PERSONAL EXPENSE TRACKER

IBM-Project-25860-1659975210

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

A PROJECT REPORT BY

Jeganraj S
Nandhakumar M
Esvar A
Kaushan R

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

MOOKAMBIGAI COLLEGE OF ENGINEERING

INDEX

3. Reports from JIRA

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

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

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. Project Overview
   Category: Cloud Application Development
   Team ID:PNT2022TMID45588

💻 **Skills Required:**

HTML,CSS,

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.

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

1. Existing problem

In a study conducted by Forrester in 2016 surveying small and medium businesses (SMBs) across the world, 56% companies reported expense management as being the biggest challenge for their finance departments.

In another survey conducted by Levvel Research in 2018 in North America, respondents reported the following pain points in expense management before adopting automation:

- Manual entry and routing of expense reports (62%)
- Lack of visibility into spend data (42%)
- Inability to enforce travel policies (29%)
- Lost expense reports (24%)
- Lengthy expense approval system and reimbursement cycles (23%)

1.References

S.No	Topic	Proposed Work
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.

S.No	Topic	Proposed Work
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. The main idea of developing this feature for giving users a clear picture that how much they are spending and where they are spending and when. We remind them to pay their utilities and card bills before the due date by using the same platform in just one tap, instead of going any other way. Also serving them by giving saving tips for their good future investment.

3.Problem Statement Definition

Customer Problem Statement :

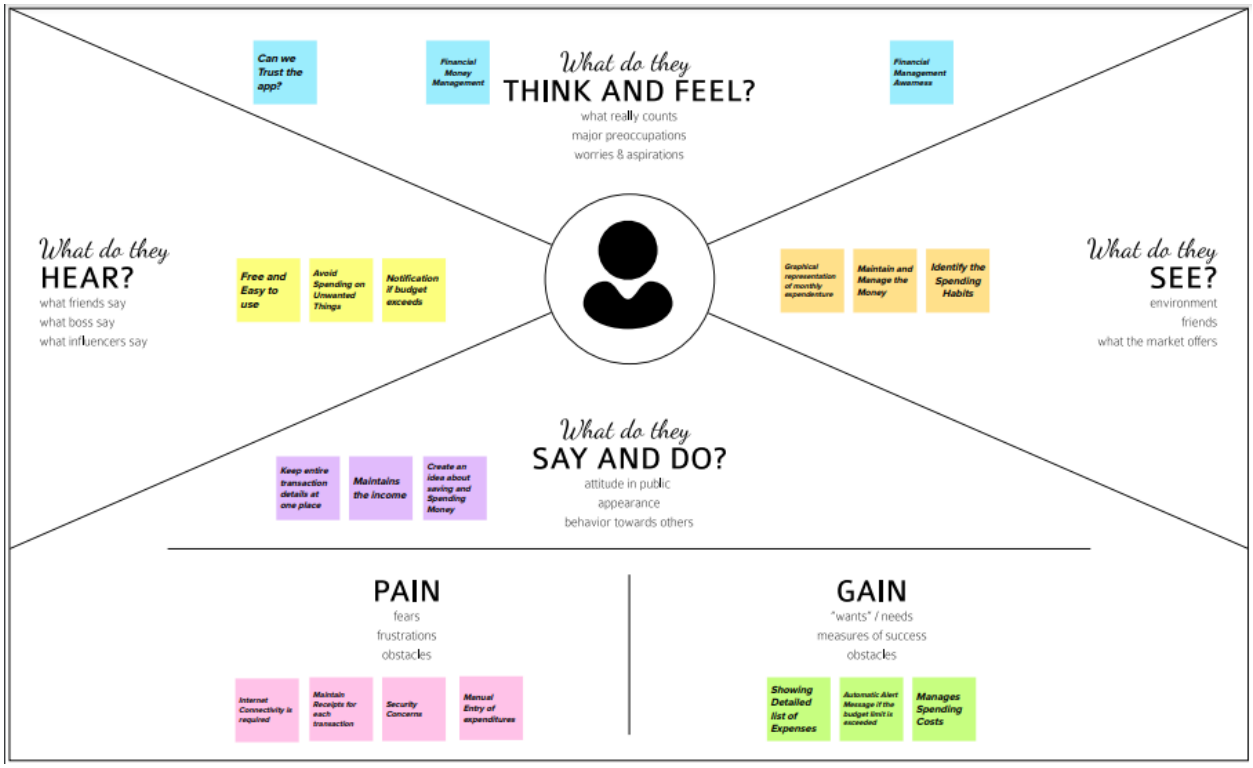
Many organizations have their own system to record their income and expenses which they feel is the main key point of their business progress. It is a good habit for a person to record daily expenses and earning but due to unawareness and lack of proper applications to suit their privacy, lacking decision 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.

Who does the problem affect?	People getting regular wages.
What is the issue?	The paper based expense tracker system does not provide the user portability , existing system only used on paper based records so unable to update anywhere expenses done and unable to update the location of the expense details disruptive that the proposed system.
When does the issue occurs?	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.
Where is the issue occurring?	The issue occurs when the person is unable to track his income and expenditure.
Why is it important that we fix theproblem?	By solving this issue those people getting regular wages can track their expenses and avoid unwanted expenses.

3.IDEATION & PROPOSED SOLUTION

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 helps 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.

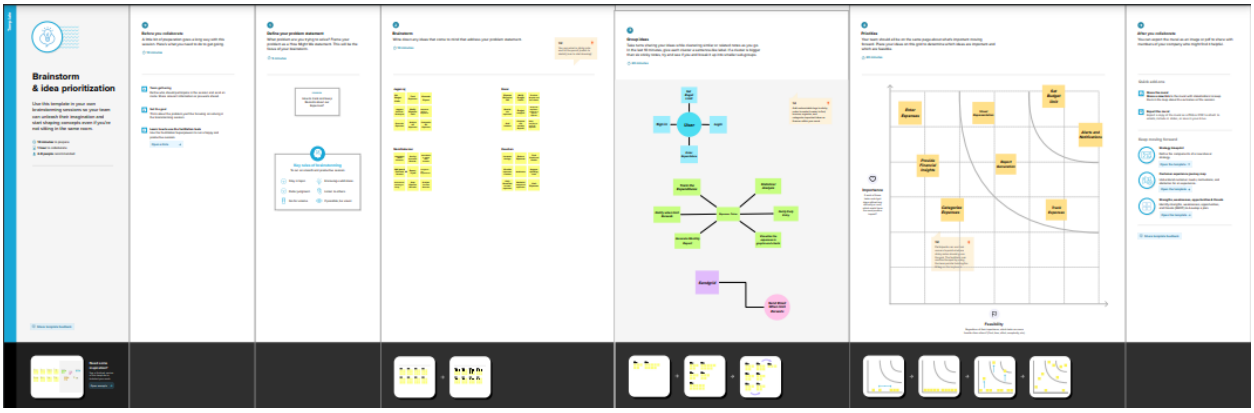


1.Ideation & Brainstorming

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

Step-2 : Brainstorm,Idea Listing and Grouping

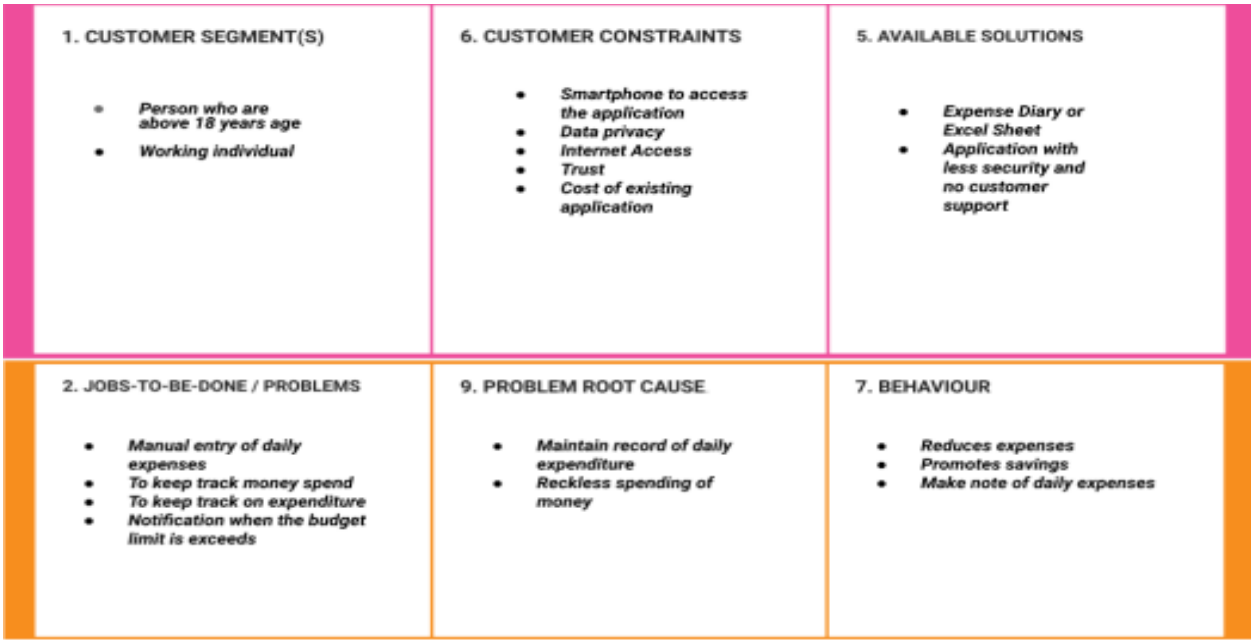
Step-3: Idea Prioritization



3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	At the end of the month we start to have money crisis. Lack of proper planning of our income. Person has to keep a log in a diary or in a computer. All the calculations need to be done by the user. Overload to rely on the daily entry of the expenditure.
2.	Idea / Solution description	An expense tracker app allows you to monitor and categorize your expenses.This application will be helpful for us in not just managing their expenses, but also in enabling them to improve their investments.
3.	Novelty / Uniqueness	Expense tracker apps help you collect and classify your purchases so that you can identify areas that might be trimmed.The user gets a notification when the budget limit exceeds.
4.	Social Impact / Customer Satisfaction	Make wise decision Manage your expenses Budget planning can be done Makes report and give accurate survey.
5.	Business Model (Revenue Model)	Cost Effective one.
6.	Scalability of the Solution	Improves financial management. Secured and safe to use. Insights about money management.

4. Problem Solution fit



4.REQUIREMENT ANALYSIS

1.Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirements (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Application Registration through E-mail
FR-2	User Confirmation	Confirmation via E-mail 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	Alert through E-mail Alert through SMS
FR-6	User Budget Plan	Planning and Tracking of user expense vs budget limit

2.Non-Functional requirements

Non-Functional Requirements:

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

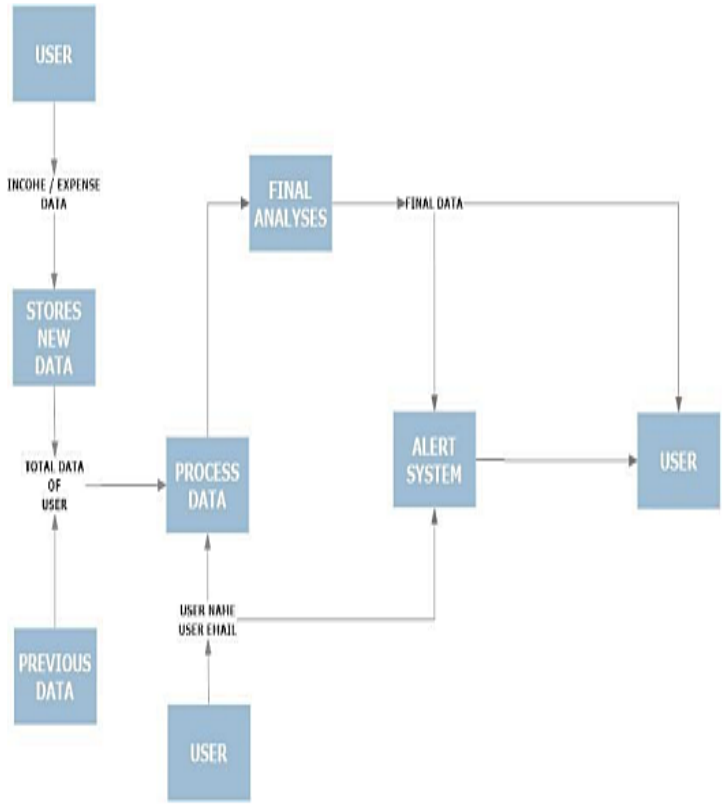
NFR No.	Non-Functional Requirements	Description
NFR-1	Usability	Effectiveness, efficiency and overall satisfaction of the user while interacting with our application.
NFR-2	Security	Authentication, authorization, encryption of the application.
NFR-3	Reliability	Probability of failure-free operations 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.

5. PROJECT DESIGN

DATAFLOW DIAGRAM

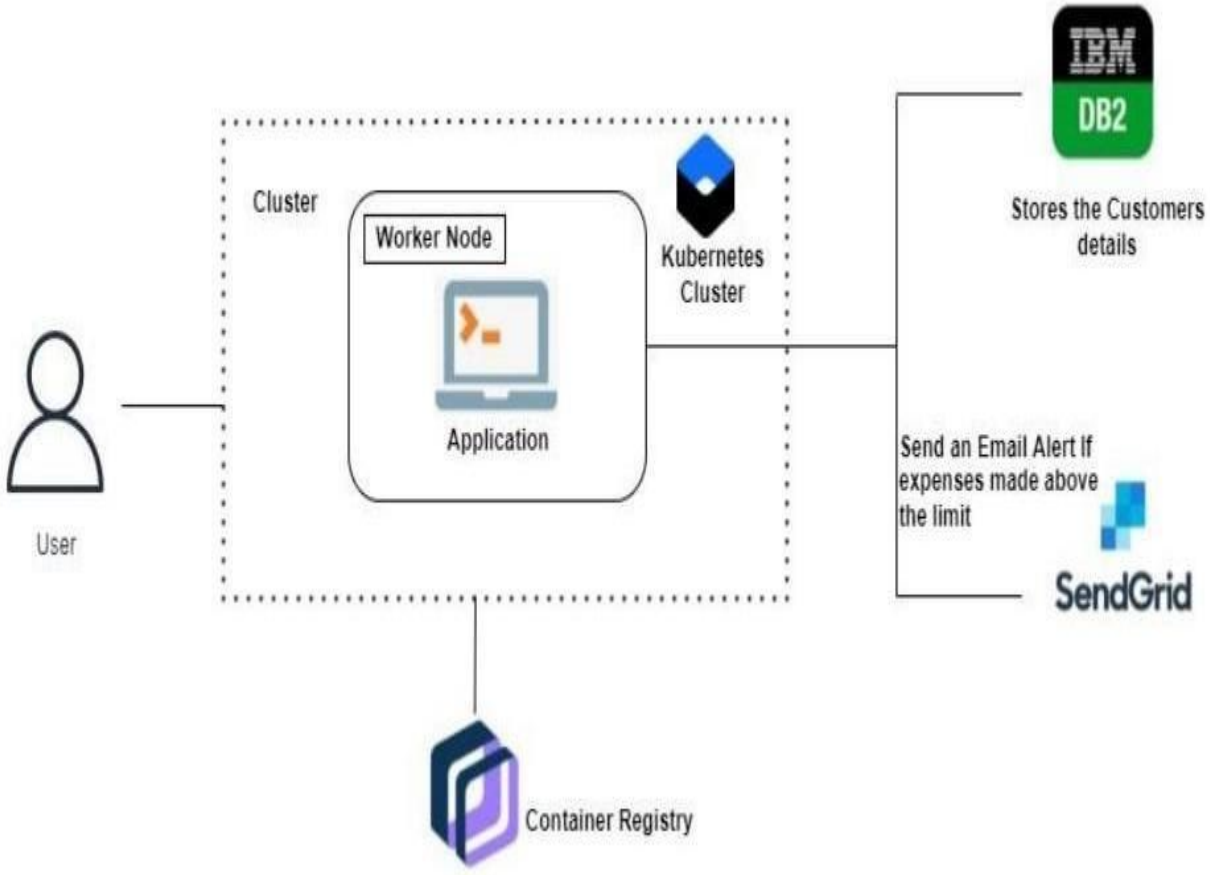
Data Flow Diagrams:

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.

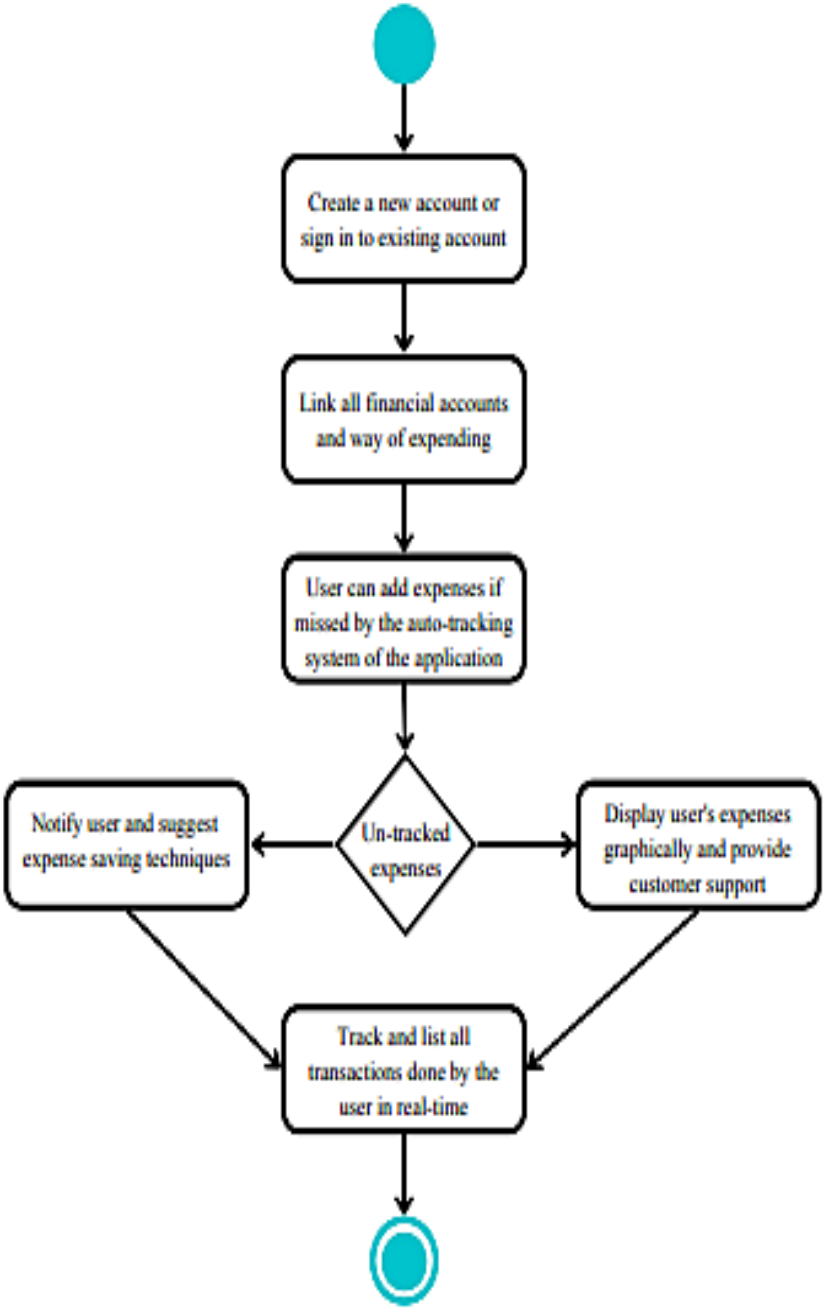


2.Solution & Technical Architecture

Technical Architecture:



SOLUTION ARCHITECTURE



3. User Stories

User Stories

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

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

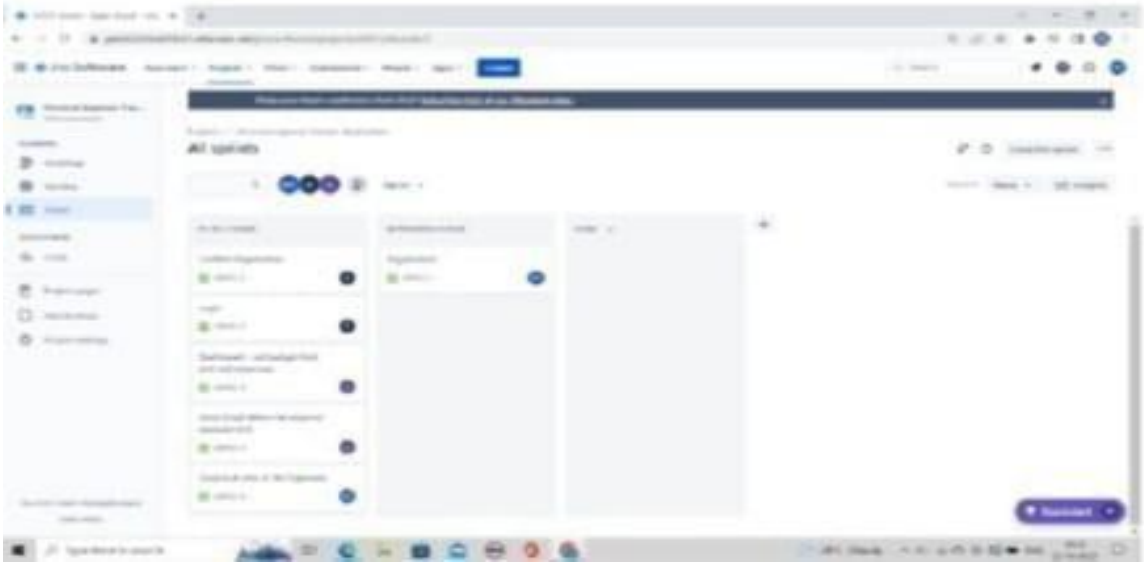
Velocity

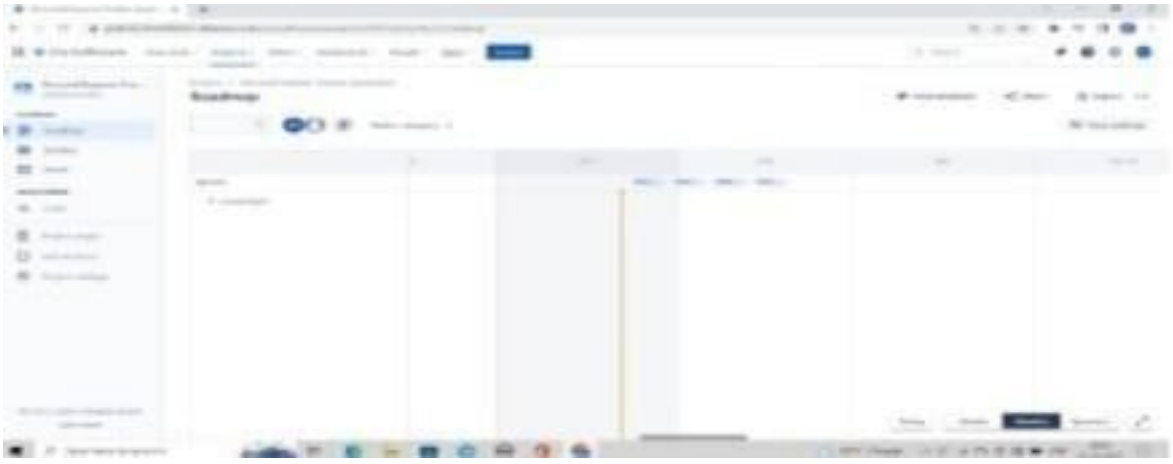
We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Calculating the team's average velocity (AV) per iteration unit (story points per day)

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

2.Sprint Delivery Schedule

2. Reports from JIRA





7. CODING & SOLUTION

1.Feature 1

We have added an option to users to create their own expense categories by typing category and a small description about the category for notes.

addcategory.html

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

 {% block title %} Category {% endblock %}

<script>
 document.querySelector('.category').classList.add('active');
 </script>
<div class="container-fluid">
 <div class="d-sm-flex align-items-center justify-content-between mb-4">
  <h1 class="h3 mb-0 text-gray-800">Category
  </h1>
 </div>
 <div class="align-items-center">
  {% with messages = get_flashed_messages() %}
  {% if messages %}
  {% for message in messages %}
  <div class="alert alert-success alert-dismissible" role="alert">
   <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span
```

```
        aria-hidden="true">&times;</span></button>
    {{ message }}
  </div>
  {% endfor %}
  {% endif %}
  {% endwith %}
 </div>

 <div class="row">

  <div class="col-xl-8 col-lg-7">
    <div class="card shadow mb-4">
<div class="card-header py-3 d-flex flex-row align-items-center justify-content-between">
<h4 class="m-0 font-weight-bold text-primary">Add Category</h4>
      </div>

      <div class="card-body">
       <form action="{{url_for('addCategory')}}" method="POST">
        <div class="form-outline  pb-3">
          <label class="form-label" for="form2Example1">Category Name:</label>
          <input type="text" name="name" id="form2Example1" class="form-control-sm form-control" required />
        </div>
        <div class="form-outline  pb-3">
          <label class="form-label" for="form2Example2">Description:</label>
```

```
          <input type="text" name="description" id="form2Example2"
class="form-control-sm form-control" required />
         </div>
         <button type="submit" class="btn btn-primary align-item-center mt-
2">ADD</button>
        </form>
       </div>
     </div>
   </div>


   <div class="col-xl-4 col-lg-5">
    <div class="card shadow mb-4">
     <div class="card-header py-3 d-flex flex-row align-items-center justify-
content-between">
       <h4 class="m-0 font-weight-bold text-primary">Category List</h4>
       <div class="dropdown no-arrow">
        <a class="dropdown-toggle" role="button" id="dropdownMenuLink"
data-toggle="dropdown" aria-haspopup="true"
          aria-expanded="false">
          <i class="fa fa-ellipsis-v" aria-hidden="true"></i>
         </a>
         <div class="dropdown-menu dropdown-menu-right shadow
animated--fade-in" aria-labelledby="dropdownMenuLink">
          <div class="dropdown-header">Default Categories</div>
  <a class="dropdown-item">Food</a>
          <a class="dropdown-item">Shopping</a>
          <a class="dropdown-item">Transport</a>
```

```
          <a class="dropdown-item">Movie</a>
          <a class="dropdown-item">Healthcare</a>
        </div>
      </div>
    </div>

    <div class="card-body">
     {% if CAT %}
     <table class="table table-sm">

       <thead>
         <tr>
           <th scope="col">Category</th>
           <th scope="col">Option</th>
         </tr>
       </thead>
       <tbody>

         {% for cate in CAT %}

         <tr>
           <td class="p-2">{{cate["NAME"]}}</td>
           <td class="p-2"><a href="/delete/{{cate['ID']}}/" class="btn btn-sm
btn-danger">Delete</a></td>
         </tr>
         {% endfor %}
         {% else %}
```

```
        </tbody>
      </table>
      <div class="text-center">No Category</div>
      {% endif %}
    </div>
  </div>
</div>
{% endblock %}
```

app.py

```
@app.route("/AddCategory", methods=['GET', 'POST'])
def addCategory():
  if request.method == "GET":
    sql = "SELECT * FROM category WHERE userid=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, session['id'])
    ibm_db.execute(stmt)
    categ=[]
    data = ibm_db.fetch_assoc(stmt)
    while data!=False:
      categ.append(data)
      data=ibm_db.fetch_assoc(stmt)
    return render_template('addCategory.html',CAT=categ)
  if request.method=="POST":
```

```
        insert_sql = "INSERT INTO category(userid,name,description) VALUES
(?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1,session['id'])
        ibm_db.bind_param(prep_stmt, 2,request.form['name'].capitalize())
        ibm_db.bind_param(prep_stmt, 3,request.form['description'])
        ibm_db.execute(prep_stmt)
        flash('Category added successfully')
        return redirect(url_for('addCategory'))



@app.route('/delete/<id>/')
def deleteCategory(id):
sql = "DELETE FROM CATEGORY WHERE ID=?"
prep_stmt = ibm_db.prepare(conn, sql)
   ibm_db.bind_param(prep_stmt, 1,id)
   ibm_db.execute(prep_stmt)
   flash("Category deleted successfully")
   return redirect(url_for('addCategory'))
```

2.Feature 2

- We have added the data visualization on methods for expenditure.
- The pie chart has 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 changes over me
  .
- In addition we have included linechart for displaying monthly expenses in graphical form.
- And also included barchart to visualize total expenses based on category in a year.

Therefore, their use should be considered if:

- You are looking to categorize and compare a set of data.

- You only have positive values.

Dashboard.html

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

{% block title %}Dashboard{% endblock %}

{% block body %}
<script>
document.querySelector('.dashboard').classList.add('activ
e');
</script>

{% if wallet and expense["1"]>0 %}
<div class="container-fluid">

  <div class="d-sm-flex align-items-center justify-content-
between mb-4">
    <h1 class="h3 mb-0 text-gray-800">Dashboard</h1>
  </div>


  <div class="row">

    <div class="col-xl-3 col-md-6 mb-2">
```

```
        <div class="card border-left-primary shadow h-100
py-2">
            <div class="card-body">
              <div class="row no-gutters align-items-center"
  <div class="col mr-2">
        <div class="text-xs font-weight-bold text-primary
text-uppercase mb-1">
                        Wallet Amount</div>
                    <div class="h5 mb-0 font-weight-bold text-
gray-800">{{wallet["AMOUNT"]}}
                    </div>
                </div>
              </div>
            </div>
          </div>
      </div>

      <div class="col-xl-3 col-md-6 mb-2">
        <div class="card border-left-success shadow h-100
py-2">
            <div class="card-body">
              <div class="row no-gutters align-items-center">
                <div class="col mr-2">
                    <div class="text-xs font-weight-bold text-
success text-uppercase mb-1">
                        This Month Expense</div>
                    <div class="h5 mb-0 font-weight-bold text-
gray-800">
```

```
                    {{current_month_expense["1"]}}</div>
                </div>
            </div>
        </div>
    </div>

    <div class="col-xl-3 col-md-6 mb-2">
        <div class="card border-left-warning shadow h-100
py-2">
            <div class="card-body">
                <div class="row no-gutters align-items-center">
                    <div class="col mr-2">
                        <div class="text-xs font-weight-bold text-
warning text-uppercase mb-1">
                            Wallet Balance</div>
                        <div class="h5 mb-0 font-weight-bold text-
gray-800">
                            {{wallet["AMOUNT"]-
current_month_expense["1"]}}</div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="col-xl-3 col-md-6 mb-2">
        <div class="card border-left-info shadow h-100 py-
```

```
2">
            <div class="card-body">
              <div class="row no-gutters align-items-center">
                <div class="col mr-2">
                  <div class="text-xs font-weight-bold text-
info text-uppercase mb-1">Expense
                    Percentage
                  </div>
                  <div class="row no-gutters align-items-
center">
                    <div class="col-auto">
                      <div class="h5 mb-0 mr-3 font-weight-
bold text-gray-800">
                        {{100-(((wallet["AMOUNT"]-
current_month_expense["1"])/wallet["AMOUNT"])*100)}}%
                      </div>
                    </div>
                    <div class="col">
                      <div class="progress progress-sm mr-
2">
                        <div class="progress-bar bg-info"
                          role="progressbar" style="width:
45%;"
                          aria-valuenow="{{100-
((wallet['AMOUNT']-
current_month_expense['1'])/wallet['AMOUNT'])*100}}"
                          aria-valuemin="0" aria-
valuemax="100"></div>
```

```
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

</div>
<div class="col-lg mt-3">
    <div class="card shadow mb-4">
        <div class="card-header py-3 d-flex flex-row align-
items-center justify-content-between">
            <h6 class="m-0 font-weight-bold text-
primary">THIS MONTH EXPESES</h6>
        </div>
        <div class="card-body">
            <div class="chart-area">
                <canvas id="myAreaChart"
height="400px"></canvas>
            </div>
        </div>
    </div>
</div>


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

```
      <div class="card shadow mb-4">
        <div class="card-header py-3 d-flex flex-row align-
items-center justify-content-between">
          <h6 class="m-0 font-weight-bold text-
primary">EXPENSES</h6>
        </div>

        <div class="card-body">
          <div class="chart-pie pt-4 pb-2">
            <canvas id="myPieChart" height="500px"
></canvas>
          </div>
        </div>
      </div>
    </div>
    <div class="col-lg">
      <div class="card shadow mb-4">
        <div class="card-header py-3">
          <h6 class="m-0 font-weight-bold text-
primary">THIS YEAR EXPENSES</h6>
        </div>
        <div class="card-body">
          <div class="chart-area">
            <canvas id="myBarChart"
height="500px"></canvas>
          </div>
        </div>
      </div>
```

```
    </div>
</div>
{% endif %}
{% if wallet and expense["1"]==0 %}
<div class="container-fluid">
    <div class="d-sm-flex align-items-center justify-content-
between mb-4">
        <h1 class="h3 mb-0 text-gray-800">Welcome</h1>
    </div>
    <div class=" col-xl-3 col-md-6 card shadow m-4">
        <div class="card-body d-flex flex-row align-items-
center justify-content-center">
            <a href="{{url_for('addExpense')}}">Add
Expense</a>
        </div>
    </div>
</div>
{% endif %}
{% if not wallet %}
<div class="container-fluid">
    <div class="d-sm-flex align-items-center justify-content-
between mb-4">
        <h1 class="h3 mb-0 text-gray-800">Welcome</h1>
    </div>
    <div class="col-xl-3 col-md-6 card shadow m-4 ">
        <div class="card-body d-flex flex-row align-items-
center justify-content-center" style="width:50%;">
            <a href="{{url_for('wallet')}}">Add Wallet</a>
```

```
      </div>
    </div>
</div>
{% endif %}

<div id="label" style="display: none;">
    {% for value in month[0] %}
    <div class="monthlabel">{{value}}</div>
    {% endfor %}
</div>

<div id="value" style="display: none;">
    {% for value in month[1] %}
    <div class="monthvalue">{{value}}</div>
    {% endfor %}
</div>
<div id="label" style="display: none;">
    {% for value in year[0] %}
    <div class="yearlabel">{{value}}</div>
    {% endfor %}
</div>

<div id="value" style="display: none;">
    {% for value in year[1] %}
    <div class="yearvalue">{{value}}</div>
    {% endfor %}
</div>
```

```
<script
src="{{url_for('static',filename='js/areaChart.js')}}"></scrip
t>
<!--Pie chart-->
<script
src="{{url_for('static',filename='js/pieChart.js')}}"></script>
<!--Bar Chart-->
<script
src="{{url_for('static',filename='js/barChart.js')}}"></script>

{% endblock %}
```

app.py

```
def getWallet():
    sql = "SELECT AMOUNT FROM WALLET WHERE
USERID=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1,session['id'])
    ibm_db.execute(stmt)
    data=ibm_db.fetch_assoc(stmt)
    return data

def ExpenseCount():
    sql = "SELECT COUNT(*) FROM EXPENSES WHERE
USERID=?"
```

```
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1,session['id'])
    ibm_db.execute(stmt)
    data = ibm_db.fetch_assoc(stmt)
    return data


def categorylist():

categoryList=['Food','Shopping','Transport','Movie','Healthc
are']
    sql = "SELECT * FROM category WHERE userid=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, session['id'])
    ibm_db.execute(stmt)
    data = ibm_db.fetch_assoc(stmt)
    while data!=False:
        categoryList.append(data["NAME"])
        data=ibm_db.fetch_assoc(stmt)
    return categoryList


def current_month_expense_amount():
    sql="SELECT SUM(AMOUNT) FROM EXPENSES WHERE
MONTH(DATEOFEXPENSE)=? AND
YEAR(DATEOFEXPENSE)=? AND USERID=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1,currentMonth.month)
```

```
    ibm_db.bind_param(stmt, 2,currentMonth.year)
    ibm_db.bind_param(stmt, 3,session['id'])
    ibm_db.execute(stmt)
    data = ibm_db.fetch_assoc(stmt)
    return data


def GetCatergoryMonth():
    Category=[]
    sql = "SELECT CATEGORY FROM EXPENSES WHERE
MONTH(DATEOFEXPENSE)=? AND  USERID=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, currentMonth.month)
    ibm_db.bind_param(stmt, 2, session['id'])
    ibm_db.execute(stmt)
    data = ibm_db.fetch_assoc(stmt)
    while data!=False:
        Category.append(data["CATEGORY"])
        data=ibm_db.fetch_assoc(stmt)
    return Category

def GetCatergoryYear():
    Category=[]
    sql = "SELECT CATEGORY FROM EXPENSES WHERE
userid=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, session['id'])
    ibm_db.execute(stmt)
```

```
    data = ibm_db.fetch_assoc(stmt)
    while data!=False:
        Category.append(data["CATEGORY"])
        data=ibm_db.fetch_assoc(stmt)
    print(Category)
    return Category



def monthExpense():
    category=GetCatergoryMonth()
    finalData=[]
    temp1=[]
    temp2=[]
    for value in category:

        sql="SELECT SUM(AMOUNT) FROM EXPENSES
WHERE CATEGORY=? AND MONTH(DATEOFEXPENSE)=?
AND YEAR(DATEOFEXPENSE)=?  AND USERID=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1,value)
        ibm_db.bind_param(stmt, 2,currentMonth.month)
        ibm_db.bind_param(stmt, 3,currentMonth.year)
        ibm_db.bind_param(stmt, 4,session['id'])
        ibm_db.execute(stmt)
        data = ibm_db.fetch_assoc(stmt)

        if data["1"]!=None:
```

```
            temp1.append(value)
            temp2.append(data["1"])

    finalData.append(temp1)
    finalData.append(temp2)
    print(finalData)
    return finalData


def yearExpense():
    category=GetCatergoryYear()
    finalData=[]
    temp1=[]
    temp2=[]
    for value in category:

        sql="SELECT SUM(AMOUNT) FROM EXPENSES
WHERE CATEGORY=? AND  YEAR(DATEOFEXPENSE)=?
AND USERID=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1,value)
        ibm_db.bind_param(stmt, 2,currentMonth.year)
        ibm_db.bind_param(stmt, 3,session['id'])
        ibm_db.execute(stmt)
        data = ibm_db.fetch_assoc(stmt)
        if data["1"]!=None:
            temp1.append(value)
            temp2.append(data["1"])
```

```
    finalData.append(temp1)
    finalData.append(temp2)
    print(finalData)
    return finalData

@app.route('/dashboard')
def dashboard():
    monthExp=monthExpense()
    yearExp=yearExpense()

current_month_expense=current_month_expense_amount
()
    wallet=getWallet()
    expense=ExpenseCount()



    return
render_template('dashboard.html',wallet=wallet,expense=e
xpense,

current_month_expense=current_month_expense,month=
monthExp,year=yearExp



    )
```

Database Schema

Adding User

```
CREATE TABLE USER
(
        id      INT NOT NULL GENERATED ALWAYS AS IDENTITY
                                (START WITH 1, INCREMENT BY 1),

        name          varchar(50) NOT NULL,
        email    varchar(100) NOT NULL unique,
        password varchar(50) NOT NULL

);
```

Adding Expense

```
CREATE TABLE EXPENSES

(
        id      INT NOT NULL GENERATED ALWAYS AS IDENTITY
                                (START WITH 1, INCREMENT BY 1),

    userid int not null,
    amount INT NOT NULL,
    category varchar(30) NOT NULL,
    DateofExpense date

);
```

Adding Category

CREATE TABLE CATEGORY

(

id INT NOT NULL GENERATED ALWAYS AS IDENTITY
(START WITH 1, INCREMENT BY 1),

userid int not null,
name varchar(20),
description varchar(40)

);

8. TESTING

1.TESTING

Test case ID	Purpose	Test Case	Result
TC1	AUTHENTICATION	Password with length less than 8 characters	Password cannot be less than 8 character
TC2	AUTHENTICATION	Email is already exits	Email should be unique

TC3	AUTHENTICATION	Email and password is valid	Login successfully
TC4	AUTHENTICATION	Password and confirm password didn't match	Please enter the password

2.User Acceptance Testing

Technical Requirment Document (TSD)	
Test Case ID	**Test Case Description**
TC_001	Verify if user is able to order single product.
TC_002	Verify if user is able to order multiple products.
TC_003	Verify if user can apply single or multiple filters
TC_004	Verify if user can apply different sort by
TC_005	Verify if user is able to pay by Master Card
TC_006	Verify if user is able to pay by Debit Card
TC_007	Verify if user is able to pay fully by reward points
TC_008	Verify if user is able to pay partially by reward points

9. RESULTS

Performance Metrics

- Tracking expenses: Monitoring the income and tracking all expenditures
- Tracking Receipts : Capture and organize your payment receipts to keep track of your expenditure
- Category: Add expenses based on your type of category
- 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.

10. 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 solving them a lot 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 interfaces.
- Provide backup and recovery of data.
- Provide a better user interface for users.
- Mobile apps advantage.

Project link:

https://github.com/IBM-EPBL/IBM-Project-25860-1659975210

Video link:

https://drive.google.com/file/d/10h7ZzcumYekWT7d5LoMeSWI-Gm6hOYC2/view?usp=sharing