PERSONAL EXPENSE TRACKER

TEAM DETAILS

Team No :PNT2022TMID43079
College Name: Sree Sakthi Engineering College
Department :Computer Science & Engineering

Team Leader : Swetha S
Team Member 1 : Kaviyan S
Team Member 2 : Vanitha P
Team Member 3 : Kathiresan S
Team Member 4 : Sakthivel R

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"PERSONAL EXPENSE TRACKER APPLICATION"** is the bonafidework of **"SWETHA S (713619104041), VANITHA P (713619104044), KATHIRESAN S (713619104015), KAVIYAN S (713619104016) SAKTHIVEL R(713619104030)",** who carried out the project work undermy supervision.

SIGNATURE:

Mrs K SELVASHEELA,
MCA.,M.E., Ph.D,

HEAD OF THE DEPARTMENT,

Dept. of Computer Science and Engg.,
Sree Sakthi Engineering College,
Karamadai, Coimbatore-641104

SIGNATURE:

Ms. S VINOTHINI,
M.E(CSE),

MENTOR,

Dept. of Computer Science and Engg.,
Sree Sakthi Engineering College,
Karamadai, Coimbatore-641104

The viva-voce is held on_____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENTS

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 & 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.
money to meet their needs.

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 REFERENCES

[1]. Palestinian Ministry of Education and Higher Education. Palestinian Higher Education Statistics
[2]. 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
[3]. Engineering Association of Palestine. Current Engineering Statistics Book. Ramallah; 2005
[4]. 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.
[5]. Chen JW, Yen M. Engineering Accreditation: A Foundation for Continuing Quality Improvement. 2005 Mar 1–5; Tainan. Exploring Innovation in Education and Research.
[6]Reto Meier, “Professional Android™ 2Application Development”, published by Wiley publishing, 2010.
[7]Zigurd Mednieks (Goodreads Author), Laird Dornin, G. Blake Meike, Masumi Nakamura,Programming Android, published by O'Reilly Media,2011.

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

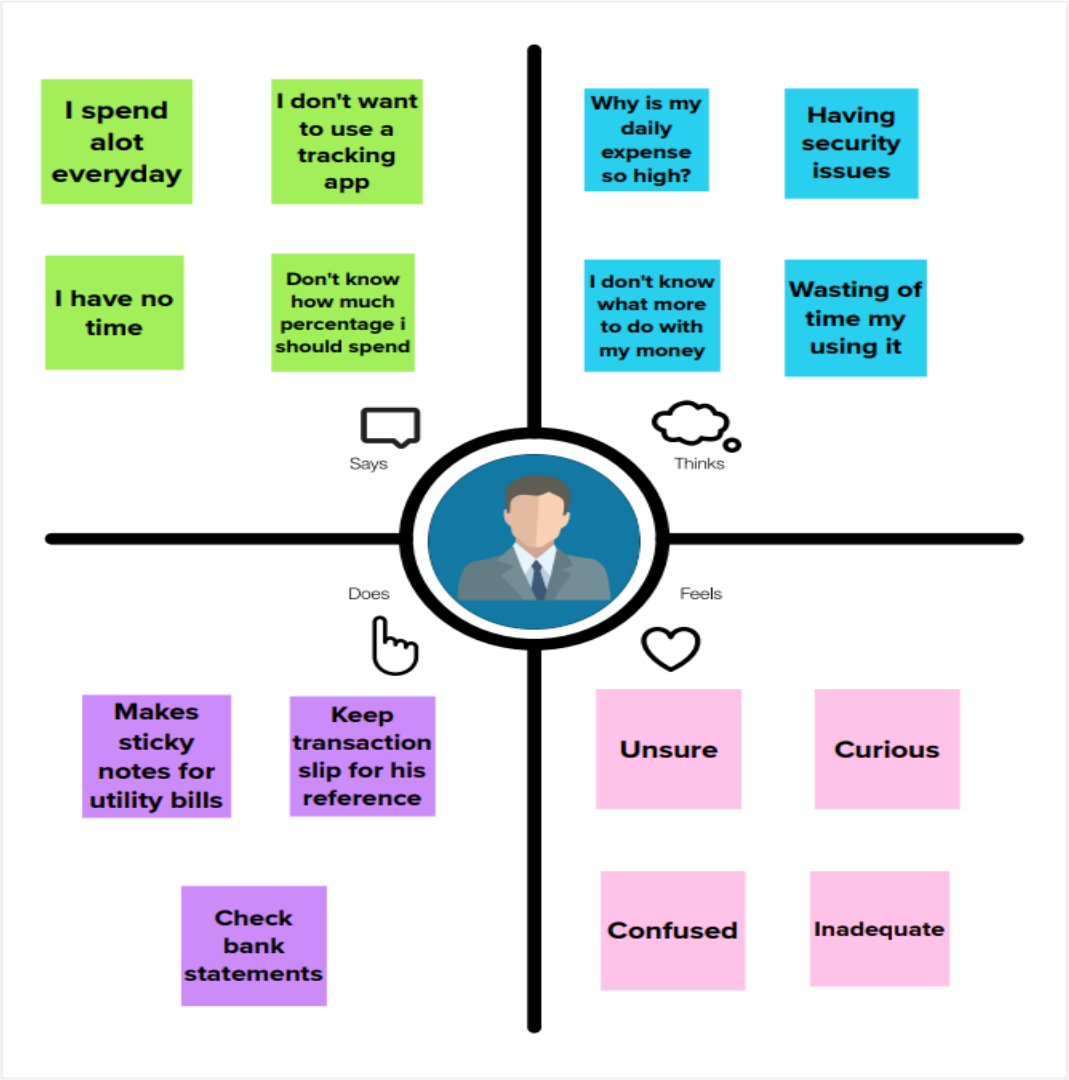
3.1 EMPATHY MAP CANVAS

Edit this template
Right-click to unlock

Empathy Map

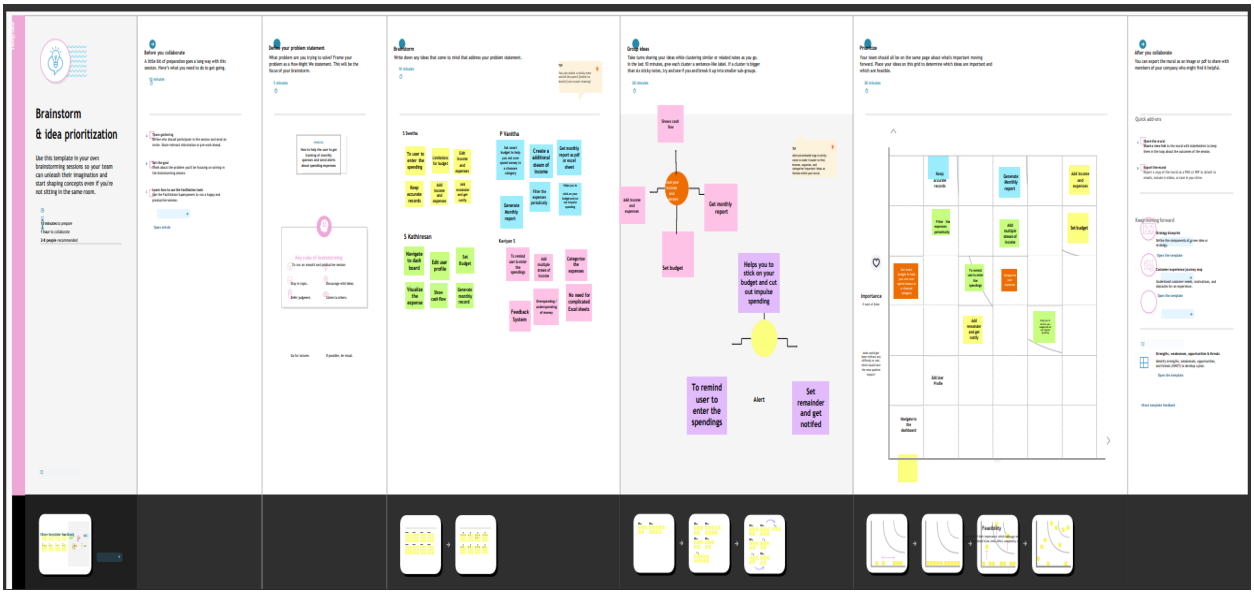
Dive into the mind of the user for focused product development

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



Share your feedback

3.2 IDEATION & BRAINSTORMING

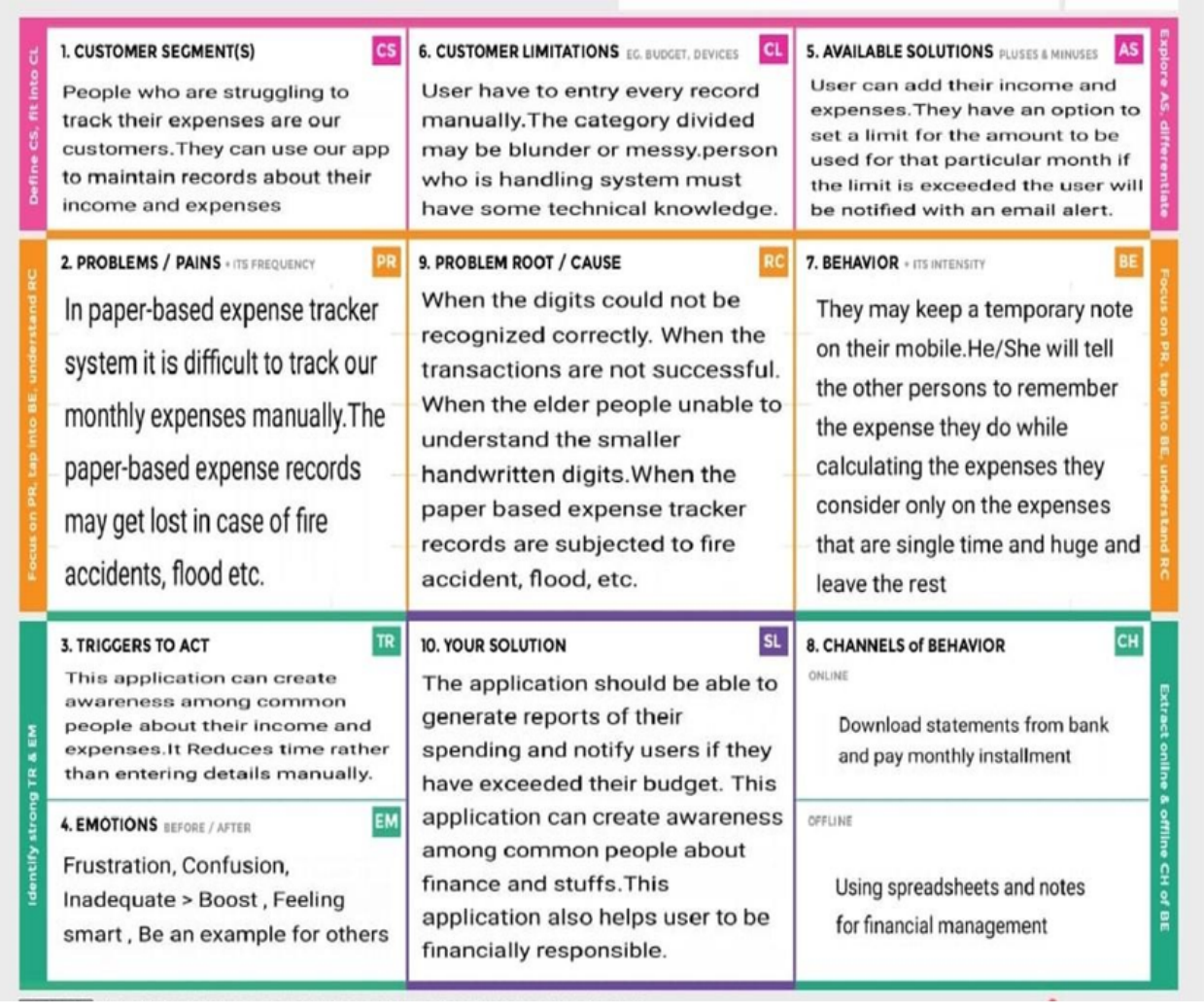


3.3 PROPOSED SOLUTION

S.NO.	Parameter	Description
1.	Problem Statement	In paper-based expense tracker system it is difficult to track our monthly expenses manually. 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.
2.	Scalability of the Solution	This application can handle large number of users and data with high performance and security. This

		application can adapt for both large-scale and small-scale purposes. Easily available in all kinds of devices.
3.	Idea / Solution description	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.
4.	Novelty / Uniqueness	The user gets notified when their expense exceeds the limit and also it reminds the user when they forgot to make entry. Tracking expenses through SMS. Data analytics on expenses. Future expense predictio
5.	Social Impact / Customer Satisfaction	The application should be able to generate reports of their spending and notify users if they have exceeded their budget. It is designed to be dynamic to produce the prediction. It also provides users' personal information, their income as well as their expenses. This application can create awareness among common people about finance and stuffs. This application also helps user to be financially responsible. It Reduces time rather than entering details manually.
6.	Business Model (Revenue Model)	This Application is provided for free of cost. But It will have some advertisement. In premium version there is no advertisement and contains some additional features.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Form for collecting details
FR-2	Login	Enter username and password
FR-3	Calendar	Personal expense tracker application must allow user to add the data to their expenses.

FR-4	Expense Tracker	This application should graphically represent the expense in the form of report.
FR-5	Report generation	Graphical representation of report must be generated.
FR-	Category	This application shallallow users to add categories of their expenses.

4.2 NON-FUNCTIONAL REQUIREMENTS

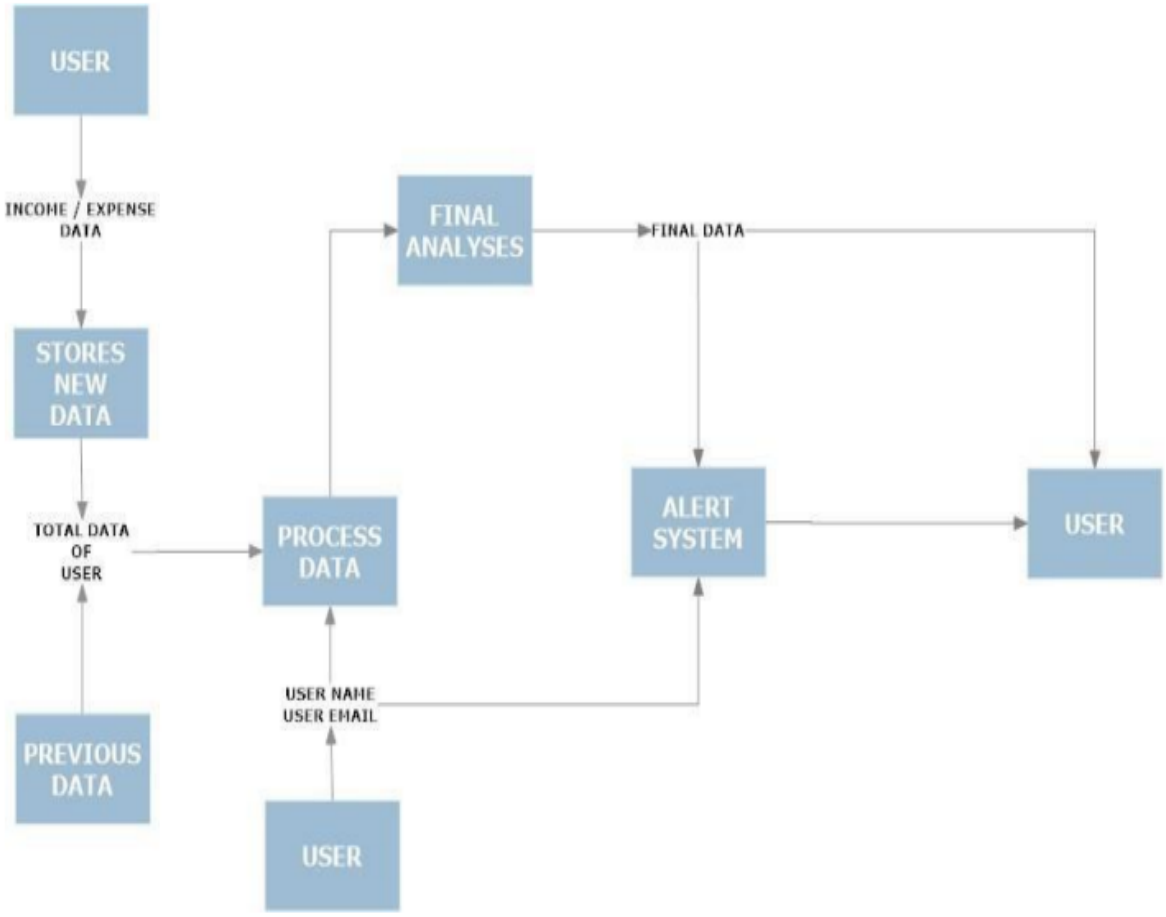
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Helps to keep an accurate record of yourincome and expenses.
NFR-2	**Security**	Budget tracking apps are considered very safe from those who commit cyber crimes.
NFR-3	**Reliability**	Each data recordis stored on a wellbuilt efficient database schema. There is no risk of data loss.
NFR-4	**Performance**	The types of expense are categories along with an option. Throughput of the systemis increased due to light weight database support.
NFR-5	**Availability**	The application must have a 100% up-time.
NFR-6	**Scalability**	The abilityto appropriately handleincreasing demands.

5. PROJECT DESIGN

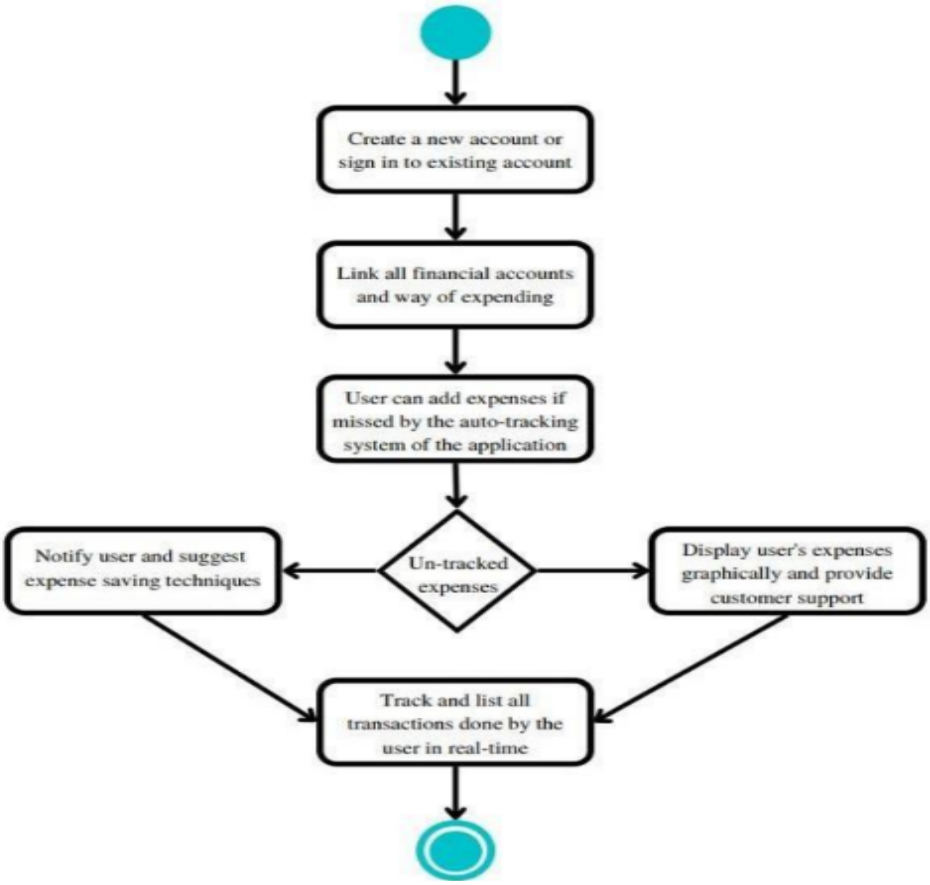
5.1 DATA FLOW DIAGRAMS:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system

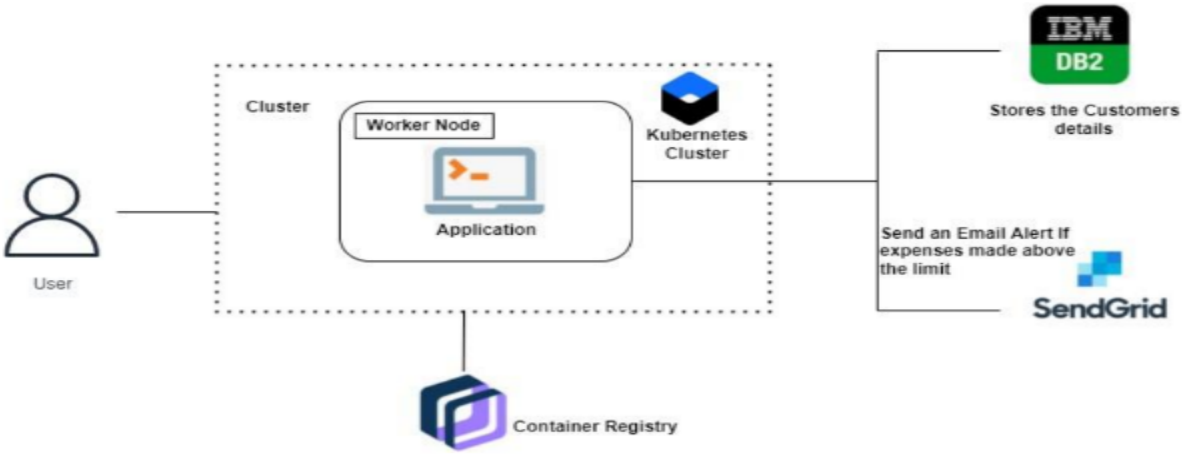
requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION ARCHITECTURE:



TECHNICAL ARCHITECTURE:



5.3 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 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	Medium	

				application		

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement(Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user,I can register for the application by entering my email,password, and confirming my password.	2	High	Swetha
		USN-2	As a user,I will receiveconfirmation email oncel have registered for the application	1	High	Kaviyan, Vanitha
	Login	USN-3	As a user, I can loginto the application by entering email & password	1	High	Kathiresan
	Dashboard	USN-4	Logging in takes to the dashboard for the logged user.	`2	High	Sakthivel
Bug fixes, routine checks and improvisation by everyone in the team*Intended bugs only						

Sprint 2	Workspace	USN-1	Workspace for personal expensetracking	2	High	Kathiresan
	Charts	USN-2	Creating various graphs and statistics of customer's data	1	Medium	Kaviyan
	Connecting to IBM DB2	USN-3	Linking database with dashboard	2	High	Swetha, Vanitha
		USN-4	Making dashboard interactive with JS	2	High	Sakthivel
Sprint-3		USN-1	Wrapping up the serverside works of frontend	1	Medium	Vanitha
	Watson Assistant	USN-2	Creating Chatbotfor expense tracking and for clarifying user's query	1	Medium	Swetha
	SendGrid	USN-3	Using SendGrid to send mailto the userabout their expenses	1	Low	Sakthivel, Kaviyan

		USN-4	Integrating bothfrontend and backend	2	High	Kathiresan
Bug fixes, routine checks and improvisation by everyone in the team*Intended bugs only						
Sprint-4	Docker	USN-1	Creating imageof website usingdocker/	2	High	Kaviyan
	Cloud Registry	USN-2	Uploading dockerimage to IBM Cloud registry	2	High	Vanitha
	Kubernetes	USN-3	Create container using the dockerimage and hosting the site	2	High	Swetha
	Exposing	USN-4	Exposing IP/Ports for the site	2	High	Kathiresan, Sakthivel

6.2 SPRINT DELIVERY SCHEDULE:

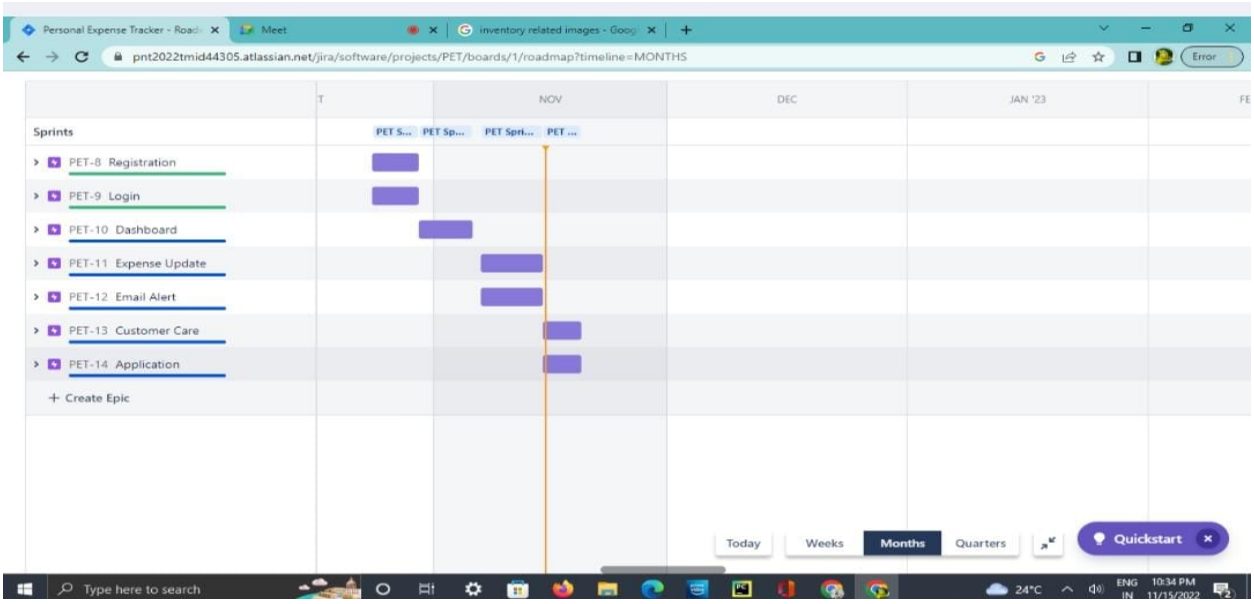
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	23 Oct 2022	28 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	30 Oct 2022	4 NOV 2022	20	05 NOV 2022
Sprint-3	20	6 Days	06 NOV 2022	11 NOV 2022	20	12 NOV 2022
Sprint-4	20	6 Days	13 NOV 2022	18 NOV 2022	20	19 NOV 2022

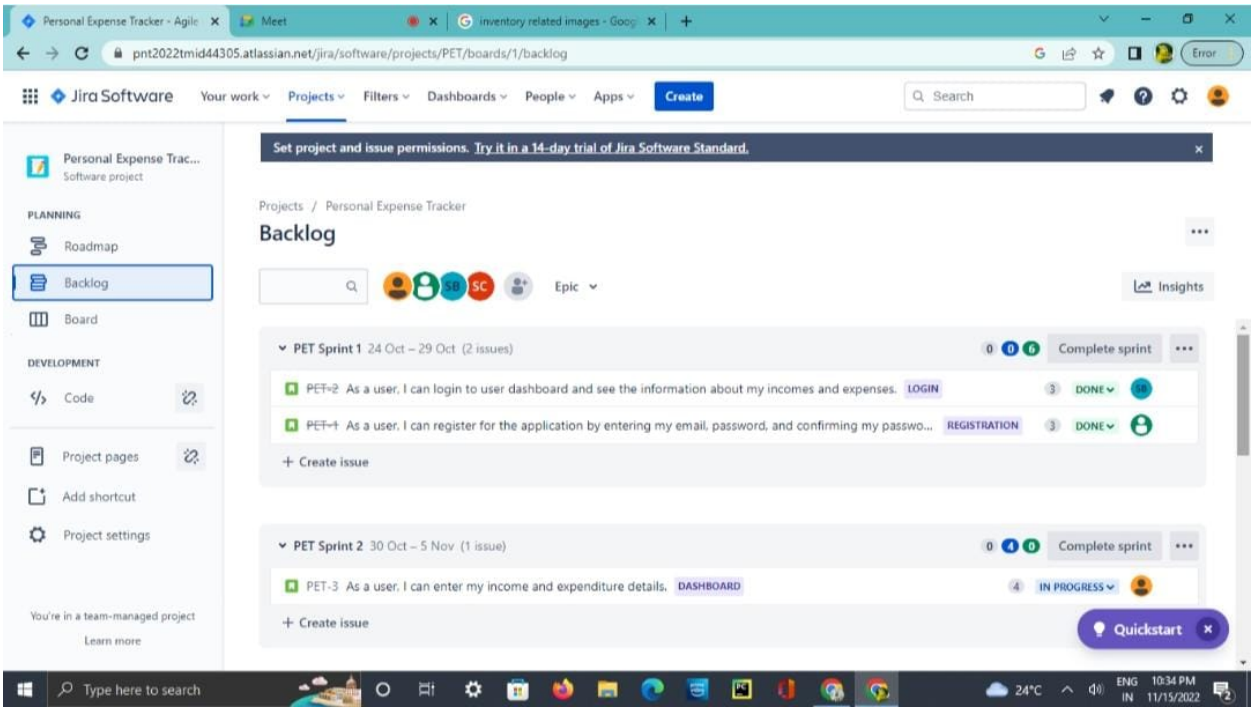
6.3 REPORTS FROM JIRA

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 points per)

$$AV = \text{sprint duration / velocity} = 20/6 = 3.33$$





7.CODING & SOLUTIONING

7.1 FEATURE 1(FRONT END):

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>, , , <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> 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.

BOOTSTRAP

Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.It is absolutely free to download and use.It is a front-end framework used for easier and faster web development.It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.It can also use JavaScript plug-ins.It facilitates you to create responsive designs.

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter.It was released as an open source product in August 2011 on GitHub.In June 2014 Bootstrap was the No.1 project on GitHub.It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap. It facilitates users to develop a responsive website. It is compatible on most browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

7.2 FEATURE 2(BACKEND):

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 Poocco.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 Pocoo projects Werkzeug and Jinja were developed.In April 2016, the Pocoo 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.

7.3 DATABASE SCHEMA:

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 1s 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 innerworkings, 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

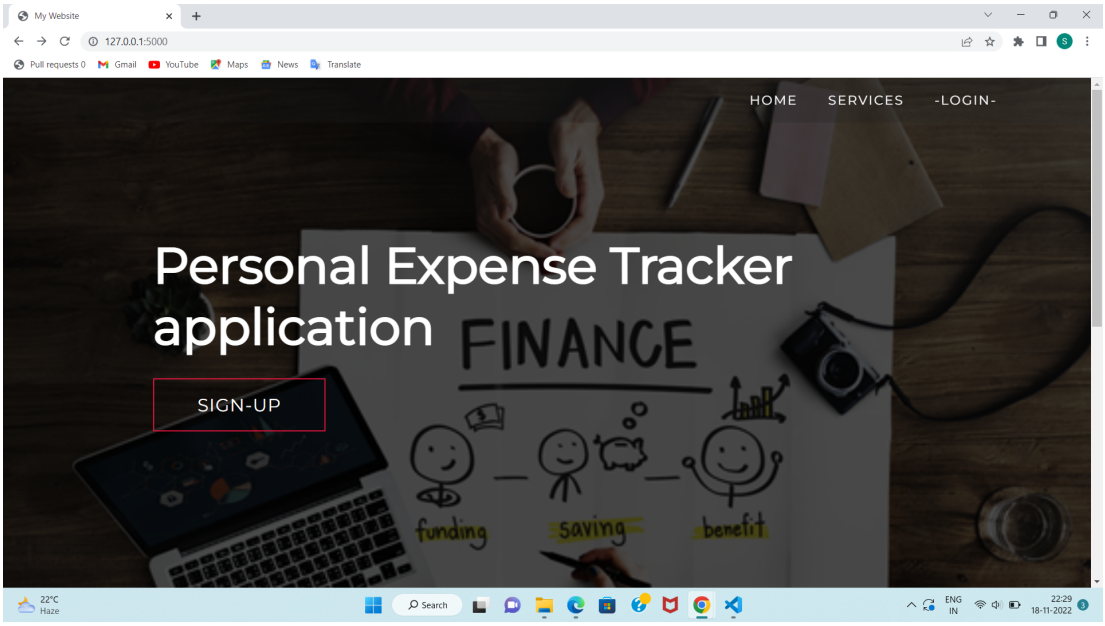
Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.The task of the integration test is to check that components or software applications, e.g. components in a software system or-one step up- software

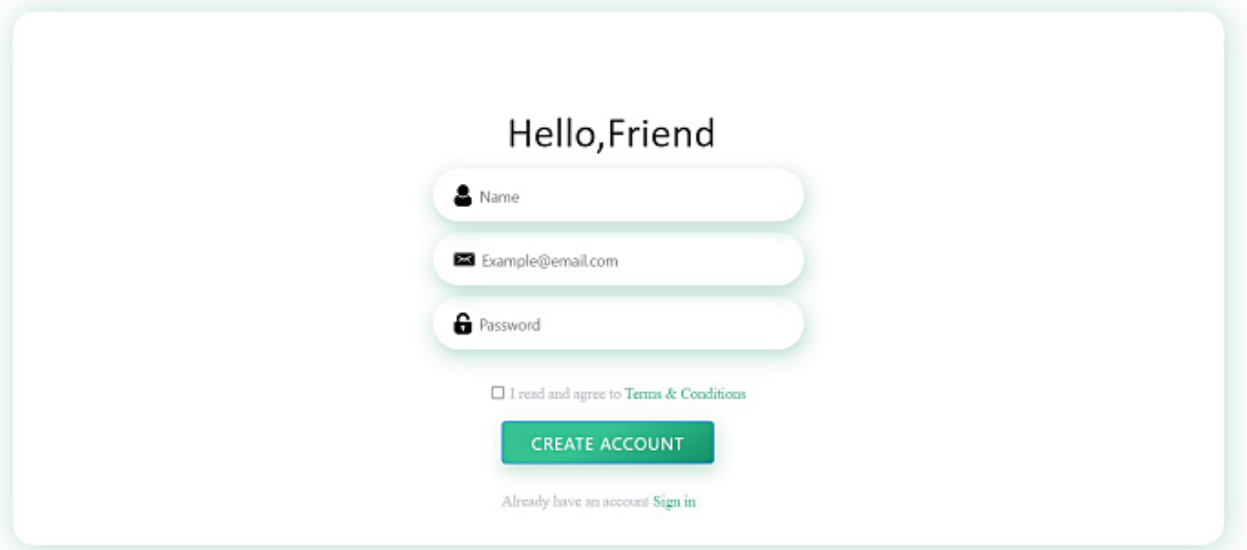
applications at the company level - interact without error.
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.2 USER ACCEPTANCE:

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





It has been implemented successfully and executed.

10.ADVANTAGES

- Track your expenses anywhere, anytime.
- Seamlessly manage your money and budget without any financial paperwork. ...
- Access, submit, and approve invoices irrespective of time and location.
- Avoid data loss by scanning your tickets and bills and saving in the app.
- Effective expense tracking and reporting to avoid conflict. As a project manager or business owner, you can set clear policies for the expense types and reimbursement limits to avoid misunderstandings are about costs.

DISADVANTAGES

- Difficulty finding a budgeting method that works for you
- Budgeting takes time and effort
- You may think it's too rigid

11.CONCLUSION

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. Since the screen provides online help messages and is very user friendly, any user will get familiarized with its usage.The project personal expensive tracker has been successfully implemented by using python, flask, html/css/java script and the database created by using ibm db2 and also successfully executed and implemented.

12.FUTURE SCOPE

Now in our application we covered almost all features but in future we will add some more futures. The features are below

- Multiple account support.
- Include currency converter.
- Reports are created in category wise any format
- It will have various options to keep record (for example Food, Travelling Fuel, Salary).
- Automatically it will keep on sending notifications for our daily expenditure.

13.APPENDIX

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 >
```

```
<img class="wave" src="..\static\images\wave.png">
<div class="container">

        <div class="img">
                <div id="png"><a href="/" title="HOME"><img style="width:75px;
height:75px ; " src="..\static\images\home-page.png"></a></div>
                <img src="..\static\images\bg.svg">
        </div>


        <div class="login-content">

                <form action='/login' method="POST">
                        <div class="msg">{{ msg }}</div>
                        <img src="..\static\images\avatar.svg">
                        <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>
        <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>
```

```
        <a href="#">Forgot Password?</a>
        <input type="submit" class="btn" value="Login">
                                <span>OR</span>

                                <div><b>Login with</b></div>
                                <div>
                                        <ul>
                                                <li><a href="#"><i class="fab fa-facebook" aria-
hidden="true"></i></a></li>
                                                <li><a href="#"><i class="fab fa-twitter" aria-
hidden="true"></i></a></li>
                                                <li><a href="#"><i <i class="fab fa-google"  aria-
hidden="true"></i></a></li>
                                                <li><a href="#"><i class="fab fa-linkedin" aria-
hidden="true"></i></a></li>
                                                <li><a href="#"><i class="fab fa-instagram" aria-
hidden="true"></i></a></li>
                                        </ul>

                                </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>
```

HOMEPAGE

```
% 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">
        <img src="../static/images/images (13).jpeg">
        <h1>LET START JOURNEY</h1>
        <P>MyBudget web application helps<br> you to maintain budget<br>
            and analyse the expense</P>
        <img height="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 %}
```

APP.PY

```
from flask import Flask, render_template, request, redirect, session
from flask_mysqldb 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'])))
     texpense = cursor.fetchall()
```

```
    print(texpense)

    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", texpense = texpense, 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'])))
    texpense = cursor.fetchall()
    print(texpense)

    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", texpense = texpense, 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'])))
    texpense = cursor.fetchall()
    print(texpense)

    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", texpense = texpense, 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-48659-1660811088

PROJECT DEMO LINK:

https://www.loom.com/share/fb1c9e45829344ba94cbf4902fb60b57