

PERSONAL EXPENSE TRACKER

TEAM DETAILS:

Team No : PNT2022TMID43064

College Name : Sree Sakthi Engineering College

Department : Computer Science & Engineering

Team Leader - Mahalakshmi.S

Team Member 1 - Manikandan.M

Team Member 2 - Mareeswari.K

Team Member 3 - Vignesh.G

TABLE CONTENT

1. INTR ODUCTION

1.1 Project Overview

1.2 Purpose

2.LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3Proposed Solution

3.4Problem Solution fit

4.REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5.PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

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

7.1 Feature 1

7.2 Feature 2

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE WORK

13. APPENDIX

Source Code

Github & project demo link

1. INTRODUCTION:

1.1 PROJECT OVERVIEW

The web application "Personal Expense Tracker" is developed to manage the daily expenses in a more efficient and manageable way. By using this application we can reduce the manual calculations of the daily expenses and keep track of the expenditure. In this application, user can provide his income to calculate his total expenses per day and these results will be stored for each user.

The application has the provision to predict the income and expense for the manager using data mining. Users are add type of expense, verify expense add type of income, verify income and generate reports. The application's interface is designed using custom art elements.

1.2 Purpose

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.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM :

Accounting, for any business, can be a complex undertaking. A manual accounting system requires you to understand the accounting process in a way that may be unnecessary with a computerized accounting system. This can be an advantage or a disadvantage, depending on the person doing the bookkeeping; often, a specially trained professional is needed to ensure that accounting is done properly. Unraveling the complexity of your financial records by hand may be time consuming. Since it takes time to generate reports.

2.2 Reference:

- i. Accreditation and Quality Assurance Committee (AQAC) in Palestine. General Report of Information Technology and Engineering Higher Education in Palestine. Accreditation and Quality Assurance Commission (AQAC). Ramallah, Palestine: Palestinian Ministry of Education and Higher Education; 2007 Apr.
- ii. Chen JW, Yen M. Engineering Accreditation: A Foundation for Continuing Quality Improvement. 2005 Mar 1–5; Tainan. Exploring Innovation in Education and Research..
- iii. Engineering Association of Palestine. Current Engineering Statistics Book. Ramallah; 2005.
- iv. Homma H. Accreditation System in Indonesia. JSME news. 2004 May;
- v. Oberst B, Jones R. International Trends in Engineering Accreditation and Quality Assurance. World Expertise L.L.C.
- vi. Palestinian Ministry of Education and Higher Education. Palestinian Higher Education Statistics.
- vii. Prados J, Peterson G, Lattuca L. Quality Assurance of Engineering Education Through Accreditation: The Impact of Engineering Criteria 2000 and Its Global Influence. Journal of Engineering Education. 2005 Jan; 94(1):165–84.
- viii. THE OMNI GROUP, OmniGraffle for Mac. The Omni Group, <http://www.omnigroup.com/products/omnigraffle/>, accessed November 2012, n.d.
- ix. T. WOOLEY, A comparative study of the Android and iPhone operating systems. University of Central Florida, <http://www.cs.ucf.edu/~dcm/Teaching/COP5611Spring2010/Project/TravisWooley-Presentation.pdf>, accessed October 2012, April 2010.

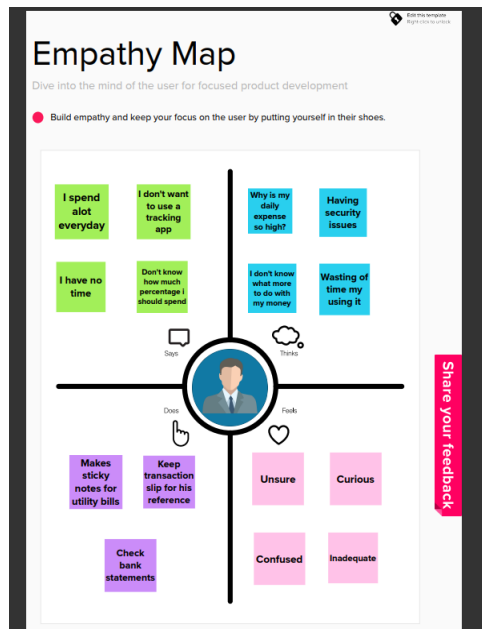
- X. L.VANDALANDO. GENTZ, Homepage. InApp SettingsKit, [http:// www. inapp settingskit .com/home](http://www.inappsettingskit.com/home), accessed October 2012, n.d. [18]
- XI. MOBILE INNOVATION, Smartphones operating systems war. The Mobile Innovation, <http://www.themobileinnovation.net/smartphones-operating-systems-war-android-vsblackberry-vs-ios-vs-symbian>, accessed October 2012, n.d
- XII. P. ALESSI, Professional iPhone and iPad Database Application Programming, Wiley Publishing, Indianapolis, Indiana, 2011.
- XII G. AMBROZIO, Block-based action sheet. Cocoanetics, <http://www.cocoanetics.com/2012/06/block-based-action-sheet/>, accessed October 2012, n.d.

2.3 Problem Statement Definition :

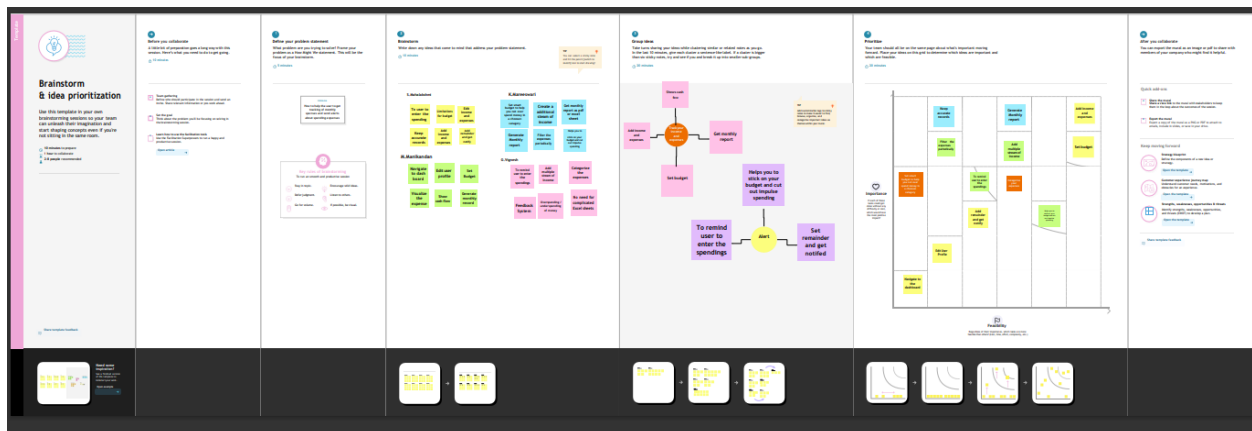
Many organizations have their own system to record their income and expenses, which they feel is the main key point of their business progress. It is good habit for a person to record daily expenses and earning but due to unawareness and lack of proper applications to suit their privacy, lacking decision making capacity people are using traditional note keeping methods to do so. Due to lack of a complete tracking system, there is a 2 constant overload to rely on the daily entry of the expenditure and total estimation till the end of the month.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy map



3.2 Ideation & Brainstorming



3.3 Proposed Solution

Daily expense management system which is specially designed for non-salaried and salaried personnel for keeping track of their daily expenditure with easy and effective way through computerized system which tends to eliminate manual paper works. 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. 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.

3.4 Problem Solution Fit

1. CUSTOMER SEGMENT(S) People who are struggling to track their expenses are our customers. They can use our app to maintain records about their income and expenses.	6. CUSTOMER LIMITATIONS User have to entry every record manually. The category divided may be blunder or messy. person who is handling system must have some technical knowledge.	5. AVAILABLE SOLUTIONS User can add their income and expenses. 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.
2. PROBLEMS / PAINS In paper-based expense tracker system it is difficult to track our monthly expenses manually. The paper-based expense records may get lost in case of fire accidents, flood etc.	9. PROBLEM ROOT / CAUSE When the digits could not be recognized correctly. When the transactions are not successful. When the elder people unable to understand the smaller handwritten digits. When the paper based expense tracker records are subjected to fire accident, flood, etc.	7. BEHAVIOR They may keep a temporary note on their mobile. He/She will tell the other persons to remember the expense they do while calculating the expenses they consider only on the expenses that are single time and huge and leave the rest
3. TRIGGERS TO ACT This application can create awareness among common people about their income and expenses. It Reduces time rather than entering details manually.	10. YOUR SOLUTION The application should be able to generate reports of their spending and notify users if they have exceeded their budget. This application can create awareness among common people about finance and stuffs. This application also helps user to be financially responsible.	8. CHANNELS of BEHAVIOR ONLINE Download statements from bank and pay monthly installment OFFLINE Using spreadsheets and notes for financial management
4. EMOTIONS Frustration, Confusion, Inadequate > Boost, Feeling smart, Be an example for others		

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

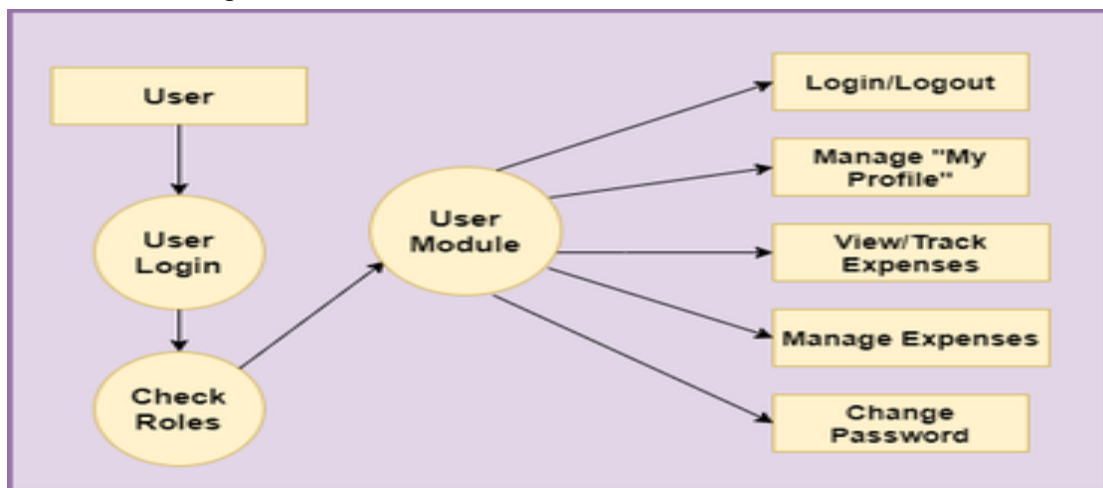
IBM Cloud, HTML, Javascript, IBM Cloud Object Storage, Python Flask, Kubernetes, Docker, IBM DB2, IBM Container Registry

4.2 Non Functional Requirement

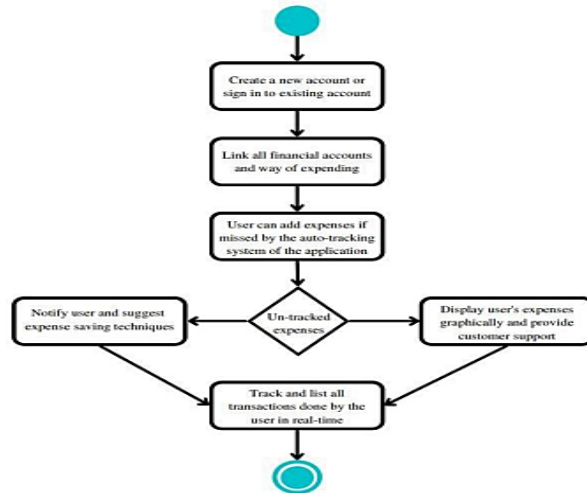
8GB RAM, Intel Core i3, OS-Windows/Linux/MAC, Laptop or Desktop

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.Users Story

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user & web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	
		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 a 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

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Mahalakshmi
		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Manikandan
	Login	USN-3	As a user, I can login to the application by entering email & password	1	High	Mareeswari
	Dashboard	USN-4	Logging in takes to the dashboard for the logged user.	2	High	Vignesh
Bug fixes, routine checks and improvisation by everyone in the team*Intended bugs only						

Sprint 2	Workspace	USN-1	Workspace forpersonal expense tracking	2	High	mahalakshmi
	Charts	USN-2	Creating variousgraphs and statistics of customer’s data	1	Medium	Mareeswari
	Connecting to IBM DB2	USN-3	Linking database with dashboard	2	High	vignesh
		USN-4	Making dashboard interactive with JS	2	High	Manikandan
Sprint-3		USN-1	Wrapping up the serverside works of frontend	1	Medium	Mareeswari
	Watson Assistant	USN-2	Creating Chatbot for expense tracking and for clarifying user’s query	1	Medium	vignesh
	SendGrid	USN-3	Using SendGrid to send mail to the userabout their expenses	1	Low	Manikandan
		USN-4	Integrating both frontend and backend	2	High	mahalakshmi
Bug fixes,routine checks andimprovisation by everyone in the team*Intended bugs only						

Sprint-4	Docker	USN-1	Creating image of website using docker/	2	High	mahalakshmi
	Cloud Registry	USN-2	Uploading docker image to IBM Cloud registry	2	High	vignesh
	Kubernetes	USN-3	Create container using the docker image and hosting the site	2	High	Manikandan
	Exposing	USN-4	Exposing IP/Ports for the site	2	High	Mareeswari

6.2 Sprint Delivery Schedule

SPRINT DELIVERY PLAN

Project Tracker, Velocity & BurndownChart: (4marks)

Sprint	Total Story Points	Duration	Sprint StartDate	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint ReleaseDate (Actual)
Sprint-1	20	6 Days	23 Oct2022	28 Oct 2022	20	29 Oct2022
Sprint-2	20	6 Days	30 Oct2022	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

Velocity

We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint).

Calculating the team's average velocity (AV) per iteration unit (story point per day)

$$AV = \text{sprint duration} / \text{velocity} = 20/6 = 3$$

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

7.1 FEATURE 1

HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript). "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , , <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, , , and many others. An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. However, the convention and recommended practice is to write tags in lowercase.

CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; rather, everything is now CSS without a version number.

7.2 FEATURE 2

PYTHON (FLASK)

Flask is a web application framework written in Python. It was developed by Armin Ronacher in 2004, who led a team of international Python enthusiasts called Pooeco. According to Ronacher, the idea was originally an April Fool's joke that was

popular enough to make into a serious application. The name is a play on the earlier Bottle framework. When Ronacher and Georg Brandl created a bulletin board system written in Python in 2004, the Pocco projects Werkzeug and Jinja were developed. In April 2016, the Pocco team was disbanded and development of Flask and related libraries passed to the newly formed Pallets project. Since 2018, Flask-related data and objects can be rendered with Bootstrap. Flask has become popular among Python enthusiasts. As of October 2020, it has second most stars on GitHub among Python web-development frameworks, only slightly behind Django, and was voted the most popular web framework in the Python Developers Survey 2018, 2019, 2020 and 2021.

Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine. Both are Pocco projects. To install flask on the system, we need to have python 2.7 or higher installed on our system. However, we suggest using python 3 for the development in the flask.

WSGI : It is an acronym for web server gateway interface which is a standard for python web application development. It is considered as the specification for the universal interface between the web server and web application.

Jinja2 : Jinja2 is a web template engine which combines a template with a certain data source to render the dynamic web pages.

SQL

SQL (Structured Query Language) is used to perform operations on the records stored in the database, such as updating records, inserting records, deleting records, creating and modifying database tables, views, etc. SQL is not a database system, but it is a query language. Suppose you want to perform the queries of SQL language on the stored data in the database. You are required to install any database management system in your systems, for example, Oracle, MySQL, MongoDB, PostgreSQL, SQL Server, DB2, etc. SQL is a short-form of the structured query language, and it is pronounced as S-Q-L or sometimes as See-Quell.

This database language is mainly designed for maintaining the data in relational database management systems. It is a special tool used by data professionals for handling structured data (data which is stored in the form of tables). It is also designed for stream processing in RDBMS. You can easily create and manipulate the database, access and modify the table rows and columns, etc. This query language became the standard of ANSI in the year of 1986 and ISO in the year of 1987. If you want to get a job in the field of data science, then it is the most important query language to learn. Big enterprises like Facebook, Instagram, and LinkedIn, use SQL for storing the data in the back-end.

8.TESTING

8.1 Test Cases

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

There are various types of tests. Each test type addresses a specific testing requirement. Following this step, a variety of tests are conducted.

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application; it is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basicTests at component level and test a specific business process, application, and/or System configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successively unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problem that arises from the combination of components.

Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- | | |
|---------------|--|
| Valid Input | : identified classes of valid input must be accepted. |
| Invalid Input | : identified classes of invalid input must be rejected. |
| Function | : identified functions must be exercised. |
| Output | : identified classes of application outputs must be exercised. |

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It has a purpose. It is used to test areas that cannot be reached from a blackbox level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. Such as specification or requirements document. It is a test in which the software under test is treated as a black box you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit test is usually conducted as part of a combined code and unit test and unit testing phase of the software lifecycle, although it is not uncommon for coding and unit tests to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format.

No duplicate entries should be allowed.

All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or one step up- software applications at the company level - interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.2. USER ACCEPTANCE TESTING:

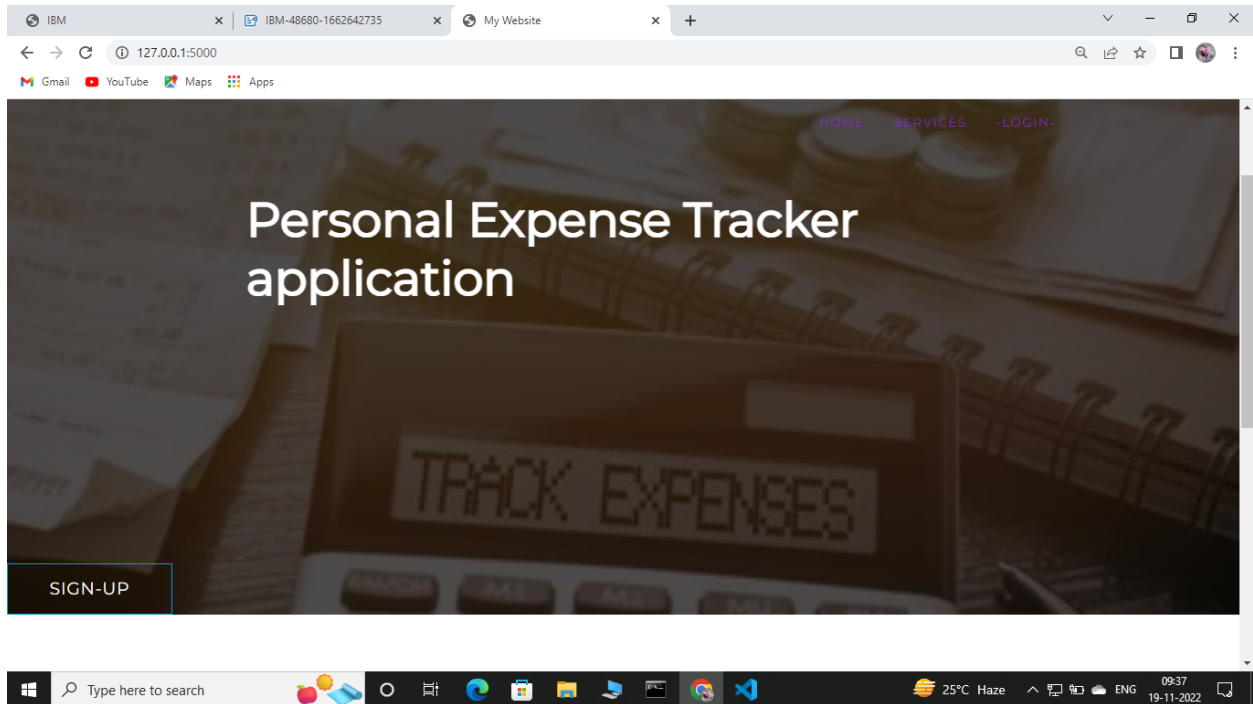
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

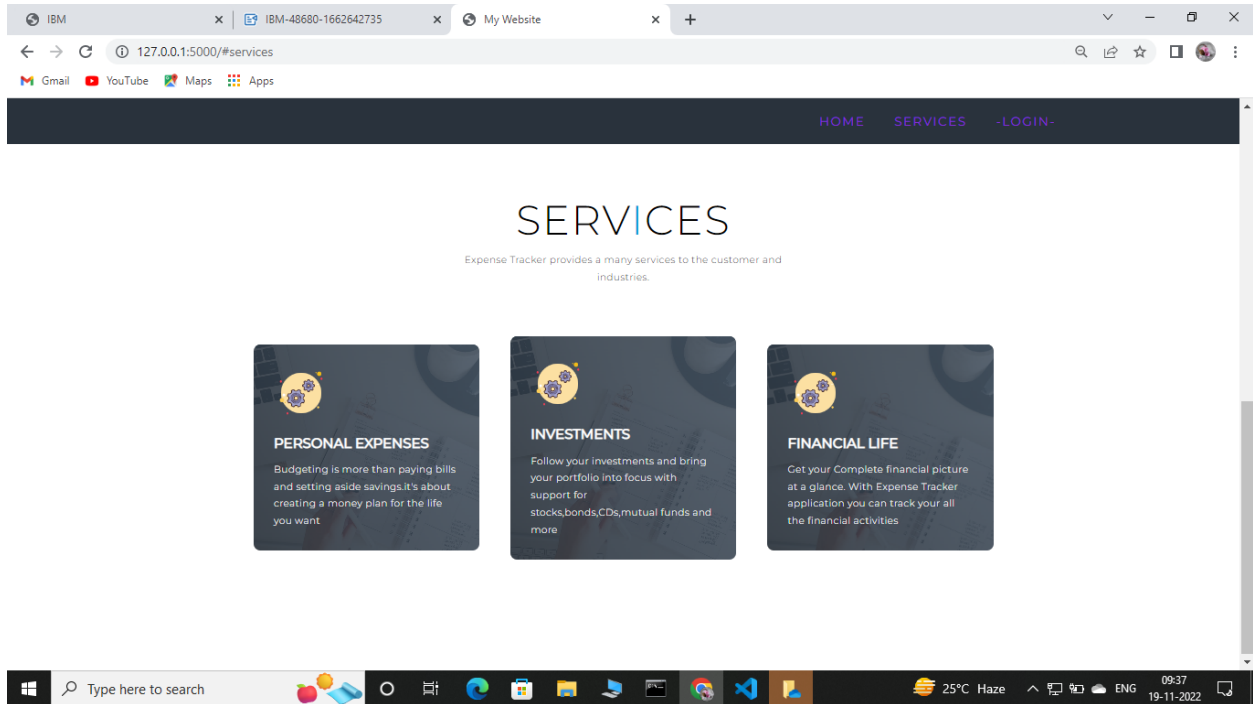
9.RESULTS

9.1 Performance Matrics

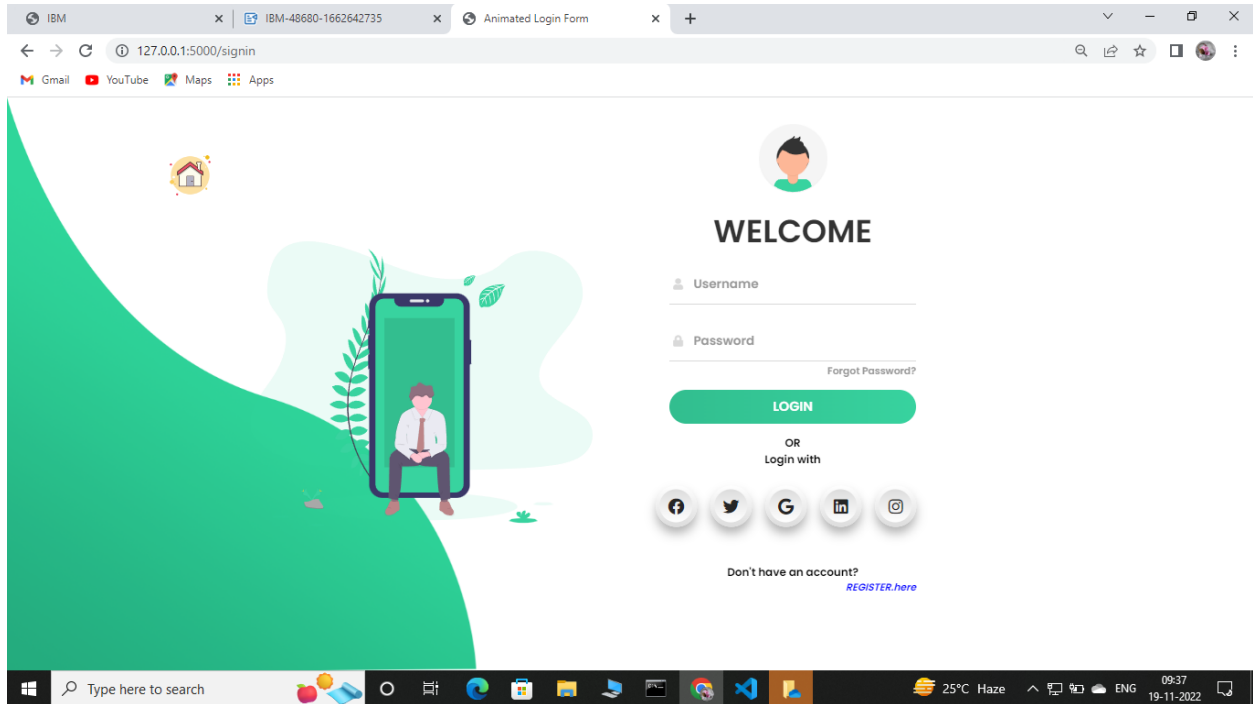
I)HOME PAGE



II)SERVICE



III) LOGIN PAGE



IV)ADD EXPENSE

The screenshot displays a web browser window with the URL `127.0.0.1:5000/add`. The browser's address bar shows the address and a search icon. The page title is "MyBudget" and the navigation bar includes links for "Home", "Add", "History", "LIMIT", and "Report". A user profile icon labeled "User" is in the top right corner.

The main content area is titled "Add Expense" and contains the following form fields:

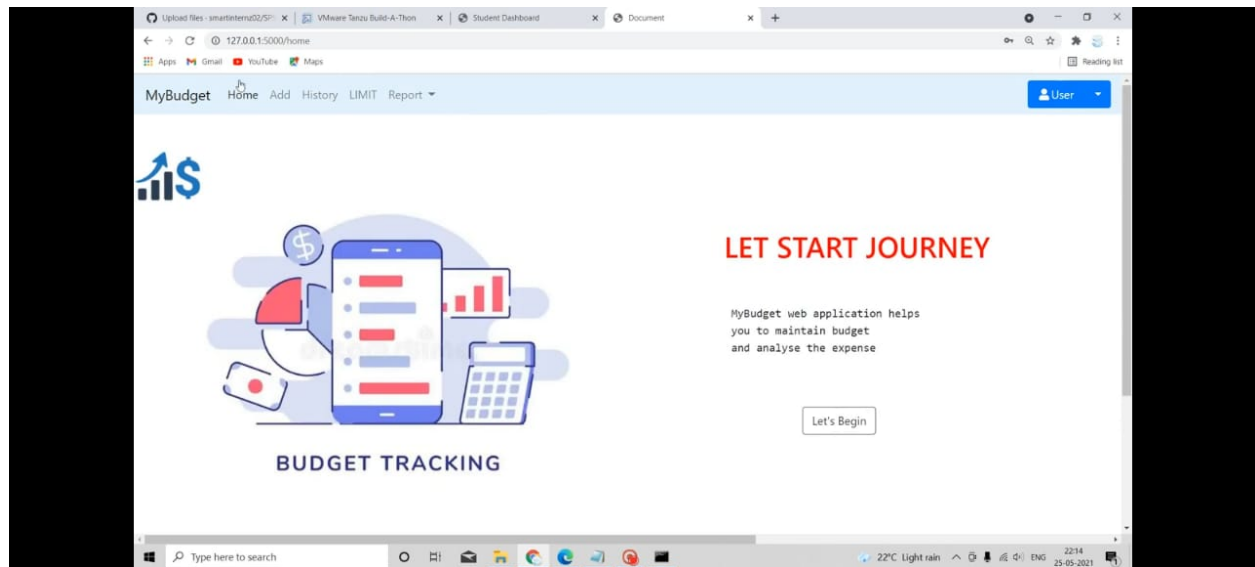
- Date:** A text input field with a date picker icon, showing the placeholder "dd-mm-yyyy".
- Expense name:** A text input field.
- Expense Amount:** A text input field.
- Pay-Mode:** A dropdown menu.
- Category:** A dropdown menu.
- Add:** A red button to submit the form.

On the right side of the form, there is a decorative illustration of a notepad with the title "EXPENSES". The notepad has a checklist of expenses with checkboxes:

- FOOD ☒
- ELECTRIC ☐
- WATER ☒
- PHONE ☒
- INTERNET ☐

The Windows taskbar at the bottom shows the search bar, task view, and several application icons. The system tray on the right indicates the temperature is 22°C, there is light rain, and the date is 25-05-2021.

V) BUDGET PAGE



10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Financially Aware and Improve Money Management tracking your expenditures ensures you achieve your project financial targets.
- Helps anticipate the costs of similar projects When you formally track and report expenses, you have a permanent documentation which helps you correctly anticipate expenses for similar projects in the future. This is even more significant when it comes to budget-making process.

DISADVANTAGES:

- Searching functionality is missing in the current version of the application.
- The user can only enter the expense/income amount in USD Currency. So, this application would primarily be used in USA only and cannot be used all over the world because other currencies are not being addressed in this version. The expenses/income can only be tracked in US currency thereby.

11.CONCLUSION

The project what we have developed is work more efficient than the other income and expense tracker. The project successfully avoids the manual calculation for avoiding calculating the income and expense per month. The modules are developed with efficient and also in an attractive manner. The developed systems dispense the problem and meet the needs of by providing reliable and comprehensive information.Despite of the mentioned flaws, the application would work as a charm for tracking expenses and provides the user, the flexibility to see the reports

12.FUTURE SCOPE

All the limitations discussed should be addressed in the next version of the application so that it can be more enhanced and user friendly.Provision to add different currencies will be added so that this application is not just limited to USA but also can be used worldwide and the currency converters will be designed and added in order to convert the different currency rates.

A new tab named “Search” will be implemented so that if the user searches for any vendor, category or subcategory by name, he can see the expenses made on that particular search in a table view list with the total number of transactions made and the total expense amount for that search. This would provide a lot more flexibility for the users to track the particular expenses on particular items.

13.APPENDIX

SOURCE CODE

I)HOME PAGE:

```
"13%" width="13%" style="position: relative; top: -690PX; left: -60PX;"  
src="../../../static/images/istockphoto-943300706-612x612.jpg">
```

```
</div>
```

```
<span class="btn btn-outline-dark">Let's Begin</span>
```

```
</div>
```

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

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

```
<style>
```

```
  H1 {
```

```
    position: relative;
```

```
    right: -790PX;
```

```
    top: -400PX;
```

```
    color: RED;
```

```
  }
```

```
p{
```

```
position: relative;
```

```
right: -800px;
```

```
top: -350px;
```

```
font-family:monospace;
```

```
}
```

```
span{
```

```

    position: relative;
right: -800px;
top: -360px;}
.ccc {
    position: relative;
    top:80px;
    left:-100px;

}
</style>
<div id=aa class="container">
<div class="ccc">
    
    <h1>LET START JOURNEY</h1>
    <P>MyBudget web application helps<br> you to maintain budget<br>
        and analyse the expense</P>
    <img height=

```

LOGIN PAGE

```

<!DOCTYPE html>
<html>
<head>
    <title>Animated Login Form</title>
    <link rel="stylesheet" type="text/css" href="../../static/css/login.css">
    <link href="https://fonts.googleapis.com/css?family=Poppins:600&display=swap"
rel="stylesheet">

```

```
<script src="https://kit.fontawesome.com/a81368914c.js"></script>

<meta name="viewport" content="width=device-width, initial-scale=1">

</head>

<body >

    <div class="container">

        <div class="img">

            <div id="png"><a href="/" title="HOME"></a></div>

        </div>

        <div class="login-content">

            <form action="/login" method="POST">

                <div class="msg">{{ msg }}</div>

                <h2 class="title">Welcome</h2>

                <div class="input-div one">

                    <div class="i">

                        <i class="fas fa-user"></i>

                    </div>

                    <div class="div">

                        <h5>Username</h5>

                        <input type="text" name="username" class="input" required>

                    </div>

                </div>

            </form>

        </div>

    </div>

</body>

</html>
```

</div>

<div class="input-div pass">

<div class="i">

<i class="fas fa-lock"></i>

</div>

<div class="div">

<h5>Password</h5>

<input type="password" name="password" class="input" required>

</div>

</div>

Forgot Password?

<input type="submit" class="btn" value="Login">

OR

<div>Login with</div>

<div>

<i class="fab fa-facebook" aria-hidden="true"></i>

<i class="fab fa-twitter" aria-hidden="true"></i>

<i class="fab fa-google" aria-hidden="true"></i>

<i class="fab fa-linkedin" aria-hidden="true"></i>

<i class="fab fa-instagram" aria-hidden="true"></i>


```

        </div>

        <div class="app" ><b>Don't have an account?</b><a id="app1"
href="\signup">REGISTER.here</a></div>

    </form>

</div>

</div>

<script type="text/javascript" src="..\static\js\login.js"></script>
</body>
</html>

```

EDIT :

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

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

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

```
    <div class="row">
```

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

```
            <h3>Edit Expense</h3>
```

```
            <form action="/update/{{expenses[0]}}" method="POST">
```

```
                <input type="hidden" class="form-control" name="" value = "{{expenses[0]}}"
```

id="">

<div class="form-group">

<label for="">Date</label>

<input class="form-control" type="datetime-local" name="date"
value="{{expenses[2]}}" id="date"></div>

<script type="text/javascript">

var d = new Date(value="{{expenses[2]}}");

var elem = document.getElementById("date");

elem.value = d.toISOString().slice(0,16);

</script>

<div class="form-group"> <label for="">Expense name</label>

<input class="form-control" type="text" name="expensename"
value="{{expenses[3]}}" id="expensename">

</div>

<div class="form-group">

<label for="">Expense Amount</label>

<input class="form-control" type="number" min="0" name="amount"
value="{{expenses[4]}}" id="amount">

</div>

<div class="form-group">

<label for=""></label>

```
        <select class="form-control" name="paymode" value="{{expenses[5]}}"
id="paymode">

        <option selected hidden>{{expenses[5]}}</option>

        <option value="cash">cash</option>

        <option value="debitcard">debitcard</option>

        <option value="creditcard">creditcard</option>

        <option value="epayment">epayment</option>

        <option value="onlinebanking">onlinebanking</option>

        </select>

    <div class="form-group">

        <label for=""></label>

        <select class="form-control" name="category" value="{{expenses[6]}}"
id="category">

        <option selected hidden>{{expenses[6]}}</option>

        <option value="food">food</option>

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

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

        <option value="rent">Rent</option>

        <option value="EMI">EMI</option>

        <option value="other">other</option>

        </select>

    </div>
```

```
<input class="btn btn-danger" type="submit" value="Update" id="">
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
{% endblock %}
```

PYTHON (Flask)

```
from flask import Flask, render_template, request, redirect, session
```

```
from flask_mysqlldb import MySQL
```

```
import MySQLdb.cursors
```

```
import re
```

```
app = Flask(__name__)
```

```
app.secret_key = 'a'
```

```
app.config['MYSQL_HOST'] = 'remotemysql.com'
```



```
app.config['MYSQL_USER'] = 'D2DxDUPBii'
app.config['MYSQL_PASSWORD'] = 'r8XBO4GsMz'
app.config['MYSQL_DB'] = 'D2DxDUPBii'
```

```
mysql = MySQL(app)
```

#HOME--PAGE

```
@app.route("/home")
def home():
    return render_template("homepage.html")
```

```
@app.route("/")
def add():
    return render_template("home.html")
```

#SIGN--UP--OR--REGISTER

```
@app.route("/signup")
def signup():
    return render_template("signup.html")
```

```

@app.route('/register', methods=['GET', 'POST'])

def register():

    msg = ''

    if request.method == 'POST' :

        username = request.form['username']

        email = request.form['email']

        password = request.form['password']


        cursor = mysql.connection.cursor()

        cursor.execute('SELECT * FROM register WHERE username = % s', (username, ))

        account = cursor.fetchone()

        print(account)

        if account:

            msg = 'Account already exists !'

        elif not re.match(r'^[@]+@[^@]+\.[^@]+', email):

            msg = 'Invalid email address !'

        elif not re.match(r'[A-Za-z0-9]+', username):

            msg = 'name must contain only characters and numbers !'

        else:

            cursor.execute('INSERT INTO register VALUES (NULL, % s, % s, % s)', (username,
email,password))

            mysql.connection.commit()

            msg = 'You have successfully registered !'

            return render_template('signup.html', msg = msg)

```

```
#LOGIN--PAGE
```

```
@app.route("/signin")
```

```
def signin():
```

```
    return render_template("login.html")
```

```
@app.route('/login',methods=['GET', 'POST'])
```

```
def login():
```

```
    global userid
```

```
    msg = "
```

```
if request.method == 'POST' :
```

```
    username = request.form['username']
```

```
    password = request.form['password']
```

```
    cursor = mysql.connection.cursor()
```

```
        cursor.execute('SELECT * FROM register WHERE username = % s AND password = % s',  
(username, password ),)
```

```
    account = cursor.fetchone()
```

```
    print (account)
```

```
if account:
```

```
session['loggedin'] = True  
  
session['id'] = account[0]  
  
userid= account[0]  
  
session['username'] = account[1]
```

```
return redirect('/home')
```

```
else:
```

```
    msg = 'Incorrect username / password !'
```

```
return render_template('login.html', msg = msg)
```

#ADDING----DATA

```
@app.route("/add")
```

```
def adding():
```

```
    return render_template('add.html')
```

```
@app.route('/addexpense',methods=['GET', 'POST'])
```

```
def addexpense():
```

```
    date = request.form['date']
```

```
    expensename = request.form['expensename']
```

```
    amount = request.form['amount']
```

```
    paymode = request.form['paymode']
```

```
    category = request.form['category']
```

```

cursor = mysql.connection.cursor()

    cursor.execute('INSERT INTO expenses VALUES (NULL, % s, % s, % s, % s, % s, % s)',
(session['id'],date, expensename, amount, paymode, category))

mysql.connection.commit()

print(date + " " + expensename + " " + amount + " " + paymode + " " + category)


return redirect("/display")

```

#DISPLAY---graph

```
@app.route("/display")
```

```
def display():
```

```
    print(session["username"],session['id'])
```

```
    cursor = mysql.connection.cursor()
```

```
        cursor.execute('SELECT * FROM expenses WHERE userid = % s AND date ORDER BY
`expenses`.`date` DESC',(str(session['id'])))
```

```
    expense = cursor.fetchall()
```

```
    return render_template('display.html' ,expense = expense)
```

#delete---the--data

@app.route('/delete/<string:id>', methods = ['POST', 'GET'])

def delete(id):

cursor = mysql.connection.cursor()

cursor.execute('DELETE FROM expenses WHERE id = {0}'.format(id))

mysql.connection.commit()

print('deleted successfully')

return redirect("/display")

#UPDATE---DATA

@app.route('/edit/<id>', methods = ['POST', 'GET'])

def edit(id):

cursor = mysql.connection.cursor()

cursor.execute('SELECT * FROM expenses WHERE id = %s', (id,))

row = cursor.fetchall()

print(row[0])

return render_template('edit.html', expenses = row[0])

```
@app.route('/update/<id>', methods = ['POST'])
```

```
def update(id):
```

```
    if request.method == 'POST' :
```

```
        date = request.form['date']
```

```
        expensename = request.form['expensename']
```

```
        amount = request.form['amount']
```

```
        paymode = request.form['paymode']
```

```
        category = request.form['category']
```

```
        cursor = mysql.connection.cursor()
```

```
        cursor.execute("UPDATE `expenses` SET `date` = % s , `expensename` = % s , `amount` = % s, `paymode` = % s, `category` = % s WHERE `expenses`.`id` = % s ",(date, expensename, amount, str(paymode), str(category),id))
```

```
        mysql.connection.commit()
```

```
        print('successfully updated')
```

```
        return redirect("/display")
```

```
#limit
```

```
@app.route("/limit" )
```

```
def limit():
```

```
    return redirect('/limitn')
```

```
@app.route("/limitnum" , methods = ['POST' ])
```

```
def limitnum():
```

```
    if request.method == "POST":
```

```
        number= request.form['number']
```

```
        cursor = mysql.connection.cursor()
```

```
        cursor.execute('INSERT INTO limits VALUES (NULL, % s, % s) ',(session['id'], number))
```

```
        mysql.connection.commit()
```

```
        return redirect('/limitn')
```

```
@app.route("/limitn")
```

```
def limitn():
```

```
    cursor = mysql.connection.cursor()
```

```
    cursor.execute('SELECT limitss FROM `limits` ORDER BY `limits`.`id` DESC LIMIT 1')
```

```
    x= cursor.fetchone()
```

```
    s = x[0]
```

```
    return render_template("limit.html" , y= s)
```


#REPORT

@app.route("/today")

def today():

cursor = mysql.connection.cursor()

cursor.execute('SELECT TIME(date) , amount FROM expenses WHERE userid = %s AND DATE(date) = DATE(NOW()) ',(str(session['id'])))

texpanse = cursor.fetchall()

print(texpanse)

cursor = mysql.connection.cursor()

cursor.execute('SELECT * FROM expenses WHERE userid = % s AND DATE(date) = DATE(NOW()) AND date ORDER BY `expenses`.`date` DESC',(str(session['id'])))

expense = cursor.fetchall()

total=0

t_food=0

t_entertainment=0

t_business=0

t_rent=0

t_EMI=0

t_other=0

for x in expense:

total += x[4]

```
if x[6] == "food":
```

```
    t_food += x[4]
```

```
elif x[6] == "entertainment":
```

```
    t_entertainment += x[4]
```

```
elif x[6] == "business":
```

```
    t_business += x[4]
```

```
elif x[6] == "rent":
```

```
    t_rent += x[4]
```

```
elif x[6] == "EMI":
```

```
    t_EMI += x[4]
```

```
elif x[6] == "other":
```

```
    t_other += x[4]
```

```
print(total)
```

```
print(t_food)
```

```
print(t_entertainment)
```

```
print(t_business)
```

```
print(t_rent)
```

```
print(t_EMI)
```

```
print(t_other)
```

```
return render_template("today.html", texpanse = texpanse, expense = expense, total =
total ,
```

```
    t_food = t_food,t_entertainment = t_entertainment,
```

```
    t_business = t_business, t_rent = t_rent,
```

```
    t_EMI = t_EMI, t_other = t_other )
```

```
@app.route("/month")
```

```
def month():
```

```
    cursor = mysql.connection.cursor()
```

```
    cursor.execute('SELECT DATE(date), SUM(amount) FROM expenses WHERE userid= %s AND
MONTH(DATE(date))= MONTH(now()) GROUP BY DATE(date) ORDER BY DATE(date)
',(str(session['id'])))
```

```
    texpanse = cursor.fetchall()
```

```
    print(texpanse)
```

```
    cursor = mysql.connection.cursor()
```

```
    cursor.execute('SELECT * FROM expenses WHERE userid = % s AND MONTH(DATE(date))=
MONTH(now()) AND date ORDER BY `expenses`.`date` DESC',(str(session['id'])))
```

```
    expense = cursor.fetchall()
```

```
total=0
```

```
t_food=0
```

```
t_entertainment=0
```

```
t_business=0
```

t_rent=0

t_EMI=0

t_other=0

for x in expense:

total += x[4]

if x[6] == "food":

t_food += x[4]

elif x[6] == "entertainment":

t_entertainment += x[4]

elif x[6] == "business":

t_business += x[4]

elif x[6] == "rent":

t_rent += x[4]

elif x[6] == "EMI":

t_EMI += x[4]

elif x[6] == "other":

t_other += x[4]

print(total)

```
print(t_food)
```

```
print(t_entertainment)
```

```
print(t_business)
```

```
print(t_rent)
```

```
print(t_EMI)
```

```
print(t_other)
```

```
return render_template("today.html", texpanse = texpanse, expense = expense, total =  
total ,
```

```
    t_food = t_food,t_entertainment = t_entertainment,
```

```
    t_business = t_business, t_rent = t_rent,
```

```
    t_EMI = t_EMI, t_other = t_other )
```

```
@app.route("/year")
```

```
def year():
```

```
    cursor = mysql.connection.cursor()
```

```
    cursor.execute('SELECT MONTH(date), SUM(amount) FROM expenses WHERE userid= %s  
AND YEAR(DATE(date))= YEAR(now()) GROUP BY MONTH(date) ORDER BY MONTH(date)  
,(str(session['id']))')
```

```
    texpanse = cursor.fetchall()
```

```
    print(texpanse)
```

```
    cursor = mysql.connection.cursor()
```

```
    cursor.execute('SELECT * FROM expenses WHERE userid = % s AND YEAR(DATE(date))=  
YEAR(now()) AND date ORDER BY `expenses`.`date` DESC',(str(session['id']))')
```

```
expense = cursor.fetchall()
```

```
total=0
```

```
t_food=0
```

```
t_entertainment=0
```

```
t_business=0
```

```
t_rent=0
```

```
t_EMI=0
```

```
t_other=0
```

```
for x in expense:
```

```
    total += x[4]
```

```
    if x[6] == "food":
```

```
        t_food += x[4]
```

```
    elif x[6] == "entertainment":
```

```
        t_entertainment += x[4]
```

```
    elif x[6] == "business":
```

```
        t_business += x[4]
```

```
    elif x[6] == "rent":
```

```
        t_rent += x[4]
```

```
    elif x[6] == "EMI":
```

```
        t_EMI += x[4]
```

```
elif x[6] == "other":
```

```
    t_other += x[4]
```

```
print(total)
```

```
print(t_food)
```

```
print(t_entertainment)
```

```
print(t_business)
```

```
print(t_rent)
```

```
print(t_EMI)
```

```
print(t_other)
```

```
    return render_template("today.html", texpanse = texpanse, expense = expense, total =  
total ,
```

```
        t_food = t_food,t_entertainment = t_entertainment,
```

```
        t_business = t_business, t_rent = t_rent,
```

```
        t_EMI = t_EMI, t_other = t_other )
```

```
#log-out
```

```
@app.route('/logout')
```

```
def logout():
```

```
session.pop('loggedin', None)

session.pop('id', None)

session.pop('username', None)

return render_template('home.html')
```

```
if __name__ == "__main__":

    app.run(debug=True)
```

GITHUB LINK :

<https://github.com/IBM-EPBL/IBM-Project-48680-1660811246>

DEMO LINK :

<https://youtu.be/MghyjM0rbdg>