can change log entry
3	1	delete_logentry	Can delete log entry
4	1	view_logentry	Can view log entry
5	2	add_permission	Can add permission
6	2	change_permission	Can change permission
7	2	delete_permission	Can delete permission
8	2	view_permission	Can view permission
9	3	add_group	Can add group
10	3	change_group	Can change group
11	3	delete_group	Can delete group
12	3	view_group	Can view group
13	4	add_user	Can add user
14	4	change_user	Can change user
15	4	delete_user	Can delete user
16	4	view_user	Can view user
17	5	add_contenttype	Can add content type

File Edit Selection View Go Run Terminal Help ExpenseTracker

EXPLORER

- OPEN EDITORS
- EXPENSETRACKER
- OUTLINE
- TIMELINE
- SQLITE EXPLORER
 - db.sqlite3
 - auth_group
 - auth_group_permissions
 - auth_permission
 - auth_user
 - auth_user_groups
 - id: integer
 - user_id: integer
 - group_id: integer
 - auth_user_user_permission...
 - django_admin_log
 - django_content_type
 - django_migrations
 - django_session
 - home_addmoney_info
 - id: integer
 - add_money: varchar(10)
 - quantity: bigint
 - Date: date
 - Category: varchar(20)
 - user_id: integer
 - home_userprofile
 - id: integer
 - profession: varchar(10)

SQLite X

12	3	view_group	Can view group
13	4	add_user	Can add user
14	4	change_user	Can change user
15	4	delete_user	Can delete user
16	4	view_user	Can view user
17	5	add_contenttype	Can add content type
18	5	change_contenttype	Can change content type
19	5	delete_contenttype	Can delete content type
20	5	view_contenttype	Can view content type
21	6	add_session	Can add session
22	6	change_session	Can change session
23	6	delete_session	Can delete session
24	6	view_session	Can view session
25	7	add_userprofile	Can add user profile
26	7	change_userprofile	Can change user profile
27	7	delete_userprofile	Can delete user profile
28	7	view_userprofile	Can view user profile
29	8	add_addmoney_info	Can add addmoney_info
30	8	change_addmoney_info	Can change addmoney_info
31	8	delete_addmoney_info	Can delete addmoney_info
32	8	view_addmoney_info	Can view addmoney_info

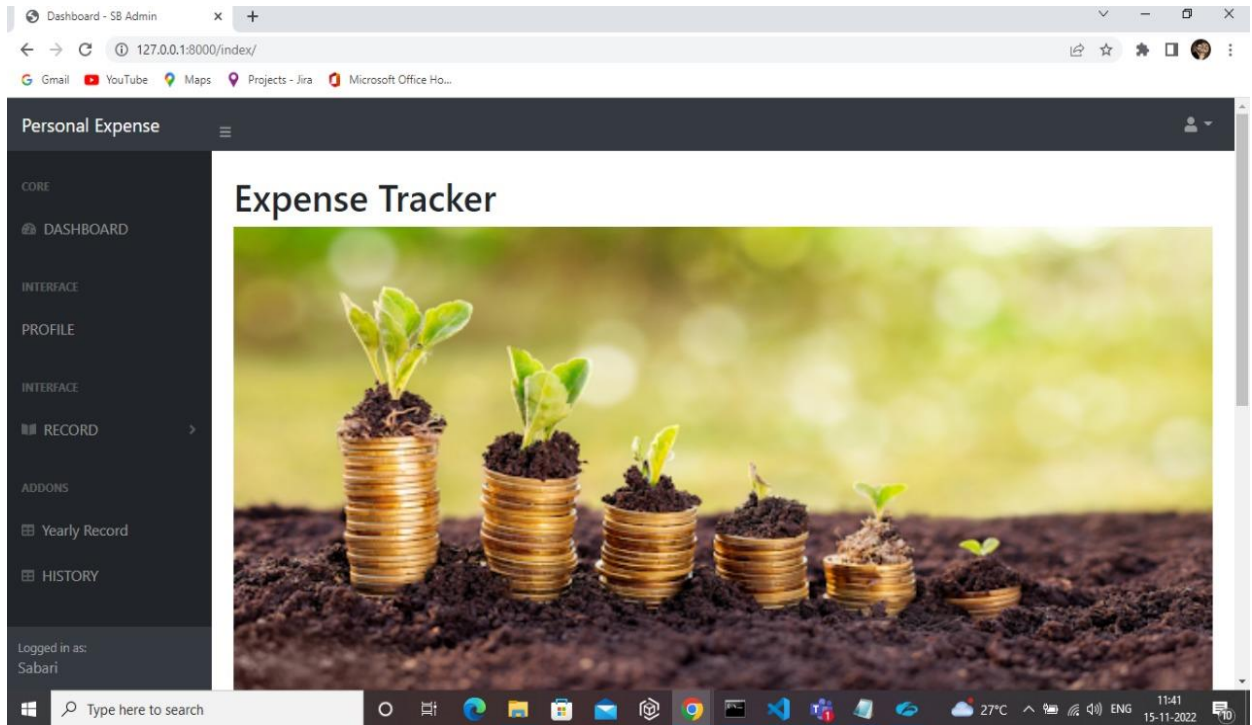
Go Live

Type here to search

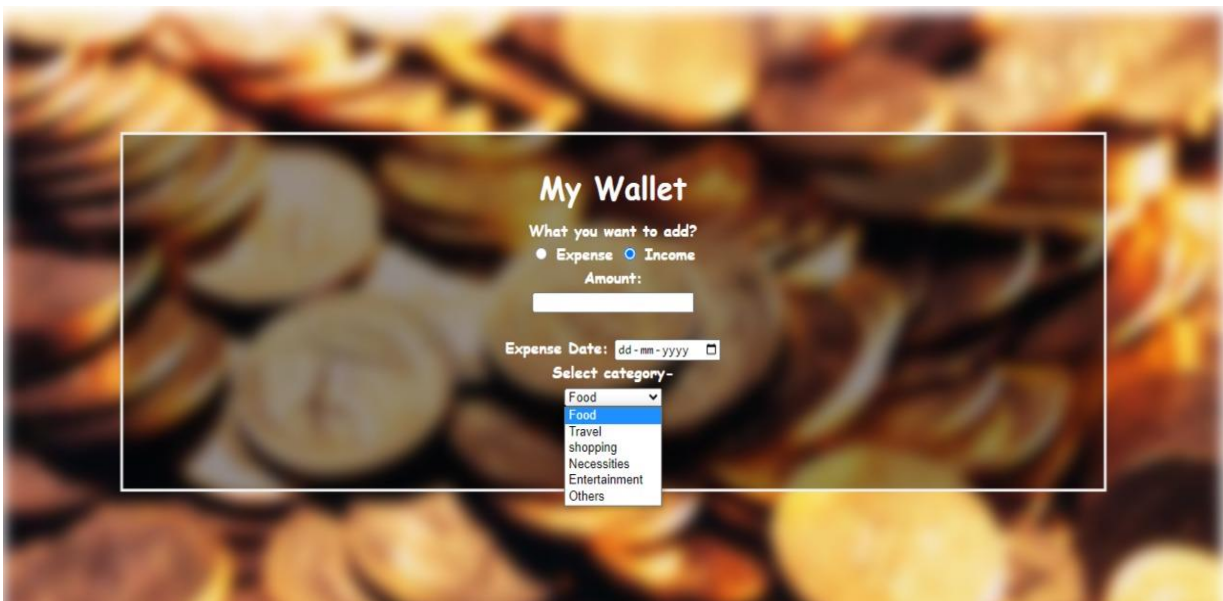
30°C 14:34 16-11-2022

8. TESTING

8.1 Test Cases



8.2 User Acceptance Testing



9. RESULTS

Daily spending tracking can not only help you save money, but it can also help you set financial objectives for the future. If you know exactly where your money goes each month, you can quickly see where you can make some savings and compromises. We have designed a project that is more efficient than other income and expense trackers. The project succeeds in avoiding manual calculations for estimating monthly revenue and expenses. The modules have been designed to be both efficient and appealing

10. ADVANTAGES

- Create a monthly budget
- Know where you're spending more than you actually think you are
- Figure out ways to cut back on your spending
- Know how much extra payments you can make towards your debt
- Plan for future large purchases
- Create a savings plan for putting money away every month
- Plan for retirement
- Create an investment strategy with extra money
- Prepare to file income taxes

11. CONCLUSION

We now present and discuss the limitations of the new product, issues faced, and the remedies to those limitations. While building this expense tracking software tool, the major focus was to make this tool less user intensive and more user productive. The first version of this application is only applicable to the USA. It could have been used in other countries if I would have used currency converters in the application, which I will improvise in the later version. Certain issues were faced while implementing this tool and various important things were kept in mind. For example, the user interface is designed simple yet creative so that the user doesn't face any difficulty in using the software tool and the expense data is persisted on the device even if the user deletes the application from the memory background. Core Data was chosen over SQLite to persist the data which is very beneficial even though the data would reside on the device locally. iCloud functionality is missing. This would allow data to be saved to a Cloud and shared between various devices. The application lacks the feature of PDF by which the list of expenses.

12. FUTURE SCOPE

- In this project, a rich model's design and implementation are discussed, allowing third-party cloud apps to access a client's virtual machines (VMs) and carry out privileged operations. The infrastructure support required to support cloud apps was discussed. Different design approaches to deploy cloud apps were also addressed. Various examples were used to demonstrate and assess the practicality of cloud applications.
- In order to address the primary financial issues, this study looked at the potential usefulness of the mobile application "Manage on Money (MoM)". OCR technology was created using Google Cloud Vision API. This technique works well for locating a single precise keyword on a receipt printed in black ink. MoM enables users to arrange their recurring expenses and sends a push reminder prior to the due date. OneSignal API serves as the foundation for this notification.

- This application allows the user to enter their income to determine their daily expenses, and the results are saved for each user. The application has a feature that uses data mining to predict the manager's income and expenses. The suggested system takes less time to process, and all the information is updated and processed right away.

13. **APPENDIX** Source Code

GitHub & Project Demo Link

<https://github.com/IBM-EPBL/IBM-Project-4914-1658742869>

<https://youtu.be/4tvpKvfoZVY>

BASE.HTML

```
{% load static %}

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>

    <meta name="description" content="" />

    <meta name="author" content="" />

    <title>Page Title - SB Admin</title>

    {% comment %} <link rel="stylesheet" type="text/css" href="{ % static 'styles.css' % }">
{% endcomment %}

    <link rel="stylesheet" type="text/css" href="{ % static 'login1.css' % }">

    <link rel="preconnect" href="https://fonts.gstatic.com">

    <link
href="https://fonts.googleapis.com/css2?family=Big+Shoulders+Stencil+Text:wght@700&di
splay=swap" rel="stylesheet">

    <script    src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/js/all.min.js"
crossorigin="anonymous"></script>

  </head>

  <body>
```

```

<div class="messages" id="alert-message">

{% for message in messages %}

<div class="alert alert-{{ message.tags }} mb-0 alert-dismissible fade show" role="alert">

<strong>Message :</strong>{{ message }}

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

{% endfor %}

</div>

{% block body %}

{% endblock body %}

<div id="layoutAuthentication_footer">

<footer class="py-4 bg-light mt-auto">

<div class="container-fluid">

<div class="d-flex align-items-center justify-content-between small">

{% comment %} <div class="text-muted">Copyright &copy; Your Website
2020</div>

<div>

<a href="#">Privacy Policy</a>

&middot;

<a href="#">Terms & Conditions</a>

</div>

</div> {% endcomment %}

</div>

```

```

        </footer>

    </div>

</div>

<script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>

<script type="text/javascript">

$(function(){

    setTimeout(function(){

        $("#alert-message").hide();

        }, 2000);

    });

</script>

<script                                src="https://code.jquery.com/jquery-3.5.1.min.js"
crossorigin="anonymous"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>

<script src="js/scripts.js"></script>

</body>

</html>

```

BASE1.HTML

```

{ % load static % }

<!DOCTYPE html>

<html lang="en">

```

```

<head>

    <meta charset="utf-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>

    <meta name="description" content="" />

    <meta name="author" content="" />

    <title>Page Title - SB Admin</title>

    <link rel="stylesheet" type="text/css" href="{ % static 'registration.css' % }">

    <script    src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/js/all.min.js"
crossorigin="anonymous"></script>

</head>

<body>

    <div class="messages" id="alert-message">

        { % for message in messages % }

        <div class="alert alert-{ { message.tags } } mb-0 alert-dismissible fade show" role="alert">

            <strong>Message :</strong>{ { message } }

            <button type="button" class="close" data-dismiss="alert" aria-label="Close">

                <span aria-hidden="true">&times;</span>

            </button>

        </div>

        { % endfor % }

    </div>

    { % block body % }

    { % endblock body % }

```

```

<div id="layoutAuthentication_footer">

    <footer class="py-4 bg-light mt-auto">

        <div class="container-fluid">

            <div class="d-flex align-items-center justify-content-between small">

                <div class="text-muted">Copyright &copy; Your Website 2020</div>

                <div>

                    <a href="#">Privacy Policy</a>

                    &middot;

                    <a href="#">Terms & Conditions</a>

                </div>

            </div>

        </div>

    </div>

    </div>

    </div>

    <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>

    <script type="text/javascript">

        $(function(){

        setTimeout(function(){

            $("#alert-message").hide();

            }, 2000);

        });

    </script>

```

```

        <script                                src="https://code.jquery.com/jquery-3.5.1.min.js"
crossorigin="anonymous"></script>

        <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>

        <script src="js/scripts.js"></script>

</body>

</html>

```

BASE2.HTML

```

{ % load static % }

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8" />

        <meta http-equiv="X-UA-Compatible" content="IE=edge" />

        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>

        <meta name="description" content="" />

        <meta name="author" content="" />

        <title>Page Title - SB Admin</title>

        <link rel="stylesheet" type="text/css" href="{ % static 'home.css' % }">

        <link rel="stylesheet" type="text/css" href="{ % static 'addmoney.css' % }">

```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/js/all.min.js"
crossorigin="anonymous"></script>
```

```
</head>
```

```
<body>
```

```
{ % block body % }
```

```
{ % endblock body % }
```

```
<div id="layoutAuthentication_footer">
```

```
<footer class="py-4 bg-light mt-auto">
```

```
<div class="container-fluid">
```

```
<div class="d-flex align-items-center justify-content-between small">
```

```
<div class="text-muted">Copyright &copy; Your Website 2020</div>
```

```
<div>
```

```
<a href="#">Privacy Policy</a>
```

```
&middot;
```

```
<a href="#">Terms & Conditions</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</footer>
```

```
</div>
```

```
</div>
```

```
<script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>
```

```
<script type="text/javascript">
```

```
$(function(){
```

```
setTimeout(function(){
```



```

        $("#alert-message").hide();

    }, 2000);

});

</script>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"
crossorigin="anonymous"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>

<script src="js/scripts.js"></script>

</body>

</html>

```

TABLES.HTML

```

{ % load static % }

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>

    <meta name="description" content="" />

```

```

<meta name="author" content="" />

<title>Tables - SB Admin</title>

<link rel="stylesheet" type="text/css" href="{ % static 'home.css' % }">

<link      href="https://cdn.datatables.net/1.10.20/css/dataTables.bootstrap4.min.css"
rel="stylesheet" crossorigin="anonymous" />

<script    src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/js/all.min.js"
crossorigin="anonymous"></script>

</head>

<body class="sb-nav-fixed">

<nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">

    <a class="navbar-brand" href="index.html">Start Bootstrap</a>

    <button class="btn btn-link btn-sm order-1 order-lg-0" id="sidebarToggle"
href="#"><i class="fas fa-bars"></i></button>

    <!-- Navbar Search-->

    <form class="d-none d-md-inline-block form-inline ml-auto mr-0 mr-md-3 my-2 my-
md-0" method="POST">

        { % csrf_token % }

        <div class="input-group">

            </div>

        </form>

    <!-- Navbar-->

    <ul class="navbar-nav ml-auto ml-md-0">

        <li class="nav-item dropdown">

```

```

        <a class="nav-link dropdown-toggle" id="userDropdown" href="#"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false"><i class="fas fa-user fa-fw"></i></a>

        <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="userDropdown">

            <a class="dropdown-item" href="#">Settings</a>

            <a class="dropdown-item" href="#">Activity Log</a>

            <div class="dropdown-divider"></div>

            <a class="dropdown-item" href="/handleLogout">Logout</a>

        </div>

    </li>

</ul>

</nav>

<div id="layoutSidenav">

    <div id="layoutSidenav_nav">

        <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">

            <div class="sb-sidenav-menu">

                <div class="nav">

                    <div class="sb-sidenav-menu-heading">Core</div>

                    <a class="nav-link" href="/index">

                        <div class="sb-nav-link-icon"><i class="fas fa-tachometer-
alt"></i></div>

                        Dashboard

                    </a>

                    <div class="sb-sidenav-menu-heading">Interface</div>

```

<div>

<i class="glyphicon glyphicon-user"></i>

PROFILE

<div class="sb-sidenav-menu-heading">Interface</div>

<div class="sb-nav-link-icon"><i class="fas fa-book-open"></i></div>

RECORD

<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>

<div class="collapse" id="collapsePages" aria-labelledby="headingTwo" data-parent="#sidenavAccordion">

<nav class="sb-sidenav-menu-nested nav accordion" id="sidenavAccordionPages">

WEEKLY RECORD

<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>

MONTHLY RECORD

<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>

```

        </a>

    </nav>

</div>

<div class="sb-sidenav-menu-heading">Addons</div>

{ % comment % } <a class="nav-link" href="charts.html">

    <div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>

    Charts

</a> { % endcomment % }

<a class="nav-link" href="/info">

    <div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>

    Yearly Record

</a>

<a class="nav-link" href="/tables">

    <div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>

    History

</a>

</div>

</div>

<br>

<br>

<br>

<br>

<br>

<br>

<div class="sb-sidenav-footer">

```

```

        <div class="small">Logged in as:</div>

        { {request.user.username} }

    </div>

</nav>

</div>

<div id="layoutSidenav_content">

    <main>

        <div class="container-fluid">

            <h1 class="mt-4">History</h1>

            <ol class="breadcrumb mb-4">

                <li class="breadcrumb-item"><a href="/index">Dashboard</a></li>

                <li class="breadcrumb-item active">HISTORY</li>

            </ol>

            <form action="/search/" method="get">

                From : <input type="date" name="fromdate" required/>

                To : <input type="date" name="todate" required/>

                { % csrf_token % }

                <input type="submit" value="search"/>

                <a href="{ % url 'tables' % }" class="btn btn-success" role="button"
                aria-pressed="true" style="height:32px;width:100px">Search all</a>

            </a>

            </form>

            <br/>

            <br>

            <div class="card mb-4">

```

```

<div class="card-header">

    <i class="fas fa-table mr-1"></i>

    DataTable

</div>

<div class="card-body">

    <div class="table-responsive">

        <table      class="table      table-bordered"      id="dataTable"
width="100%" cellspacing="0">

            <thead>

                <tr>

                    <th>What you added</th>

                    <th>Amount</th>

                    <th>Category</th>

                    <th>Date</th>

                </tr>

            </thead>

            { % for money in addmoney % }

            <tbody>

                <tr>

                    <td>{ { money.add_money } }</td>

                    <td>{ { money.quantity } }</td>

                    <td>{ { money.Category } }</td>

                    <td>{ { money.Date } }</td>

                </tr>

                { % endfor% }


```

```

        </tbody>

    </table>

</div>

</div>

</div>

</div>

</main>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"
crossorigin="anonymous"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>

<script src="js/scripts.js"></script>

<script src="https://cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js"
crossorigin="anonymous"></script>

<script src="https://cdn.datatables.net/1.10.20/js/dataTables.bootstrap4.min.js"
crossorigin="anonymous"></script>

<script src="assets/demo/datatables-demo.js"></script>

</body>

</html>

```


ADDMMONEY.HTML

```
{% load static% }

<!DOCTYPE html>

<html lang="en">

<head>

    {% comment %}

    <link rel="stylesheet" type="text/css" href=try.css> {% endcomment %}

    <link rel="stylesheet" type="text/css" href="{% static 'addmoney.css' %}">

    <link rel="preconnect" href="https://fonts.gstatic.com">

    <link          href="https://fonts.googleapis.com/css2?family=Anton&display=swap"
rel="stylesheet">

    <title>My Wallet</title>

</head>

<body>

    <div class="bg-image"></div>

    <div class="bg-text">

    <div id="layoutAuthentication">

    <div id="layoutAuthentication_content">

    <main>

    <div class="container2">

    <div class="row justify-content-center">
```

```

<div class="col-lg-5">

<div class="card shadow-lg border-0 rounded-lg mt-5">

<div class="card-header">

<h1 class="text-center font-weight-light my-4">My Wallet</h1>

</div>

<form method="post" action="/addmoney_submission/">

{ % csrf_token % }

<div class="form-check">

What you want to add?<br>

<label class="form-check-label" for="add_money">
<input class="form-check-input" type="radio" name="add_money"
id="add_money" value="Expense" checked required>
Expense</label>

<label class="form-check-label" for="add_money">
<input class="form-check-input" type="radio" name="add_money"
id="add_money" value="Income" checked required>Income</label>

</div>

<div><label for="quantity">Amount:</label><br>

<input type="number" name="quantity" required><br><br>

</div>

<div>

<label>

Expense Date:

<input type="date" name="Date">

```

</label>

</div>

<div>

<div class="f1">

 Select category-

<select class="Category" name="Category" required>

<option value="Food">Food</option>

<option value="Travel">Travel</option>

<option value="Shopping">shopping</option>

<option value="Necessities">Necessities</option>

<option value="Entertainment">Entertainment</option>

<option value="Others">Others</option>

</select>

</div>

</div>

<div class="btn">

<button type="submit" class="b2" href="/addmoney"> Submit </button>

```

</div>

<!--<input    type="button"    class="b2"        onclick="location.href='/addmoney'"
value="submit!">-->

</form>

</div>

</div>

</div>

</div>

</main>

        </div>

    </div>

</div>

</body>

</html>

```

REGISTER.HTML

```

{% extends 'base1.html' %}

{% block body %}

<div class="bg-image"></div>

<div class="bg-text">

    <body class="bg-primary">

        <div id="layoutAuthentication">

            <div id="layoutAuthentication_content">

                <main>

                    <div class="container" style="margin-bottom:15px;">

                        <div class="row justify-content-center">

                            <div class="col-lg-7">

                                <div class="card shadow-lg border-0 rounded-lg mt-5">

                                    <div class="card-header"><h1 class="text-center font-weight-light my-4">Expense Tracker by-DataFlair Create Account</h1></div>

                                    <div class="card-body">

                                        <form action="/handleSignup/" method='post'>

                                            {% csrf_token %}

                                            <div class="form-row">

                                                <div class="col-md-6">

                                                    <div class="form-group">

                                                        <label class="small mb-1" for="uname">User
Name</label>

                                                        <input class="form-control py-4" id="uname"
name="uname" type="text" placeholder="Enter User Name" required/>

                                                    </div>

```

```

        </div>

    </div>

    <div class="form-row">

        <div class="col-md-6">

            <div class="form-group">

                <label class="small mb-1" for="inputFirstName">First
Name</label>

                <input class="form-control py-4" id="fname"
name="fname" type="text" placeholder="Enter first name" />

            </div>

        </div>

        <div class="col-md-6">

            <div class="form-group">

                <label class="small mb-1" for="inputLastName">Last
Name</label>

                <input class="form-control py-4" id="lname"
name="lname" type="text" placeholder="Enter last name" />

            </div>

        </div>

    </div>

    <div class="form-group">

        <label class="small mb-1"
for="inputEmailAddress">Email</label>

        <input class="form-control py-4" id="email" name="email"
type="email" aria-describedby="emailHelp" placeholder="Enter email address" required />

```

```

</div>

<div class="form-row">

  <div class="col-md-6">

    <div class="form-group">

      <label                class="small                mb-1"
for="inputprofession">Profession</label>

      <select    name="profession"    id="profession"    style="
width:250px;height: 25px">

        <option value="Employee">Employee</option>

        <option value="Business">Business</option>

        <option value="Student">Student</option>

        <option value="Other">Other</option>

      </select>

    </div>

  </div>

</div>

<div class="form-row">

  <div class="col-md-6">

    <div class="form-group">

      <label                class="small                mb-1"
for="inputincome">Income</label>

      <input    class="form-control    py-4"    id="income"
name="income" type="text" placeholder="Income" required/>

    </div>

```

```

        </div>

    </div>

    <div class="form-row">

        <div class="col-md-6">

            <div class="form-group">

                <label                class="small                mb-1"
for="inputsavings">savings</label>

                <input    class="form-control    py-4"    id="Savings"
name="Savings" type="text" placeholder="Amount to be saved" required />

            </div>

        </div>

    </div>

    <div class="form-row">

        <div class="col-md-6">

            <div class="form-group">

                <label                class="small                mb-1"
for="inputPassword">Password</label>

                <input    class="form-control    py-4"    id="pass1"
name="pass1" type="password" placeholder="Enter password" required />

            </div>

        </div>

    </div>

    <div class="col-md-6">

        <div class="form-group">

            <label                class="small                mb-1"
for="inputConfirmPassword">Confirm Password</label>

```



```

                                <input      class="form-control      py-4"      id="pass2"
name="pass2" type="password" placeholder="Confirm password" required />

                                </div>

                                </div>

                                </div>

                                <div class = "btn">

                                <div class="form-group mt-4 mb-0"><button type="submit"
class="btn btn-primary btn-block mt-2">Create User</button></div></div></div>

                                <div class="card-footer text-center">

                                <div class="small"><a href="/">Have an account? Go to
login</a></div>

                                </div>

                                </form>

                                </div>

                                </div>

                                </div>

                                <div>

                                </div>

                                </div>

                                </main>

                                </div>

                                </div>

                                </body>

</html>

{ % endblock body % }

```

LOGIN.HTML

```
{% extends 'base.html' %}

{% block body %}

<div class="bg-image"></div>


<div class="bg-text">

    <body class="bg-primary">

        <div id="layoutAuthentication">

            <div id="layoutAuthentication_content">

                <main>

                    <div class="container">

                        <div class="row justify-content-center">

                            <div class="col-lg-5">

                                <div class="card shadow-lg border-0 rounded-lg mt-5">

                                    <div class="card-header"><h3 class="text-center font-weight-light my-4">Expense Tracker by DataFlair-Login</h3></div>

                                    <div class="card-body">

                                        <form action="/handlelogin/" method="post">

                                            {% csrf_token %}

                                            <div class="form-group">

                                                <label                                class="small                                mb-1"
for="loginusername">Username</label>
```

```

        <input class="form-control py-4" id="loginuname"
name="loginuname" type="text" placeholder="Enter user name" required/>

    </div>

    <div class="form-group">

        <label class="small mb-1"
for="loginpassword1">Password</label>

        <input class="form-control py-4" id="loginpassword1"
name="loginpassword1" type="password" placeholder="Enter password" required/>

    </div>

    <div class="form-group">

        <div class="custom-control custom-checkbox">

            <input class="custom-control-input"
id="rememberPasswordCheck" type="checkbox" required />

            <label class="custom-control-label"
for="rememberPasswordCheck">Remember password</label>

        </div>

    </div>

    <div class="form-group d-flex align-items-center justify-content-
between mt-4 mb-0">

        <div class="btnn">

            <button type="submit" class="btn">Login</button><br>

        </div>

        <div class="link">

            <a class="small" href="\reset_password">Forgot Password?</a><br>

        </div>

    </div>

    <div class="card-footer text-center">

```

```
        <div class="small"><a href="/register">Need an account? Sign  
up!</a></div>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</main>
```

```
</div>
```

```
</body>
```

```
{% endblock body %}
```

MANAGE.PY

```
"""Django's command-line utility for administrative tasks."""
```

```
import os
```

```
import sys
```

```
def main():
```

```
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'ExpenseTracker.settings')
```

```
    try:
```

```
        from django.core.management import execute_from_command_line
```

```
    except ImportError as exc:
```

```

raise ImportError(

    "Couldn't import Django. Are you sure it's installed and "

    "available on your PYTHONPATH environment variable? Did you "

    "forget to activate a virtual environment?"

) from exc

execute_from_command_line(sys.argv)

if __name__ == '__main__':

    main()

```

VIEWS.PY

```

from django.shortcuts import render,HttpResponse,redirect

from django.contrib import messages

from django.contrib.auth import authenticate ,logout

from django.contrib.auth import login as dj_login

from django.contrib.auth.models import User

from .models import Addmoney_info,UserProfile

from django.contrib.sessions.models import Session

from django.core.paginator import Paginator, EmptyPage , PageNotAnInteger

from django.db.models import Sum

from django.http import JsonResponse

import datetime

from django.utils import timezone

def home(request):

```

```

if request.session.has_key('is_logged'):

    return redirect('/index')

return render(request,'home/login.html')

def index(request):

    if request.session.has_key('is_logged'):

        user_id = request.session["user_id"]

        user = User.objects.get(id=user_id)

        addmoney_info = Addmoney_info.objects.filter(user=user).order_by('-Date')

        paginator = Paginator(addmoney_info , 4)

        page_number = request.GET.get('page')

        page_obj = Paginator.get_page(paginator,page_number)

        context = {

            'page_obj' : page_obj

        }

        return render(request,'home/index.html',context)

    return redirect('home')

def register(request):

    return render(request,'home/register.html')

def password(request):

    return render(request,'home/password.html')

def charts(request):

    return render(request,'home/charts.html')

def search(request):

    if request.session.has_key('is_logged'):

```

```

    user_id = request.session["user_id"]

    user = User.objects.get(id=user_id)

    fromdate = request.GET['fromdate']

    todate = request.GET['todate']

    addmoney = Addmoney_info.objects.filter(user=user,
Date__range=[fromdate,todate]).order_by('-Date')

    return render(request,'home/tables.html',{'addmoney':addmoney})

    return redirect('home')

def tables(request):

    if request.session.has_key('is_logged'):

        user_id = request.session["user_id"]

        user = User.objects.get(id=user_id)

        fromdate = request.POST.get('fromdate')

        todate = request.POST.get('todate')

        addmoney = Addmoney_info.objects.filter(user=user).order_by('-Date')

        return render(request,'home/tables.html',{'addmoney':addmoney})

    return redirect('home')

def addmoney(request):

    return render(request,'home/addmoney.html')

def profile(request):

    if request.session.has_key('is_logged'):

        return render(request,'home/profile.html')

    return redirect('/home')

def profile_edit(request,id):

    if request.session.has_key('is_logged'):

```

```

    add = User.objects.get(id=id)

    return render(request,'home/profile_edit.html',{'add':add})

return redirect("/home")

def profile_update(request,id):

    if request.session.has_key('is_logged'):

        if request.method == "POST":

            user = User.objects.get(id=id)

            user.first_name = request.POST["fname"]

            user.last_name = request.POST["lname"]

            user.email = request.POST["email"]

            user.userprofile.Savings = request.POST["Savings"]

            user.userprofile.income = request.POST["income"]

            user.userprofile.profession = request.POST["profession"]

            user.userprofile.save()

            user.save()

            return redirect("/profile")

        return redirect("/home")

def handleSignup(request):

    if request.method == 'POST':

        # get the post parameters

        uname = request.POST["uname"]

        fname=request.POST["fname"]

        lname=request.POST["lname"]

        email = request.POST["email"]

```



```

profession = request.POST['profession']

Savings = request.POST['Savings']

income = request.POST['income']

pass1 = request.POST["pass1"]

pass2 = request.POST["pass2"]

profile = UserProfile(Savings = Savings,profession=profession,income=income)

# check for errors in input

if request.method == 'POST':

    try:

        user_exists = User.objects.get(username=request.POST['uname'])

        messages.error(request, " Username already taken, Try something else!!!")

        return redirect("/register")

    except User.DoesNotExist:

        if len(uname)>15:

            messages.error(request, " Username must be max 15 characters, Please try
again")

            return redirect("/register")

        if not uname.isalnum():

            messages.error(request, " Username should only contain letters and numbers,
Please try again")

            return redirect("/register")

        if pass1 != pass2:

            messages.error(request, " Password do not match, Please try again")

            return redirect("/register")

    user = User.objects.create_user(uname, email, pass1)

```

```

        user.first_name=fname

        user.last_name=lname

        user.email = email

        user.save()

        profile.user = user

        profile.save()

        messages.success(request," Your account has been successfully created")

        return redirect("/")

    else:

        return HttpResponseRedirect('404 - NOT FOUND ')

    return redirect('/login')

def handlelogin(request):

    if request.method == 'POST':

        loginuname = request.POST["loginuname"]

        loginpassword1=request.POST["loginpassword1"]

        user = authenticate(username=loginuname, password=loginpassword1)

        if user is not None:

            dj_login(request, user)

            request.session['is_logged'] = True

            user = request.user.id

            request.session["user_id"] = user

            messages.success(request, " Successfully logged in")

            return redirect('/index')

        else:

```

```

        messages.error(request, " Invalid Credentials, Please try again")

        return redirect("/")

    return HttpResponseRedirect('404-not found')

def handleLogout(request):

    del request.session['is_logged']

    del request.session["user_id"]

    logout(request)

    messages.success(request, " Successfully logged out")

    return redirect('home')

def addmoney_submission(request):

    if request.session.has_key('is_logged'):

        if request.method == "POST":

            user_id = request.session["user_id"]

            user1 = User.objects.get(id=user_id)

            addmoney_info1 = Addmoney_info.objects.filter(user=user1).order_by('-Date')

            add_money = request.POST["add_money"]

            quantity = request.POST["quantity"]

            Date = request.POST["Date"]

            Category = request.POST["Category"]

            add = Addmoney_info(user = user1,add_money=add_money,quantity=quantity,Date =
Date,Category= Category)

            add.save()

            paginator = Paginator(addmoney_info1, 4)

            page_number = request.GET.get('page')

            page_obj = Paginator.get_page(paginator,page_number)

```

```

        context = {

            'page_obj' : page_obj

        }

        return render(request,'home/index.html',context)

    return redirect('/index')

def addmoney_update(request,id):

    if request.session.has_key('is_logged'):

        if request.method == "POST":

            add = Addmoney_info.objects.get(id=id)

            add.add_money = request.POST["add_money"]

            add.quantity = request.POST["quantity"]

            add.Date = request.POST["Date"]

            add.Category = request.POST["Category"]

            add.save()

            return redirect("/index")

        return redirect("/home")

def expense_edit(request,id):

    if request.session.has_key('is_logged'):

        addmoney_info = Addmoney_info.objects.get(id=id)

        user_id = request.session["user_id"]

        user1 = User.objects.get(id=user_id)

        return render(request,'home/expense_edit.html',{ 'addmoney_info':addmoney_info })

    return redirect("/home")

def expense_delete(request,id):

```

```

if request.session.has_key('is_logged'):

    addmoney_info = Addmoney_info.objects.get(id=id)

    addmoney_info.delete()

    return redirect("/index")

return redirect("/home")

def expense_month(request):

    todays_date = datetime.date.today()

    one_month_ago = todays_date-datetime.timedelta(days=30)

    user_id = request.session["user_id"]

    user1 = User.objects.get(id=user_id)

    addmoney = Addmoney_info.objects.filter(user = user1,Date__gte=one_month_ago,Date__lte=todays_date)

    finalrep = {}

    def get_Category(addmoney_info):

        return addmoney_info.Category

    Category_list = list(set(map(get_Category,addmoney)))

    def get_expense_category_amount(Category,add_money):

        quantity = 0

        filtered_by_category = addmoney.filter(Category = Category,add_money="Expense")

        for item in filtered_by_category:

            quantity+=item.quantity

        return quantity

    for x in addmoney:

        for y in Category_list:

            finalrep[y]= get_expense_category_amount(y,"Expense")

```

```

return JsonResponse({'expense_category_data': finalrep}, safe=False)

def stats(request):

    if request.session.has_key('is_logged') :

        todays_date = datetime.date.today()

        one_month_ago = todays_date-datetime.timedelta(days=30)

        user_id = request.session["user_id"]

        user1 = User.objects.get(id=user_id)

        addmoney_info = Addmoney_info.objects.filter(user =
user1,Date__gte=one_month_ago,Date__lte=todays_date)

        sum = 0

        for i in addmoney_info:

            if i.add_money == 'Expense':

                sum=sum+i.quantity

        addmoney_info.sum = sum

        sum1 = 0

        for i in addmoney_info:

            if i.add_money == 'Income':

                sum1 =sum1+i.quantity

        addmoney_info.sum1 = sum1

        x= user1.userprofile.Savings+addmoney_info.sum1 - addmoney_info.sum

        y= user1.userprofile.Savings+addmoney_info.sum1 - addmoney_info.sum

        if x<0:

            messages.warning(request,'Your expenses exceeded your savings')

            x = 0

        if x>0:

```

```

        y = 0

    addmoney_info.x = abs(x)

    addmoney_info.y = abs(y)

    return render(request,'home/stats.html',{'addmoney':addmoney_info})

def expense_week(request):

    todays_date = datetime.date.today()

    one_week_ago = todays_date-datetime.timedelta(days=7)

    user_id = request.session["user_id"]

    user1 = User.objects.get(id=user_id)

    addmoney = Addmoney_info.objects.filter(user = user1,Date__gte=one_week_ago,Date__lte=todays_date)

    finalrep ={}

    def get_Category(addmoney_info):

        return addmoney_info.Category

    Category_list = list(set(map(get_Category,addmoney)))

    def get_expense_category_amount(Category,add_money):

        quantity = 0

        filtered_by_category = addmoney.filter(Category = Category,add_money="Expense")

        for item in filtered_by_category:

            quantity+=item.quantity

        return quantity

    for x in addmoney:

        for y in Category_list:

            finalrep[y]= get_expense_category_amount(y,"Expense")

    return JsonResponse({'expense_category_data': finalrep}, safe=False)

```

```

def weekly(request):

    if request.session.has_key('is_logged') :

        todays_date = datetime.date.today()

        one_week_ago = todays_date-datetime.timedelta(days=7)

        user_id = request.session["user_id"]

        user1 = User.objects.get(id=user_id)

        addmoney_info = Addmoney_info.objects.filter(user =
user1,Date__gte=one_week_ago,Date__lte=todays_date)

        sum = 0

        for i in addmoney_info:

            if i.add_money == 'Expense':

                sum=sum+i.quantity

        addmoney_info.sum = sum

        sum1 = 0

        for i in addmoney_info:

            if i.add_money == 'Income':

                sum1 =sum1+i.quantity

        addmoney_info.sum1 = sum1

        x= user1.userprofile.Savings+addmoney_info.sum1 - addmoney_info.sum

        y= user1.userprofile.Savings+addmoney_info.sum1 - addmoney_info.sum

        if x<0:

            messages.warning(request,'Your expenses exceeded your savings')

            x = 0

        if x>0:

            y = 0

```



```

    addmoney_info.x = abs(x)

    addmoney_info.y = abs(y)

    return render(request, 'home/weekly.html', { 'addmoney_info': addmoney_info })

def check(request):

    if request.method == 'POST':

        user_exists = User.objects.filter(email=request.POST['email'])

        messages.error(request, "Email not registered, TRY AGAIN!!!")

        return redirect("/reset_password")

def info_year(request):

    todays_date = datetime.date.today()

    one_week_ago = todays_date - datetime.timedelta(days=30*12)

    user_id = request.session["user_id"]

    user1 = User.objects.get(id=user_id)

    addmoney = Addmoney_info.objects.filter(user = user1, Date__gte=one_week_ago, Date__lte=todays_date)

    finalrep = { }

    def get_Category(addmoney_info):

        return addmoney_info.Category

    Category_list = list(set(map(get_Category, addmoney)))

    def get_expense_category_amount(Category, add_money):

        quantity = 0

        filtered_by_category = addmoney.filter(Category = Category, add_money="Expense")

        for item in filtered_by_category:

            quantity += item.quantity

        return quantity

```

```
for x in addmoney:

    for y in Category_list:

        finalrep[y]= get_expense_category_amount(y,"Expense")

    return JsonResponse({'expense_category_data': finalrep}, safe=False)

def info(request):

    return render(request,'home/info.html')
```