

PERSONAL EXPENSE TRACKER APPLICATION
IBM-Project-47989-1660803769

NALAIYA THIRAN PROJECT BASED LEARNING ON
PROFESSIONAL READLINESS FOR INNOVATION,
EMPLOYMENT AND ENTERPRENEURSHIP

A PROJECT REPORT
BY

DHINAKARAN V S (2019103517)
PRAVEEN V M (2019103559)
NAVEEN G (2019103040)
SAKTHI KUMAR (2019103574)

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING, GUINDY

INDEX

1. INTRODUCTION

1. Project Overview
2. Purpose

2. LITERATURE SURVEY

1. Existing problem
2. References
3. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

4. REQUIREMENT ANALYSIS

1. Functional requirement
2. Non-Functional requirements

5. PROJECT DESIGN

1. Data Flow Diagrams
2. Solution & Technical Architecture
3. User Stories

6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule

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

1. Feature 1
2. Feature 2
3. Database Schema (if Applicable)

8. TESTING

1. Test Cases

2. User Acceptance Testing

9. RESULTS

1. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub

1. INTRODUCTION

1.1 Project Overview

Category: Cloud App Development

Team ID : PNT2022TMID35263

Skills Required:

IBM Cloud, HTML, Javascript, IBM Cloud Object Storage,
Python- Flask, Kubernetes, Docker, IBM DB2, IBM
Container Registry

Project Description:

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

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

1.2 Purpose

Personal finance management is an important part of people's lives. However, everyone does not have the knowledge or time to manage their finances in a proper manner. And, even if a person has time and knowledge, they do not bother with tracking their expenses as they find it tedious and time-consuming. Now, you don't have to worry about managing your expenses, as you can get access to an expense tracker that will help in the active management of your finances. Also known as expense manager and money manager, an expense tracker is a software or application that helps to keep an accurate record of your money inflow and outflow. Many people in India live on a fixed income, and they find that towards the end of the month they don't have sufficient money to meet their needs. While this problem can arise due to low salary, invariably it is due to poor money management skills.

People tend to overspend without realizing, and this can prove to be disastrous. Using a daily expense manager can help you keep track of how much you spend every day and on what. At the end of the month, you will have a clear picture where your money is

going. This is one of the best ways to get your expenses under control and bring some semblance of order to your finances. Today, there are several expense manager applications in the market. Some are paid managers while others are free. Even banks like ICICI offer their customers expense tracker to help them out. Before you decide to go in for a money manager, it is important to decide the type you want.

2. LITERATURE SURVEY

2.1 Existing problem

[1] Intelligent Online Budget Tracker

The development of this application has been conducted in a stepwise manner using the well-defined methodology, RUP, customized according to the requirements of the system. Most of the goals set at the start of the development phase have been met. Security problems like web security or network security have also been treated in the design and development of the system, thus increasing the reliability of the system. Quality management issues have also been handled satisfactorily.

[2] Online Income and Expense Tracker

This project is work more efficient than the other income and expense tracker. The project successfully avoids the manual calculation for calculating the income and expense per month. The modules are developed efficiently and also in an attractive manner.

[3] Family Expense Manager Application

As the result, the user can make use of this application in his/her daily life. After being used it can be a part of daily life to update and view daily expenses and family expenses. This helps to keep track of expenses & manage it for the user as they are busy in their daily routine, they are not able to keep track of their incomes & expenses.

[4] Personalized Expense Managing Assistant Using Android

Some of the features are like enabling users to register to the application using an existing email or social network account, it will synchronize the user's profile information to the application. Apart from this, the application can be used to gather samples of data related to user's expenses with consents and use those sample data as parameters to assess patterns of spending. Using some data mining techniques expenses can be classified and can be used in market analysis and planning.

[5] Mobiwik Expense Tracking Application

Mobikwik came up with a new feature in their app called Expense Manager. With this feature, you can track and manage your expenditures(expenses), savings, reminders and bill payments. This is a personal budget management app that tracks your expenditures and income and gives you recommendations to make you economically strong.

References

- [1] https://www.researchgate.net/publication/237448489_Intelligent_Online_Budget_Tracker (Bekaroo, Girish & Sunhaloo, Sameer. (2007). Intelligent Online Budget Tracker.)
- [2] <https://www.irjet.net/archives/V6/i3/IRJET-V6I31110.pdf>
- [3] M N Rajaprabha 2017 IOP Conf. Ser.: Mater. Sci. Eng. 263 042050
- [4] <https://easychair.org/publications/preprint/73S7>
- [5] <https://medium.com/@rajotiya.ravi2/case-study-of-expense-tracking-app-get-daily-alerts-of-your-expense-a0561526973d>

2.3 Problem Statement Definition

Customer Problem Statement:

A well-articulated customer problem statement allows us to find the ideal solution for the challenges our customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	User	Limit and manage the expenses	I end up spending more than I intended to	I couldn't keep track of my expenses and was careless in calculations	Extravagant

I am	<small>Describe customer with 3-4 key characteristics - who are they?</small>	Describe the customer and their attributes here
I'm trying to	<small>List their outcome or "job" the care about - what are they trying to achieve?</small>	List the thing they are trying to achieve here
but	<small>Describe what problems or barriers stand in the way - what bothers them most?</small>	Describe the problems or barriers that get in the way here
because	<small>Enter the "root cause" of why the problem or barrier exists - what needs to be solved?</small>	Describe the reason the problems or barriers exist
which makes me feel	<small>Describe the emotions from the customer's point of view - how does it impact them emotionally?</small>	Describe the emotions the result from experiencing the problems or barriers

Personal Expense Tracker Application:

I am	I'm trying to	But	Because	Which makes me feel
a Traveller.	track my expenses during every trip.	unable to sort out my expenses with my records.	there is no root for me to do so.	worried.

PS-2	A person who wants to manage finances	Develop budget plan	I have no time	Keeping track of money spent in paper makes it a tedious process	Unsure about my monthly expenses
------	---------------------------------------	---------------------	----------------	--	----------------------------------

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation & Brainstorming

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

Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization

2

Brainstorm

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

🕒 10 minutes

TIP

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

DHINAKARAN

Secure Access to data

Allow user to add daily expenses

Prepare Graphical Charts on expenses made

Show date and time of transaction

SAKTHI KUMAR

Track expenses

Create reports

Transaction history should be shown

Users are able to plan their budget

NAVEEN

Send reminder to add monthly/ daily expenses

Update wallet balance based on expenses

Detailed report at end of each month

Notify about monthly bill payments

PRAVEEN

Send email alert on exceeding expenses

Make easy UI view of expenses mode

Create expense categories for better experience

Give useful insights about financial management

3

Group Ideas

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

🕒 20 minutes

Secure Access to data

Notify about monthly bill payments

Track expenses

Send email alert on exceeding expenses

Detailed report at end of each month

Create reports

4

Prioritize

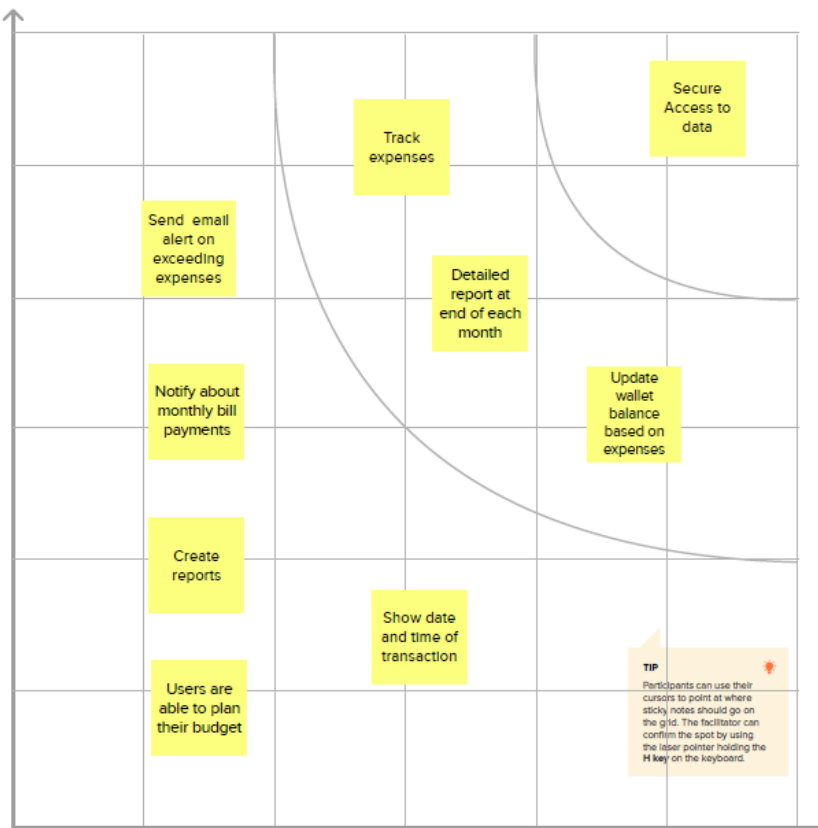
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

🕒 20 minutes

📌

Importance

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



TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H** key on the keyboard.

📌

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The problem of today people's is finding hard to keep track of the Monthly and Yearly expenses that do and to prevent unwanted spending.
2.	Idea / Solution description	Create a web application that will allow users to input their daily expenses and have useful visualizations and reports sent to user.
3.	Novelty / Uniqueness	The personal expense tracker application helps the user not only in budgeting and accounting; it also provides the insights about money management through the analysis. The user also gets notified if the monthly limit is exceeded
4.	Social Impact / Customer Satisfaction	The customer might be pleased to have his spending in check and have an app to remind him when he is over budget and if he misses daily entry.
5.	Business Model (Revenue Model)	The application can have free and premium version where the user can upgrade to premium version to access additional features. Also, the premium version may be advertisement free.
6.	Scalability of the Solution	This project is highly feasible and can later on be further updated with other additional features as <u>well</u> .

3.4 Problem Solution fit

Project Title: Personal Expense Tracker Application

Project Design Phase-I - Solution Fit

Team ID: PNT2022TMID35263

Define CS, fit into CC	1. Customer Segment CS To help the person who is busy and doesn't care about the expenses .They can manage their expenses regularly and keep a track of it and will notify them.	6. Customers Constraints CC Customer need to make sure the internet connection is stable and connected.	5.Available Solutions AS <ul style="list-style-type: none">• A notification system can be enabled in case when the expenses cross over the Income generated by the user to warn user about the situation.• User can also scan the barcode on the price tag which decrease the effort of entering the data in the input fields.	Explore AS, differentiate

<p>3.Triggers</p> <ul style="list-style-type: none"> • The customer is triggered by their Surrounding (friends ,neighbors, etc.) talking about the approach of tracking the expenses 	<p>10.Solution</p> <ul style="list-style-type: none"> • The proposed system makes an attempt to notify them through mail and user will get an analysed report when their expenses exceed the fixed budget via tracking the user expenses daily. • If the user spends large amount of money in a particular area continuously, we will notify them to reduce the spending in that particular area 	<p>8.Channels of behavior</p> <p>1.Offline :</p> <ul style="list-style-type: none"> • Maintain a separate diary , note the expenses at the moment and calculate the daily expenses at the end of the day. <p>2.Online :</p> <ul style="list-style-type: none"> • User will take actions as evaluate their expenses and can track it in online. • User will exhibit this behavior until an authenticated application serve its purpose rightly.
<p>4.Emotions : Before / After</p> <p>Before :</p> <p>Having a fear of spending lot of money and couldn't manage their expenses.</p> <p>After:</p> <p>They can manage their expense regularly.</p>		

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional Requirements:

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 Application or through email.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User monthly expense tentative data	Data to be registered in the app
FR-4	User monthly income data	Data to be registered in the app
FR-5	Alert/ Notification	E-mail alert SMS alert
FR-6	User Budget Plan	Planning and Tracking of user expense vs budget limit based on monthly income of individual.
FR-7	User Savings Budget	Every month has to save some amount and in emergency situations he can access the savings account.

Non-functional Requirements:

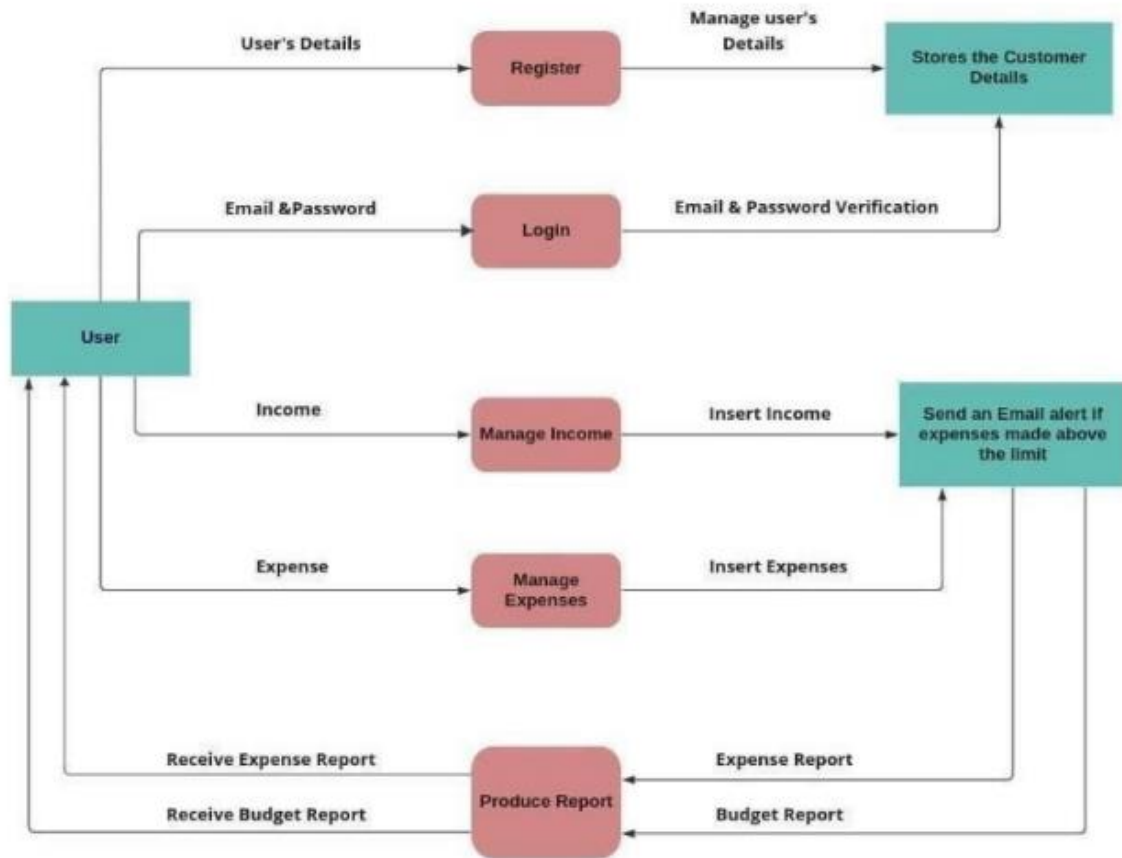
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
--------	----------------------------	-------------

NFR-1	Usability	Effectiveness, efficiency and overall satisfaction of the user while interacting with our application. It should be a GUI interface which is the most easy use of users.
NFR-2	Security	Authentication, authorization, encryption of the application.
NFR-3	Reliability	Probability of failure-free operations in a specified environment for a specified time.
NFR-4	Performance	How the application is functioning and how responsive the application is to the end-users.
NFR-5	Availability	Without near 100% availability, application reliability and the user satisfaction will affect the solution.
NFR-6	Scalability	Capacity of the application to handle growth, especially in handling more users.
NFR-7	Maintenance	The application should maintain the overall history of monthly expenses when the user registered in a safer manner.

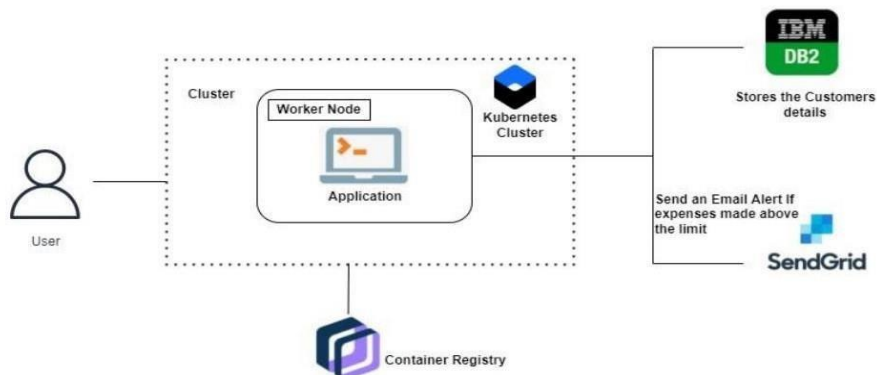
PROJECT DESIGN

4.2 Data Flow Diagrams

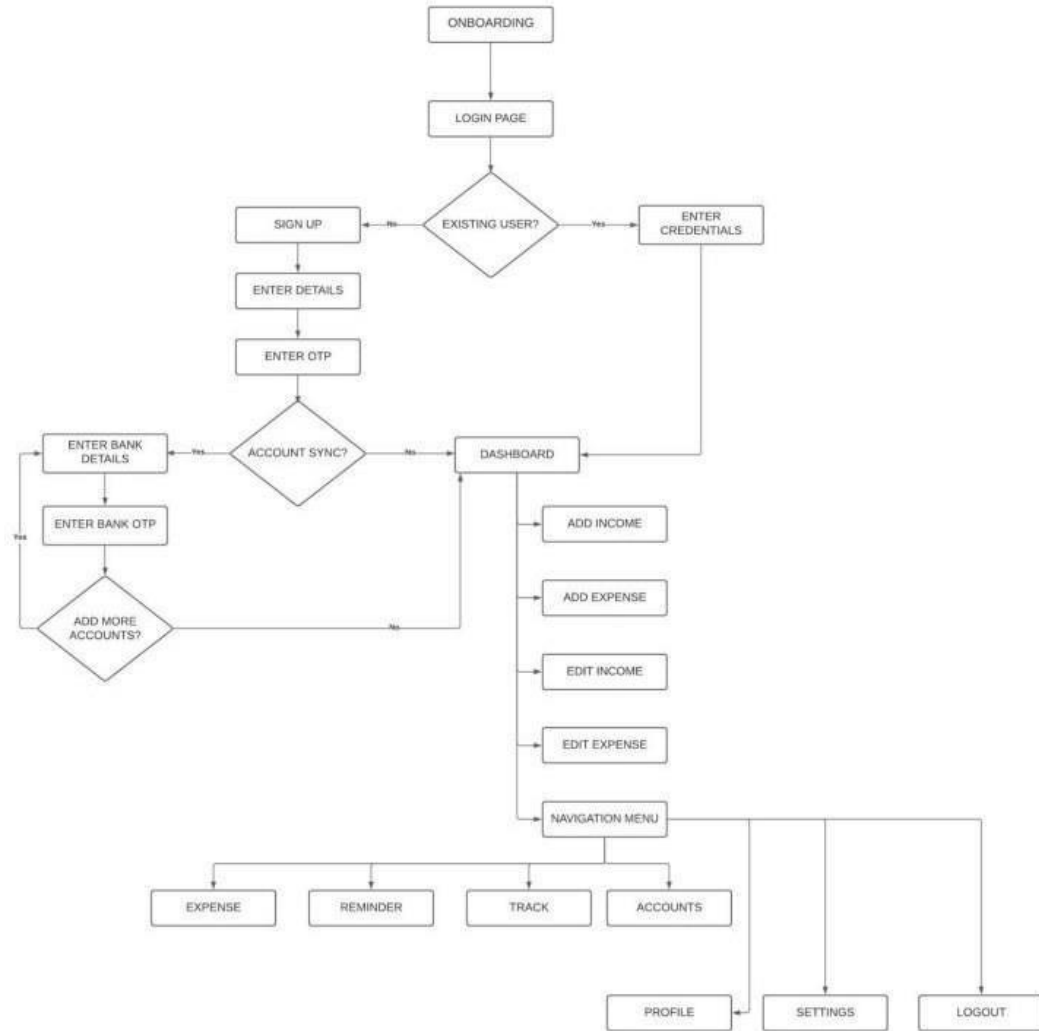


4.3 Solution & Technical Architecture

Technical Architecture:



Solution architecture:



4.4 User Stories

User stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user & Web User)	Registration	USN-1	As a user, I can register to the application by providing my email, password, and confirm my password.	I can access my account / dashboard	High	
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	
	Login	USN-4	As a user, I can log into the application by entering email & password	I can access the application	High	
	Dashboard	USN-5	As a user I can enter my income and expenditure details.	I can view my daily expenses	High	
Customer Care Executive		USN – 6	As a customer care executive, I can solve the log in issues and other issues of the application.	I can provide support or solution at any time 24*7	Medium	
Administrator	Application	USN – 7	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	

5. PROJECT PLANNING & SCHEDULING

5.1 Sprint Planning & Estimation

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	16	6 Days	24 Oct 2022	29 Oct 2022	16	29 Oct 2022
Sprint-2	12	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	14	6 Days	07 Nov 2022	12 Nov 2022	13	12 Nov 2022
Sprint-4	14	6 Days	14 Nov 2022	19 Nov 2022	13	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

5.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	50 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

6. CODING & SOLUTIONING

6.1 Feature 1

We have added the data visualization on methods for expenditure. The pie chart have been used to represent the monthly expenses. The pie chart is a pictorial representation of data that makes it possible to visualize the relationships between the parts and the whole of a variable. For example, it is possible to understand the industry count or percentage of a variable level from the division by areas or sectors. The recommended use for pie charts is two-dimensional, as three-dimensional use can be confusing.

The dimensions form sectors of the measurement values; they can have one or two sizes and up to two measures. The first dimension is used to define the angle of each sector that makes up the chart and the second dimension optionally determines the radius of each sector. Additionally, these plots are useful for comparing data over a fixed period since they do not show

You are looking to categorize and compare a set of data. changes over time.
Therefore, their use should be considered if:

-
- You only have positive values.
- You have less than seven categories since a larger number can make it difficult to perceive each segment.

CODE :

todayExpenses.html & app.py

```
16
17 <link rel="stylesheet" href="..\static\css\home.css">
18
19 <title>Personal Expense Tracker</title>
20
21 </head>
22
23
24
25 <body>
26
27 <!-- Header -->
28
29 <section id="header">
30
31 <div class="header container">
32
33 <div class="nav-bar">
34
35 <div class="brand">
36
37 <a href="#hero">
38
39 <h1><span>B</span>atch: <span>B8</span>-2A4E</h1>
40
41 </a>
42
43 </div>
44
45 <div class="nav-list">
46
47 <div class="hamburger">
48
49 <div class="bar"></div>
50
51 </div>
52
```

18°C
Partly sunny



ENG
IN 07:27
19-11-2022

```

82
83 <!-- Hero Section -->
84
85 <section id="hero">
86
87   <div class="hero container">
88
89     <div>
90
91       <h1>Hey, <span></span></h1>
92
93       <h1>Welcome To <span></span></h1>
94
95       <h1>Personal Expense Tracker Web application <span></span></h1>
96
97       <a href="/signup" type="button" class="cta">Sign-up</a>
98
99     </div>
100
101   </div>
102
103 </section>
104
105 <!-- End Hero Section -->
106
107
108
109 <!-- Service Section -->
110
111 <section id="services">
112
113   <div class="services container">
114
115     <div class="service-top">
116
117       <h1 class="section-title">Serv<span>i</span>ces</h1>
118

```

```

119
120       <h1 class="section-title">Serv<span>i</span>ces</h1>
121
122       <p>Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updated which will be visible to the user. Al
123     </div>
124
125     <div class="service-bottom">
126
127       <div class="service-item">
128
129         <div class="icon"></div>
130
131         <h2>Efficiency</h2>
132
133         <p>personal finance entails all the financial decisions and activities that a Finance app makes your life easier by helping you to manage your finances efficiently.<
134       </div>
135
136       <div class="service-item">
137
138         <div class="icon"></div>
139
140         <h2>Features</h2>
141
142         <p>A FLASK APP THAT USERS MAY USE ON A WEBSITE TO UPDATE THEIR DAILY EXPENSES AND KEEP TRACK OF THEIR SPENDING.</p>
143       </div>
144
145       <div class="service-item">
146
147         <div class="icon"></div>
148
149         <h2>Personal Expenses</h2>
150
151         <p>Budgeting is more than paying bills and setting aside savings.it's about creating a money plan for the life you want</p>
152       </div>
153

```

```
167     </div>
168
169   </div>
170
171 </section>
172
173 <!-- End Service Section -->
174
175
176
177 <!-- About Section -->
178
179 <section id="about">
180
181   <div class="about container">
182
183     <div class="col-left">
184
185       <div class="about-img">
186
187         
188
189       <div><h2></h2></div>
190
191     </div>
192
193   </div>
194
195
196
197   <div class="col-right">
198
199     <h1 class="section-title">About <span>Us</span></h1>
200
201     <h2>Category: Cloud App Development</h2>
202
```

18°C
Partly sunny



07:30
19-11-2022

```
204
205   <h2>Nandhini G</h2>
206
207   <h2>Rindiya S</h2>
208
209   <h2>Reni Jenita C</h2>
210
211   <h2>Rajarajeswari S</h2>
212
213   <p> </p>
214
215   <a href="#footer" class="cta">Follow Us</a>
216
217
218
219 </div>
220
221 </div>
222
223 </section>
224
225 <!-- End About Section -->
226
227
228
229 <!-- Contact Section -->
230
231 <section id="contact">
232
233   <div class="contact container">
234
235     <div>
236
237       <h1 class="section-title">Contact <span>Info</span></h1>
238
239     </div>
240
```

18°C
Partly sunny



07:31
19-11-2022

```
1 <!DOCTYPE html>
2
3 <html>
4
5 <head>
6
7     <title>Login Form</title>
8
9     <link rel="stylesheet" type="text/css" href="..\static\css\login.css">
10
11     <link href="https://fonts.googleapis.com/css?family=Poppins:600&display=swap" rel="stylesheet">
12
13     <script src="https://kit.fontawesome.com/a81368914c.js"></script>
14
15     <meta name="viewport" content="width=device-width, initial-scale=1">
16
17 </head>
18
19 <body >
20
21     
22
23     <div class="container">
24
25
26
```

18°C
Partly sunny

07:31
19-11-2022

```
1 <!DOCTYPE html>
2
3 <html>
4
5 <head>
6
7     <title>Login Form</title>
8
9     <link rel="stylesheet" type="text/css" href="..\static\css\login.css">
10
11     <link href="https://fonts.googleapis.com/css?family=Poppins:600&display=swap" rel="stylesheet">
12
13     <script src="https://kit.fontawesome.com/a81368914c.js"></script>
14
15     <meta name="viewport" content="width=device-width, initial-scale=1">
16
17 </head>
18
19 <body >
20
21     
22
23     <div class="container">
24
25
26
```

07:31
19-11-2022


```

69         <div class="input-div pass">
70
71             <div class="i">
72
73                 <i class="fas fa-lock"></i>
74
75             </div>
76
77             <div class="div">
78
79                 <h5>Password</h5>
80
81                 <input type="password" name="password" class="input" required>
82
83             </div>
84
85         </div>
86
87         <a href="#">Forgot Password?</a>
88
89         <input type="submit" class="btn" value="Login">
90
91         <span>OR</span>
92
93
94         <div><b>Login with</b></div>
95
96         <div>
97
98             <ul>
99
100
101                 <li><a href="#"><i class="fab fa-facebook" aria-hidden="true"></i></a></li>
102
103                 <li><a href="#"><i class="fab fa-twitter" aria-hidden="true"></i></a></li>
104
105                 <li><a href="#"><i class="fab fa-google" aria-hidden="true"></i></a></li>

```

```

94
95         <div><b>Login with</b></div>
96
97         <div>
98
99             <ul>
100
101                 <li><a href="#"><i class="fab fa-facebook" aria-hidden="true"></i></a></li>
102
103                 <li><a href="#"><i class="fab fa-twitter" aria-hidden="true"></i></a></li>
104
105                 <li><a href="#"><i class="fab fa-google" aria-hidden="true"></i></a></li>
106
107                 <li><a href="#"><i class="fab fa-linkedin" aria-hidden="true"></i></a></li>
108
109                 <li><a href="#"><i class="fab fa-instagram" aria-hidden="true"></i></a></li>
110
111             </ul>
112
113         </div>
114
115         <div class="app"><b>Don't have an account?</b><a id="appl" href="\signup">REGISTER here</a></div>
116
117     </form>
118
119 </div>
120
121
122 </div>
123
124
125
126
127 </div>
128
129
130

```

```
1 <html>
2
3 <head>
4
5 <meta charset="utf-8">
6
7 <title>Sign-up</title>
8
9 <link href="..\static\css\signup.css" rel="stylesheet">
10
11 <script src="https://kit.fontawesome.com/a81368914c.js"></script>
12
13 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFaw/d
14
15 </head>
16
17 <body>
18
19 <!--container----->
20
21 <div class="container" >
22
23 <!--sign-up-box-container-->
24
25 <div class="sign-up">
```

18°C
Partly sunny

07:34
19-11-2022

```
31 <!--heading-->
32
33 <form action="/register" method="post">
34
35 <div class="msg">{{ msg }}</div>
36
37 <h1 class="heading">Hello, Friend</h1>
38
39 <!--name-box-->
40
41 <div class="text">
42
43 
44
45 <input placeholder="Name" type="text" name="username"/>
46
47 </div>
48
49 <!--Email-box-->
50
51 <div class="text">
52
53 
54
55 <input placeholder=" Example@gmail.com" type="email" name="email" />
56
57 </div>
58
59 <!--Password-box-->
60
61 <div class="text">
62
63 
64
65 <input placeholder=" Password" type="password" name="password"/>
66
67 </div>
68
```

19°C
Partly sunny

07:34
19-11-2022

```

67 </div>
68
69 <div class="on"><b>OR</b></div>
70
71 <div class="s1"><p><b>Sign-up with</b></p></div>
72
73
74
75 <div>
76
77 <ul>
78
79 <li><a href="#"><i class="fab fa-facebook" aria-hidden="true"></i></a></li>
80
81 <li><a href="#"><i class="fab fa-twitter" aria-hidden="true"></i></a></li>
82
83 <li><a href="#"><i class="fab fa-google" aria-hidden="true"></i></a></li>
84
85 <li><a href="#"><i class="fab fa-linkedin" aria-hidden="true"></i></a></li>
86
87 <li><a href="#"><i class="fab fa-instagram" aria-hidden="true"></i></a></li>
88
89 </ul>
90
91
92
93 </div>
94
95 <!--trem-->
96
97
98
99 <div class="trem">
100
101 <input class="check" type="checkbox" required/>
102
103 <p class="conditions">I read and agree to <a href="#">Trem & Conditions</a></p>

```

```

105 </div>
106
107 <!--button-->
108
109 <div class="toop">
110
111 <button type="submit" class="btn btn-primary">CREATE ACCOUNT</button> </div>
112
113
114
115 </form>
116
117 <!--sign-in-->
118
119 <div class="t"><p class="conditions" id="p3">Already have an account <a href="/signin">Sign in</a></p> </div></div>
120
121 </div>
122
123 <!--text-container-->
124
125 <div class="text-container">
126
127
128
129 <h1 style="color: #2d2c2c;font-family:cursive;">Glad to see you</h1>
130
131
132
133 <div class="diag"></div>
134
135 <div class="para"> <b>Welcome</b>,Please Fill in the blanks for sign up</div>
136
137
138
139 </div>
140
141 </div>

```

```

1 <div class="form-group">
2
3     <label for="">Date</label>
4
5     <input class="form-control" type="datetime-local" name="date" id="date"></div>
6
7 <div class="form-group"> <label for="">Expense name</label>
8
9     <input class="form-control" type="text" name="expensename" id="expensename">
10
11 </div>
12
13 <div class="form-group">
14
15     <label for="">Expense Amount</label>
16
17     <input class="form-control" type="number" name="amount" id="amount">
18
19 </div>
20
21 <!-- <div class="form-group"> <label for="">Pay-Mode</label>
22
23     <input class="form-control" type="text" name="paymode" id="paymode">
24
25 </div>

```

10°C
Partly sunny



ENG IN 07:35 19-11-2022

```

21 <!-- <div class="form-group"> <label for="">Pay-Mode</label>
22
23     <input class="form-control" type="text" name="paymode" id="paymode">
24
25 </div>
26
27 <div class="form-group"> <label for="">Category</label>
28
29     <input class="form-control" type="text" name="category" id="category">
30
31 </div> -->
32
33
34
35 <div class="form-group">
36
37     <label for=""></label>
38
39     <select class="form-control" name="paymode" id="paymode">
40
41         <option selected hidden>Pay-Mode</option>
42
43         <option name="cash" value="cash">cash</option>
44
45         <option name="debitcard" value="debitcard">debitcard</option>
46
47         <option name="creditcard" value="creditcard">creditcard</option>
48
49         <option name="epayment" value="epayment">epayment</option>
50
51         <option name="onlinebanking" value="onlinebanking">onlinebanking</option>
52
53
54
55     </select>

```

18°C
Partly sunny



ENG IN 07:36 19-11-2022

```
107 lines (105 sloc) | 4.34 KB
Raw Blame

1 <div class="container">
2   <h3 class="mt-3">EXPENSES</h3>
3   {% if expense is defined %}
4   {% for row in expense %}
5
6   <div class="row">
7     <div class="col-md-12">
8       <div class="card shadow-sm mb-2 bg-white rounded"></div>
9       <div class="card-body">
10        <div class="row">
11          <div class="col-md-2">
12            <span class="btn btn-outline-dark">{{row[2]}}</span> </div>
13            <div class="col-md-2 mt-3"><h6>{{row[3]}}</h6></div>
14            <div class="col-md-2 mt-3"><span style="color: rgb(255, 0, 0)">{{row[4]}} </div>
15            <div class="col-md-2 mt-3">
16              <span class="badge badge-pill badge-info">{{row[5]}}</span>
17            </div>
18            <div class="col-md-2 mt-3">
19              <span class="badge badge-primary">{{row[6]}}</span>
20            </div>
21            <div class="col-md-1 mt-3">
22              <a href="/edit/{{row[0]}}" class="btn btn-sm btn-success">Edit</a>
23            </div>
24            <div class="col-md-1 mt-3">
25              <a href="/delete/{{row[0]}}" class="btn btn-sm btnDelete btn-success">Delete</a>

```

```
1 <div class="container">
2   <div class="row">
3     <div class="col-md-5">
4       <h3 class="mt-5">MONTH Expense Breakdown</h3></div>
5     </div>
6     <section>
7       <div class="row">
8         <div class="col-md-6">
9           <h3 class="mt-5">Expense Breakdown BY Category</h3>
10          <div class="card shadow mb-2 bg-white rounded-bottom">
11            <div class="card-body">
12              <div class="row">
13                <div class="col-md-6">Food</div>
14                <div id="tfood" class="col-md-6">{{ t_food}} </div>
15              </div>
16            </div>
17          </div>
18          <div class="card shadow mb-2 bg-white rounded">
19            <div class="card-body">
20              <div class="row">
21                <div class="col-md-6">Entertainment</div>
22                <div id="tentertainment" class="col-md-6">{{ t_entertainment}} </div>
23              </div>
24            </div>
25          <div class="card shadow mb-2 bg-white rounded">
26            <div class="card-body">

```

```

25     <div class="card shadow mb-2 bg-white rounded">
26       <div class="card-body">
27         <div class="row">
28           <div class="col-md-6">Business</div>
29           <div id="tbusiness" class="col-md-6">{{t_business}} </div>
30         </div>
31       </div>
32     </div>
33     <div class="card shadow mb-2 bg-white rounded">
34       <div class="card-body">
35         <div class="row">
36           <div class="col-md-6">Rent</div>
37           <div id="trent" class="col-md-6">{{ t_rent }} </div>
38         </div>
39       </div>
40     </div>
41     <div class="card shadow mb-2 btn-outline-danger rounded-pill">
42       <div class="card-body">
43         <div class="row">
44           <div class="col-md-6">Total</div>
45           <div class="col-md-6">₹ {{total}} </div>
46         </div>
47       </div>
48     </div>
49
50   </div>
51   <div class="col-md-6">
52     <canvas id="myChart" width="400" height="400"></canvas>
53     <script>
54       let food = document.getElementById('tfood').innerHTML
55       let entertainment = document.getElementById('tentertainment').innerHTML
56       let business = document.getElementById('tbusiness').innerHTML
57       let rent = document.getElementById('trent').innerHTML
58       let emi = document.getElementById('temi').innerHTML
59       let other = document.getElementById('tother').innerHTML
60       var ctx = document.getElementById('myChart').getContext('2d');
61       var myChart = new Chart(ctx, {

```

```

62     type: 'doughnut',
63     data: {
64       labels: ['Food', 'Entertainment', 'Business', 'Rent', 'EMI', 'Other'],
65       datasets: [{
66         label: 'Expenses Chart',
67         data: [food, entertainment, business, rent, emi, other],
68         backgroundColor: [
69           'rgb(255, 99, 132)',
70           'rgb(0, 0, 0)',
71           'rgb(255, 205, 86)',
72           'rgb(201, 203, 207)',
73           'rgb(54, 162, 235)',
74           'rgb(215, 159, 64)'
75         ],
76       }]
77     },
78   },
79   options: {
80     responsive: true,
81     plugins: {
82       legend: {
83         position: 'bottom',
84       },
85       title: {
86         display: true,
87         text: 'EXPENSE BREAKDOWN'
88       }
89     }
90   }
91   });
92   </script>
93 </div>
94 </div>
95 </div>
96 </section>
97 </div>

```

```
1 <p> Currently your MONTHLY limit is ₹ {{y}} </p>
2 <form action="/limitnum" method="POST">
3   <p> ENTER the MONTHLY LIMIT to avoid over EXPENSES</p> <br/>
4   <input type="number" name="number" required/> <span>
5     <button class="btn btn-warning" type="submit">ENTER</button>
6   </span>
7 </form>
```

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

18°C Partly sunny Windows taskbar icons ENG IN 07:38 19-11-2022

```
43 </div>
44
45 <div class="nav-list">
46
47   <div class="hamburger">
48
49     <div class="bar"></div>
50
51   </div>
52
53   <ul>
54
55     <li><a href="#hero" data-after="Home">Home</a></li>
56
57     <li><a href="#services" data-after="Service">Services</a></li>
58
59
60
61     <li><a href="#about" data-after="About">About</a></li>
62
63     <li><a href="#contact" data-after="Contact">Contact</a></li>
64
65     <li><a href="/signin" data-after="Login">Login</a></li>
66
67   </ul>
68
69 </div>
70
71 </div>
72
73 </div>
74
75 </section>
76
77 <!-- End Header -->
78
79
```

18°C Partly sunny Windows taskbar icons ENG IN 07:29 19-11-2022

26 lines (24 sloc) | 868 Bytes

RawBlame

```
1 import smtplib
2 import sendgrid
3 import os
4 from sendgrid.helpers.mail import Mail, Email, To, Content
5 SUBJECT = "IBM Personal expense tracker"
6 s = smtplib.SMTP('smtp.gmail.com', 587)
7
8 def sendmail(TEXT,email):
9     print("sorry we cant process your candidature")
10    s = smtplib.SMTP('smtp.gmail.com', 587)
11    s.starttls()
12    s.login("nandhini@gmail.com", "Nand@IBM")
13    message = 'Subject: {}'.format(SUBJECT, TEXT)
14    s.sendmail("nandhini@gmail.com", email, message)
15    s.quit()
16
17 def sendgridmail(user,TEXT):
18
19     from_email = Email("nandhini@gmail.com")
20     to_email = To(user)
21     subject = "Limit Message"
22     content = Content("text/plain",TEXT)
23     mail = Mail(from_email, to_email, subject, content)
24     mail_json = mail.get()
25     response = sg.client.mail.send.post(request_body=mail_json)
26     print(response.status_code)
27     print(response.headers)
```

25°C Mostly sunny

ENG IN 08:32 19-11-2022

703 lines (526 sloc) | 20.7 KB

RawBlame

```
1 from flask import Flask, render_template, request, redirect, session
2 from flask_db2 import DB2
3 import ibm_db
4 import ibm_db_dbi
5 import re
6 import os
7 import smtplib, ssl
8 import sendgrid
9 import os
10 from sendgrid.helpers.mail import *
11
12
13
14
15 def sendgridmail(email,text):
16     # try:
17     #     context = ssl.create_default_context()
18     #     server = smtplib.SMTP('smtp.gmail.com', 587)
19     #     # server.connect("smtp.gmail.com",465)
20     #     # server.starttls(context=context)
21     #     server.login("bowmithachandrakaran@gmail.com", "ntofulqqooyroxxi")
22     #     # print("Login Succeeded!")
23     #     # server.sendmail("bowmithachandrakaran@gmail.com", email.strip(), text)
24     #     # print("Email Sent!")
25     # except smtplib.SMTPException as e:
```

25°C Mostly sunny

ENG IN 08:33 19-11-2022


```

56     subject = "Mail from IMS"
57     content = Content("text/plain", "Hello from Sendgrid!")
58     mail = Mail(from_email, to_email, subject, content)
59
60     # Get a JSON-ready representation of the Mail object
61     mail_json = mail.get()
62     # Send an HTTP POST request to /mail/send
63     response = sg.client.mail.send.post(request_body=mail_json)
64     print(response.status_code)
65     print(response.headers)
66
67 app = Flask(__name__)
68
69 app.secret_key = 'a'
70
71 ---
72 dsn_hostname = "3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs21o90108kqb1od81cg.databases.appdomain.cloud"
73 dsn_uid = "sbb93800"
74 dsn_pwd = "wobsVlm6ccFxcNle"
75 dsn_driver = "{IBM DB2 ODBC DRIVER}"
76 dsn_database = "bludb"
77 dsn_port = "31498"
78 dsn_protocol = "tcip"
79
80 dsn = (
81     "DRIVER={0};"
82     "DATABASE={1};"
83     "HOSTNAME={2};"
84     "PORT={3};"
85     "PROTOCOL={4};"
86     "UID={5};"
87     "PWD={6};"
88 ).format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol, dsn_uid, dsn_pwd)
89 ---
90 # app.config['DB2_DRIVER'] = '{IBM DB2 ODBC DRIVER}'
91 app.config['database'] = 'bludb'
92 app.config['hostname'] = 'b1bc1829-6f45-4cd4-bef4-10cf881900bf.clogj3sd0tgtu0lqde00.databases.appdomain.cloud'

```

25°C
Mostly sunny



ENG
IN 08:33
19-11-2022

```

90 # app.config['DB2_DRIVER'] = '{IBM DB2 ODBC DRIVER}'
91 app.config['database'] = 'bludb'
92 app.config['hostname'] = 'b1bc1829-6f45-4cd4-bef4-10cf881900bf.clogj3sd0tgtu0lqde00.databases.appdomain.cloud'
93 app.config['port'] = '32304'
94 app.config['protocol'] = 'tcip'
95 app.config['uid'] = 'dmb99694'
96 app.config['pwd'] = 'FZT6YdaE0F76EFEB'
97 app.config['security'] = 'SSL'
98 try:
99     mysql = DB2(app)
100
101     conn_str='database=bludb;hostname=b1bc1829-6f45-4cd4-bef4-10cf881900bf.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;port=32304;protocol=tcip;\
102             uid=dmb99694;pwd=FZT6YdaE0F76EFEB;security=SSL'
103     ibm_db_conn = ibm_db.connect(conn_str, '', '')
104
105     print("Database connected without any error !!")
106 except:
107     print("IBM DB Connection error : " + DB2.conn_errormsg())
108
109
110 #HOME--PAGE
111 @app.route("/home")
112 def home():
113     return render_template("homepage.html")
114
115
116 @app.route("/")
117 def add():
118     return render_template("home.html")
119
120
121 #SIGN--UP--OR--REGISTER
122
123 @app.route("/signup")
124 def signup():
125     return render_template("signin.html")

```

25°C
Mostly sunny



ENG
IN 08:33
19-11-2022

```

200
201     if request.method == 'POST':
202         username = request.form['username']
203         password = request.form['password']
204
205
206         sql = "SELECT * FROM register WHERE username = ? and password = ?"
207         stmt = ibm_db.prepare(ibm_db_conn, sql)
208         ibm_db.bind_param(stmt, 1, username)
209         ibm_db.bind_param(stmt, 2, password)
210         result = ibm_db.execute(stmt)
211         print(result)
212         account = ibm_db.fetch_row(stmt)
213         print(account)
214
215         param = "SELECT * FROM register WHERE username = " + "\"" + username + "\"" + " and password = " + "\"" + password + "\""
216         res = ibm_db.exec_immediate(ibm_db_conn, param)
217         dictionary = ibm_db.fetch_assoc(res)
218
219
220
221         if account:
222             session['loggedin'] = True
223             session['id'] = dictionary["ID"]
224             print(session['id'])
225             userid = dictionary["ID"]
226             session['username'] = dictionary["USERNAME"]
227             session['email'] = dictionary["EMAIL"]
228
229             return redirect('/home')
230         else:
231             msg = 'Incorrect username / password !'
232
233     return render_template('login.html', msg = msg)
234
235
236

```

25°C
Mostly sunny



ENG
IN 08:34
19-11-2022

```

251 @app.route('/addexpense', methods=['GET', 'POST'])
252 def addexpense():
253
254     date = request.form['date']
255     expensename = request.form['expensename']
256     amount = request.form['amount']
257     paymode = request.form['paymode']
258     category = request.form['category']
259
260     print(date)
261     p1 = date[0:10]
262     p2 = date[11:13]
263     p3 = date[14:]
264     p4 = p1 + "-" + p2 + "." + p3 + ".00"
265     print(p4)
266
267
268     sql = "INSERT INTO expenses (userid, date, expensename, amount, paymode, category) VALUES (?, ?, ?, ?, ?, ?)"
269     stmt = ibm_db.prepare(ibm_db_conn, sql)
270     ibm_db.bind_param(stmt, 1, session['id'])
271     ibm_db.bind_param(stmt, 2, p4)
272     ibm_db.bind_param(stmt, 3, expensename)
273     ibm_db.bind_param(stmt, 4, amount)
274     ibm_db.bind_param(stmt, 5, paymode)
275     ibm_db.bind_param(stmt, 6, category)
276     ibm_db.execute(stmt)
277
278     print("Expenses added")
279
280     # email part
281
282     param = "SELECT * FROM expenses WHERE userid = " + str(session['id']) + " AND MONTH(date) = MONTH(current timestamp) AND YEAR(date) = YEAR(current timestamp) ORDER BY date"
283     res = ibm_db.exec_immediate(ibm_db_conn, param)
284     dictionary = ibm_db.fetch_assoc(res)
285     expense = []
286     while dictionary != False:
287         temp = []

```

25°C
Mostly sunny



ENG
IN 08:34
19-11-2022

6.2 Feature 2

Email notifications will be sent to the users once they cross the expenditure limit through send grid mail system. Most notifications are transactional, meaning a recipient's action or account activity triggers them. But some notifications are marketing related, encouraging the recipient to take a specific action. Ecommerce product notifications inform recipients about new products or discounts. Plus, unlike general marketing emails, these are highly personalized and focus on a single product. For example, if a customer views an item on your website and that item goes on sale, you can send the customer a notification to let them know this is the best time to buy. Users can also opt into receiving notifications when an out-of-stock item is back in stock.

Notification emails tend to perform well because the content is highly relevant to the recipient. But the only way for the recipient to know this is if you state the content clearly in the subject line.

For example, the subject line "New Sign-in to Your Account" gets straight to the point, letting the user know why you sent this notification.

sendemail.py

```
import smtplib import sendgrid as sg
import os from sendgrid import
SendGridAPIClient
from sendgrid.helpers.mail import Mail, Email, To, Content
SUBJECT = "personal expense tracker"
```

```

mail = Mail(from_email, to_email, subject, content)
try:
    sg=SendGridAPIClient('SG.PJq2l00SRKyfdCTfm5b4XQ.I5p737Yx2imRbcRxykMCtRlq1
iZ7s0_cTEDwQbm8nI4') response =
    sg.send(mail)
    print(response.status_code)
    print(response.body)
    print(response.headers)
except Exception as e:
    print(e)

```

6.3 Database Schema

Tables:

1) USER

id INT NOT NULL GENERATED ALWAYS AS IDENTITY,
 username VARCHAR (100) NOT NULL,
 email VARCHAR (100) NOT NULL,
 password VARCHAR (100) NOT NULL

2) EXPENSES

id INT NOT NULL GENERATED ALWAYS AS IDENTITY,
 user_id INT NOT NULL,
 date DATE NOT NULL,
 expense_name VARCHAR (100) NOT NULL,
 amount DECIMAL(10) NOT NULL,
 pay_mode VARCHAR (100) NOT NULL,
 category VARCHAR (100) NOT NULL

3) LIMITS

id INT NOT NULL GENERATED ALWAYS AS IDENTITY,

user_id VARCHAR (100) NOT NULL,

limit DECIMAL(10) NOT NULL

7. TESTING

7.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	BUG ID	Executed By
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1. Go to website 2. Enter Valid username and password	Username: Bowmi password: 123456	Login/Signup popup should display	Working as expected	Pass	-		Bowmitha
LoginPage_TC_002	Functional	Home Page	Verify that the error message is displayed when the user enters the wrong credentials	1. Go to website 2. Enter Invalid username and password	Username: XXXX Password: 12345	Error message should displayed	Working as expected	Pass	-		Vallinayagam
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup	1. Go to website 2. Enter valid credentials 3. Click Login	Username: Bowmi password: 123456	Application should show below UI elements: a. email text box b. password text box c. Login button with orange colour d. New customer? Create account link e. Last password? Recovery password link	Working as expected	Pass	-		Narmatha
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials	1. Go to website 2. Enter details and click login	Username: Bowmi password: 123456	User should navigate to user account homepage	Working as expected	Pass	-		Dhiva Bharathi
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	1. Go to website 2. Enter details and click login	Username: Bowmi password: 123456	Application should show 'incorrect email or password' validation message.	Working as expected	Pass	-		Vallinayagam
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	1. Go to website 2. Enter details and click login	Username: Bowmi password: 123456	Application should show 'incorrect email or password' validation message.	Working as expected	Pass	-		Bowmitha
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with Invalid credentials	1. Go to website 2. Enter details and click login	Username: Bowmi password: 123456	Application should show 'incorrect email or password' validation message.	Working as expected	Pass	-		Narmatha
AddExpensePage_TC_006	Functional	Add Expense page	Verify whether user is able to add expense or not	1. Add date, expense name and other details 2. Check if the expense gets added	add rent = 6000	Application adds expenses	Working as expected	Pass	-		Dhiva Bharathi

User Acceptance Testing

Acceptance Testing UAT Execution & Report Submission

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	8	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	11	20
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	0	1	8
Totals	22	14	11	22	51

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Interface	7	0	0	7
Login	20	0	0	20
Logout	2	0	0	2
Limit	3	0	0	3

8. RESULTS

8.1 Performance Metrics

- Tracking income and expenses: Monitoring the income and tracking all expenditures
(through bank accounts, mobile wallets, and credit & debit cards).
- Transaction Receipts: Capture and organize your payment receipts to keep track of your expenditure.
- Organizing Taxes: Import your documents to the expense tracking app, and it will streamline your income and expenses under the appropriate tax categories.
- Payments & Invoices: Accept and pay from credit cards, debit cards, net banking, mobile wallets, and bank transfers, and track the status of your invoices and bills in the mobile app itself. Also, the tracking app sends reminders for payments and automatically matches the payments with invoices.
- Reports: The expense tracking app generates and sends reports to give a detailed insight about profits, losses, budgets, income, balance sheets, etc.,
- E-commerce integration: Integrate your expense tracking app with your eCommerce store and track your sales through payments received via multiple payment methods.
- Vendors and Contractors: Manage and track all the payments to the vendors and contractors added to the mobile app.
- Access control: Increase your team productivity by providing access control to particular users through custom permissions.

- **Track Projects:** Determine project profitability by tracking labor costs, payroll, expenses, etc., of your ongoing project.
- **Inventory tracking:** An expense tracking app can do it all. Right from tracking products or the cost of goods, sending alert notifications when the product is running out of stock or the product is not selling, to purchase orders.
- **In-depth insights and analytics:** Provides in-built tools to generate reports with easy-to-understand visuals and graphics to gain insights about the performance of your business.
- **Recurrent Expenses:** Rely on your budgeting app to track, streamline, and automate all the recurrent expenses and remind you on a timely basis.

9. ADVANTAGES & DISADVANTAGES

1. **Achieve your business goals** with a tailored mobile app that perfectly fits your business.
2. **Scale-up** at the pace your business is growing.
3. Deliver an **outstanding** customer experience through additional control over the app.
4. Control the **security** of your business and customer data
5. Open **direct marketing channels** with no extra costs with methods such as push notifications.
6. **Boost the productivity** of all the processes within the organization.
7. Increase **efficiency** and **customer satisfaction** with an app aligned to their needs.
8. **Seamlessly integrate** with existing infrastructure.

9. Ability to provide **valuable insights**.

10. Optimize sales processes to generate **more revenue** through enhanced data collection.

11. CONCLUSION

From this project, we are able to manage and keep tracking the daily expenses as well as income. While making this project, we gained a lot of experience of working as a team. We discovered various predicted and unpredicted problems and we enjoyed a lot solving them as a team. We adopted things like video tutorials, text tutorials, internet and learning materials to make our project complete.

12. FUTURE

The project assists well to record the income and expenses in general. However, this project has some limitations:

- The application is unable to maintain the backup of data once it is uninstalled.
- This application does not provide higher decision capability.

To further enhance the capability of this application, we recommend the following features to be incorporated into the system:

- Multiple language interface.
- Provide backup and recovery of data.
- Provide better user interface for user.
- Mobile apps advantage.

13. APPENDIX

Source Code Github Link :

<https://github.com/IBM-EPBL/IBM-Project-25647-1659969628>