PROJECT REPORT

PERSONAL EXPENSE TRACKER APPLICATION

TEAM ID: PNT2022TMID39972

TEAM LEAD: RUTHYA T

TEAM MEMBER 1: SUBALAKSHMI G

TEAM MEMBER 2: SNEHA LATHA P

TEAM MEMBER 3: MAYURI SRI H

11. FUTURE SCOPE

12. APPENDIX

Source Code

GitHub & Project Demo Link

1.INTRODUCTION

Control of personal budget has become the most important factor for the good welfare of people. The reasons are the economic crisis, which prompted people to keep track of expenses and income. Accounting of personal finances will help to save money and make only necessary purchases, will help to analyse your expenses, correctly plan your budget and rationally use money. Due to the rapid development of technologies, accounting of expenses can be carried out from the phone and be so simple and convenient, taking a minimum of time and giving the maximum result. In this regard, in our time, it becomes more relevant than ever to develop an application for expense accounting.

1.1.PROJECT OVERVIEW

The Expense Manager is a multi-purpose finance related android application intended to run on android devices. The android application can be run on all android devices above android version 5.0. It is designed efficiently to give you the best suggestions for finance planning. The application size is less than 10 MB. It doesn't need any high end hardware specification. It can easily run on low end devices. The features of the app are designed in a way to help you for better finance management planning so that you can keep track of , analyse and optimize your budget or spending's. In this application we are also going to collect user's data with authenticated permissions and analyse and study their pattern expenses in certain category or by distinct kinds of spending that can be used for studying market trends. These analysis patterns can be derived using some data mining techniques such as clustering, classification and association.

1.2.PURPOSE

The purpose of the application is to track the daily expenses and to check the expenses of the purchase or the money spent for the purpose.

And to send the emails to the individuals that he/she is extending their limit of savings.That is the purpose of the application to track the expenses by uploading the images of the bills and by the text recognition algorithm the process is undertaken.

2. LITERATURE SURVEY :

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 investm

2.1 EXISTING PROBLEM

The existing methods of check recognition were analyzed. At the moment, there are a lot of different ways to scan a receipt. Some of them are: scanning the check by using a QR code, which contains the unique information of the receipt: information about the products, price, etc.

In this project particularly text recognition algorithm is used that is tesscrt OCR algorithm and by using python language the process of the system is done.

2.2. REFERENCES:

https://ieeexplore.ieee.org/document/8628070

[1] Pekerman I. L. Sistemy raspoznavaniya magazinnyh chekov. NAUKA I SOVREMENNOST': sbornik statej Mezhdunarodnoj nauchnoprakticheskoj konferencii [Store receipt recognition systems. SCIENCE AND MODERNITY: a collection of articles of the International Scientific and Practical Conference], Ufa, 2014, pp.44-45 (In Russian)

[2] Pekerman I. L. Raspoznavanie teksta chekov pri pomoshchi nejronnyh. NAUKA I SOVREMENNOST': sbornik statej Mezhdunarodnoj nauchno-prakticheskoj konferencii [Neural text recognition of receipts. SCIENCE AND MODERNITY: a collection of articles of the International Scientific and Practical Conference], Ufa, 2015, pp.33-36 (In Russian)

[3] Gladun D. P. Raspoznavanie tekstovyh dannyh izobrazhenij kassovyh chekov. Sovremennye problemy matematiki i vychislitel'noj tekhniki : sbornik materialov X Respublikanskoj nauchnoj konferencii molodyh uchenyh i studentov [Recognition of text data of cash voucher images. Modern problems of mathematics and computer engineering : a collection of materials of the X Republican Scientific Conference of Young Scientists and Students], Brest, 2017, pp. 38-41

[4] Kravchenko A. S. Razrabotka mobil'nogo prilozheniya - schityvanie

chekov i cennikov. Mezhvuzovskaya nauchno-tekhnicheskaya konferenciya studentov, aspirantov i molodyh specialistov im. E.V. Armenskogo. [Development of a mobile application - reading of checks and price lists. Inter-University Scientific and Technical Conference of Students, Postgraduate Students and Young Professionals named after E.V. Armensky], Moscow, 2016, pp. 74-76

[5] A.S. Nazdryukhin, I.N. Khramtsov, A.N. Tushev. Processing images of sales receipts for isolating and recognising text information. Herald of Daghestan State Technical University. Technical Sciences. 2019; 46 (4): 113-122. (In Russ.) DOI:10.21822/2073-6185-2019-46-4-113-122 [6] Rafi Ullah, Ali Sohani, Faraz Ali, Athaul Rai. OCR Engine to extract Food-items and Prices from Receipt Images via Pattern matching and heuristics approach. 2017, 23 p.

[6]. Homma H. Accreditation System in Indonesia. JSME news. 2004 May; 15(1) [7]. Oberst B, Jones R. International Trends in Engineering Accreditation and Quality Assurance. World Expertise L.L.C. [8]. Google, 2015. Google developers. [Online] Available at: https://developers.google.com/maps/docum entation/android/intro#accessibility

2.3.PROBLEM STATEMENT

After discussing our application functions and comparing them to other existing applications, some features were found lacking. This is a new application that will attract the public user through its features

There are always some challenges. We have to face some challenges as well, since the main

Purpose of our application is to track the user's expenses.

This is an Android-based mobile application, so if a user does not have an Android phone then this application will not help him.

After getting notifications if a user doesn check his phone for full information then the main motto of this app will fail.

3. IDEATION & PROPOSED SOLUTION:

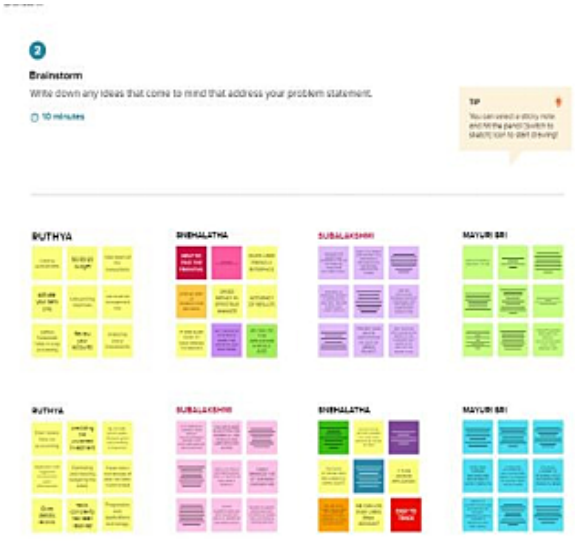
3.1 EMPATHY MAP :

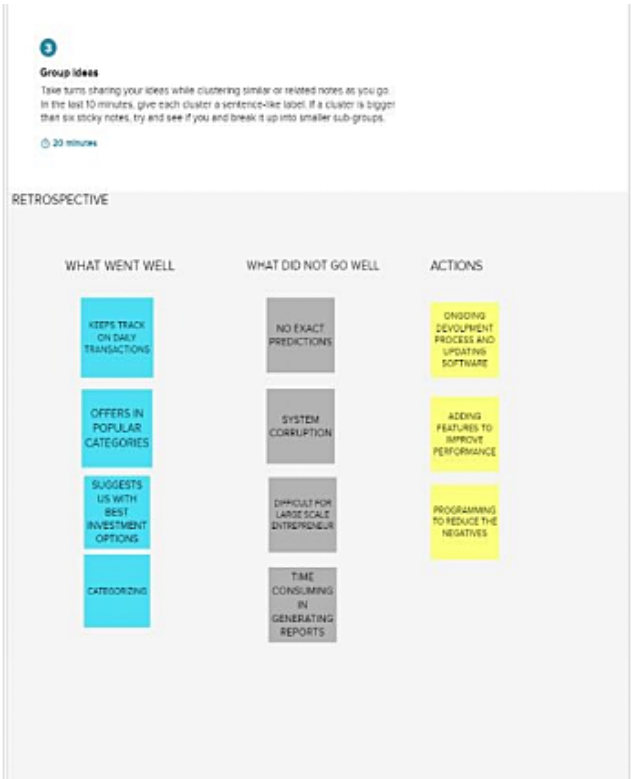
An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user personal, an empathy map can represent a group of users, such as customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.

3.2 BRAIN STORM:

Brainstorming is a group creativity technique by which efforts are made to find a conclusion for a specific problem by gathering by list of ideas spontaneously contributed by its members.

- Pick an appropriate facilitator.
- Set the agenda.
- Holding the session.





3.3. PROPOSED SOLUTION

The Tesseract is one of the best OCR systems and it supports Russian language recognition. Moreover, it provides the possibility of recognition of multiple languages together.
The process of text recognition from a receipt consists of the following:

The image received from the phone camera is pre-processed, then it is recognized.

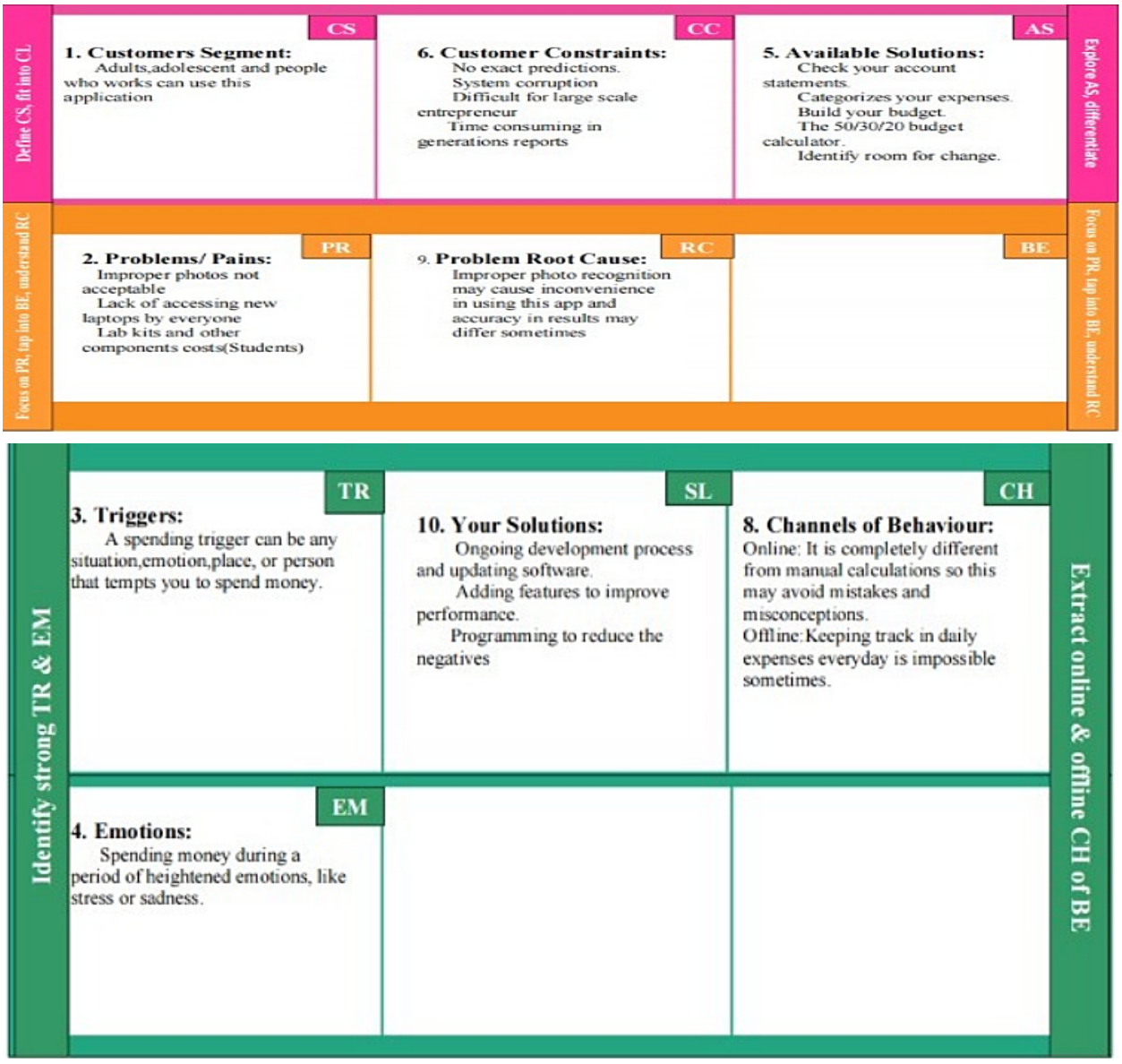
The image is pre-processed using the OpenCV library, the algorithm is written in Python.

By the process, the further procedure is taken and the calculation of expenses is done.

The accuracy of 90% of expenses is calculated.

3.4 PROBLEM SOLUTION FIT

The requirements and yhe needs of the user is defined in the templates.



4. FUNCTIONAL REQURIEMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirements	Sub Registration
FR-1	Login	Login can be done using mobile number or user name and password.
FR-2	Authentication	Login access can be verified by email the user will notify by an email.
FR-3	Reliable servers	private database and Reliable and efficient sever was used
FR-4	Access	Scanning of bill and cheque by image selection. Personal calculation can be done.

4.1 NON- FUNCTIONAL REQURIEMENTS

Following are the Non-Functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	A user-friendly interface tomake use simple and efficient
NFR-2	Security	Log in credentials ensures the user data, calculation of expenses are known only by the user.
NFR-3	Reliability	It has a fault-tolerance infrastructure it can be able to provide a reliable service
NFR-4	Performance	Text recognition, tesseract OCR where used to identify the images and text of the images.therefore it gives accurate expenses of the user and it is more secure so the application is very responsive
NFR-5	Availability	It is a cloud based web application so user can access with out any platform limitations ,just using a browsers with a internet connection is enough for use the application
NFR-6	Scalability	It has a high throughput, enough network resources and soon and quick calculation of expenses and it gives the difference.

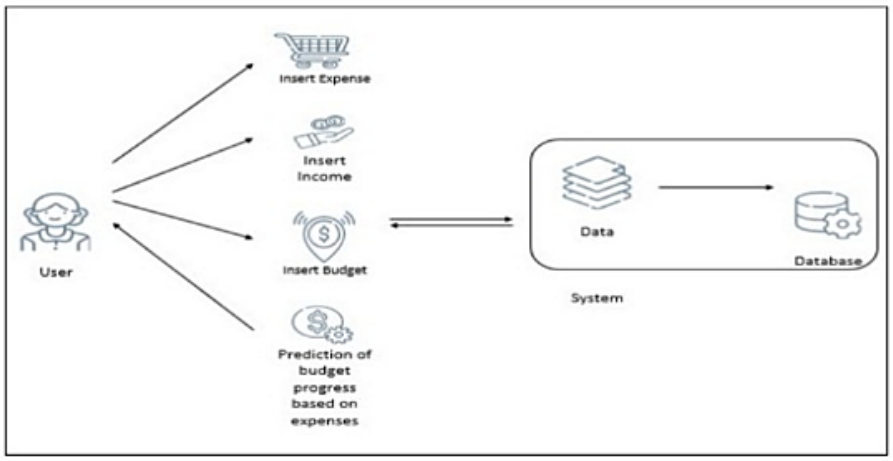
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

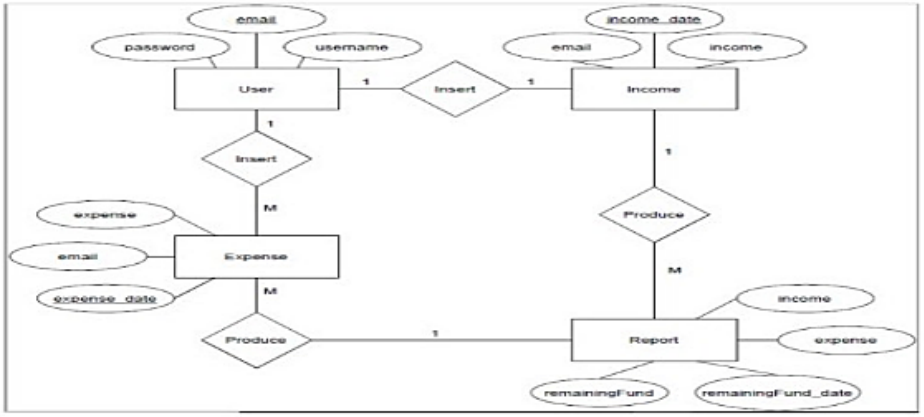
It's easy to understand the flow of data through systems with the right data flow diagram software. This guide provides everything you need to know about data flow diagrams, including definitions, history, and symbols and notations. You'll learn the different levels of a DFD, the difference between a logical and a physical DFD and tips for making a DFD.

Data Flow Diagrams:

DFD (level 0):



DFD (Level 2):



5.2 User Stories

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

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)	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	Sprint-1
		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	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can use my gmail credentials instead of manually entering my details	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can retrieve my information	High	Sprint-1
	Dashboard	USN-5	As a user, I can continue my old progress	I can continue where I left	Medium	Sprint-1
Customer (Laptop user)	Registration	USN-1	I can register for the application using browser by entering the email, password, name and personal information and conforming my password	I can access my profile	High	Sprint-1
		USN-2	As a user, I will receive a confirmation and greeting mail from the web application	I can receive the confirmation and greeting mail	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can use my gmail credentials instead of manually entering my details	Medium	Sprint-1
	login	USN-4	As a user, I can log into the application by entering email & password	I can retrieve my profile	High	Sprint-1
	Dashboard	USN-5	As a user, I can view my personalized dashboard	I can continue where I left	medium	Sprint-2
	Chatbot	USN-6	As a user, I expect a bot to assist me	I can interact with the bot	High	
Administration	Login	USN-1	As a admin, login on admin page	I can access admin menu	High	Sprint-2
		USN-2	As a admin, access the admin menu using admin credentials	I can log in using admin credentials	High	Sprint-2

5.3 Solution Architecture

Solution architecture is a complex process – with many sub-processes that bridgesthe gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of thesoftware to project stakeholders.
- Define features, development phases, and solution requirements.

- Provide specifications according to which the solution is defined, managed,and delivered.

Technical Architecture Diagram:-

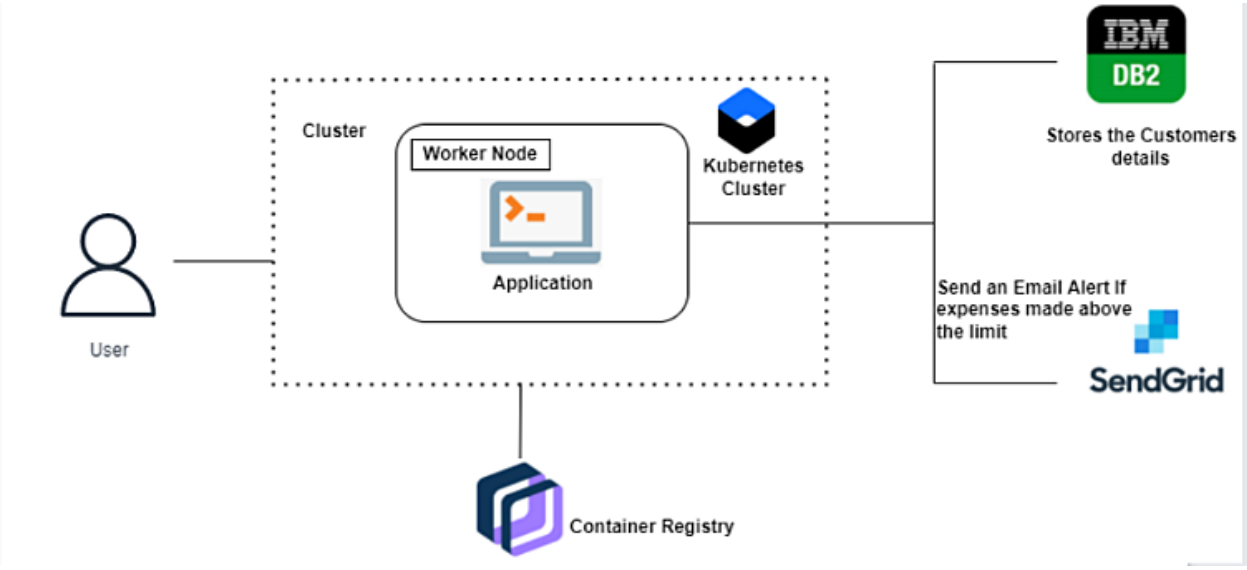


Figure 5.3.1 Technical Architecture Diagram

Solution Architecture Diagram:-

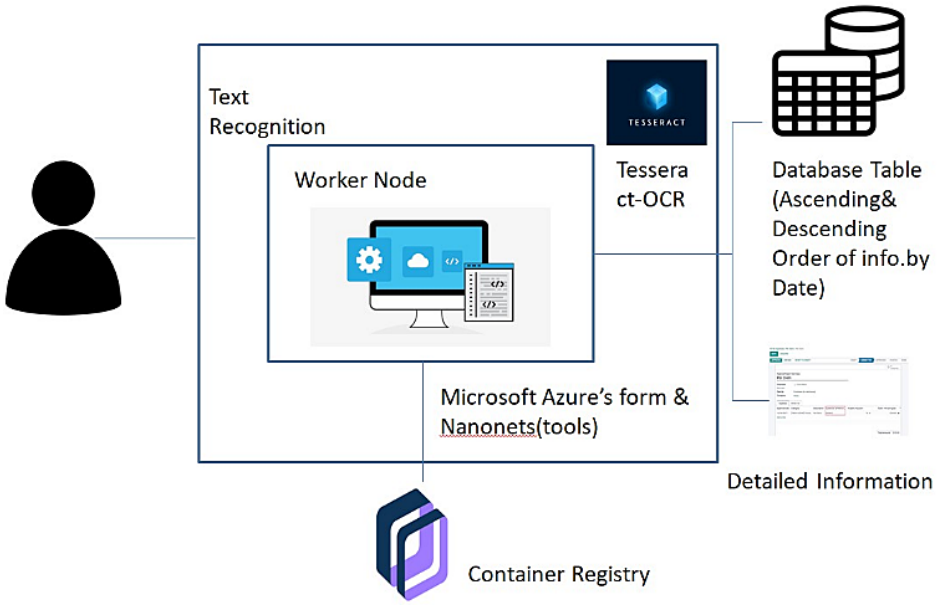


Figure 5.3.2 Solution Architecture Diagram

Detailed solution Architecture Diagram:-

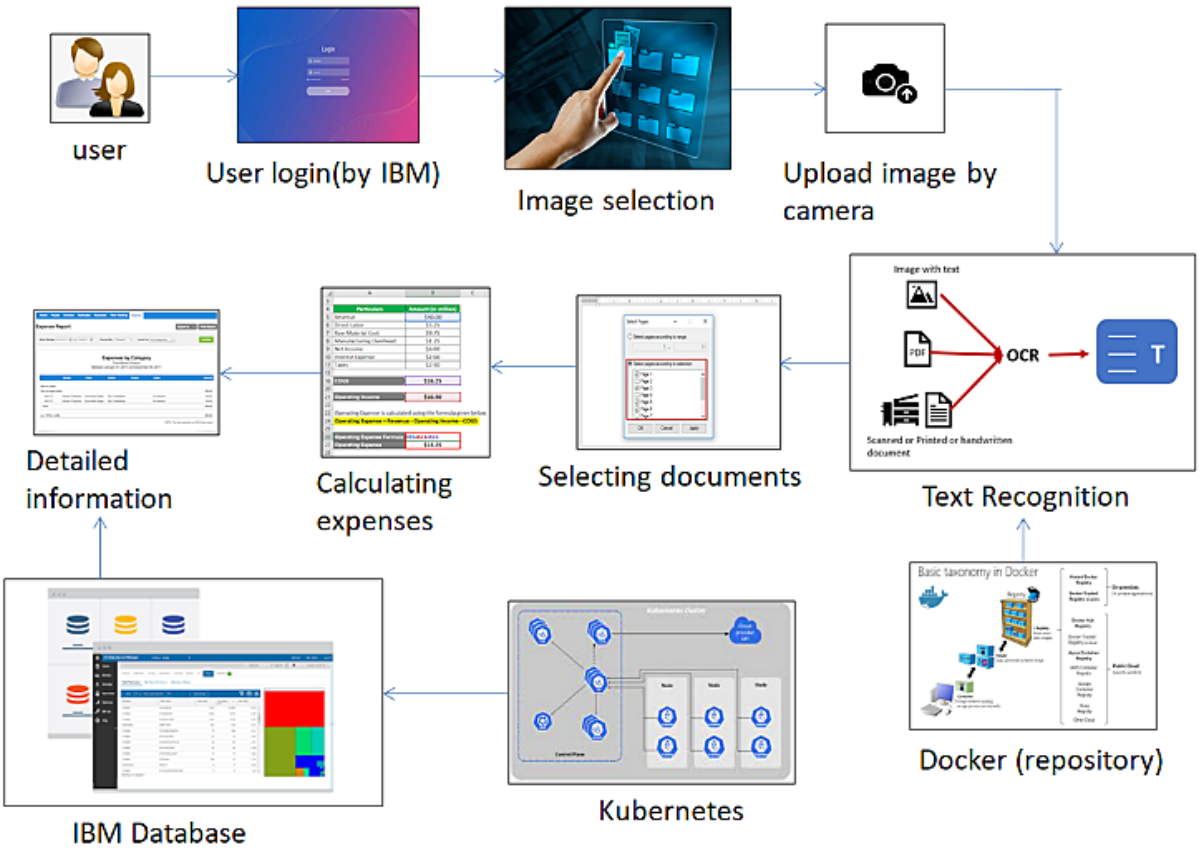


Fig 5.3.3 Detailed Solution Architecture explaining the expense tracking.

6.PROJECT PLANNING

6.1 SPRINT PLANNING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Panel	USN-1	The user will login into the website and go through the details which they have to respond.	20	High	Ruthya.T Subalakshmi.G Snehalatha.P Mayuri sri.H
Sprint-2	Admin panel	USN-2	The role of the admin is to check out the database about the details given by the user and have a track of all the things that the user access.	20	High	Ruthya.T Subalakshmi.G Snehalatha.P Mayuri sri.H
Sprint-3	Chat Bot	USN-3	The user can directly talk to Chat bot regarding the calculation. Get the recommendations based on information provided by the user.	20	High	Ruthya.T Subalakshmi.G Snehalatha.P Mayuri sri.H
Sprint-4	final delivery	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	20	High	Ruthya.T Subalakshmi.G Snehalatha.P Mayuri sri.H

6.2 SPRINT DELIVERY

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022		29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		19 Nov 2022

Velocity:

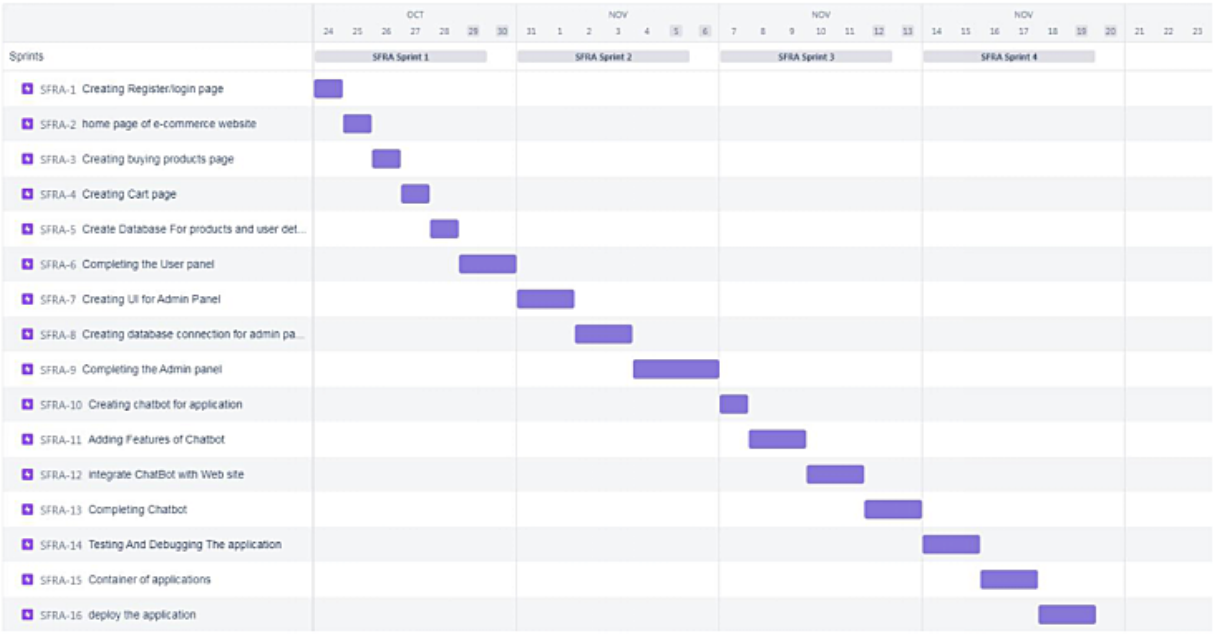
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) periteration unit (story points per day)

Average velocity= 20/6 = 3.33

BURNDOWN CHART:-

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies suchas Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

Burndown Chart:



7.CODING AND SOLUTION

app.py

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

import ibm_db

import sendgrid

 import os

from dotenv import load_dotenv

from sendgrid.helpers.mail import Mail, Email, To, Content

app = Flask(__name__)

# secret key required to maintain unique user sessions

app.secret_key =
'f39c244d6c896864abe3310b839091799fed56007a438d637baf526007609fe0'

# establish connection with IBM Db2 Database

connection = ibm_db.connect("DATABASE=bludb;HOSTNAME=8e359033-a1c9-4643- 82ef

8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30120;SECU
RITY=SSL;UID=gyq42313;PWD=TRn8xYTPTGloNwQC;", "", "")

load_dotenv()  # load keys from .env

sg = sendgrid.SendGridAPIClient(api_key=os.environ.get(     'SENDGRID_API_KEY'))

# set SendGrid API Key

# the address that sends emails to the users

from_email = Email("dhanushcodepro@gmail.com")

# Handle expense model according to ibm db

@app.route('/')

 @app.route('/dashboard')

def dashboard():
```

```
if 'username' not in session:
 # ask user to sign in if not done already
 return redirect(url_for('signin'))
# Fetch the list of expenses from db
sql = 'select * from expenses where cid = '+str(session['id'])
stmt = ibm_db.exec_immediate(connection, sql)
flag=0
expense_list = {}
 i=0
while (res:=ibm_db.fetch_assoc(stmt)) != False:
 expense_list[i] = res
flag=1
i+=1
# go to homepage if signed in
sql='select sum(eamount) from expenses where cid= '+str(session['id'])
stmt = ibm_db.exec_immediate(connection, sql)
sum_dict= ibm_db.fetch_assoc(stmt)
sum=0.0       sum = sum_dict['1']
if flag !=1:
 sum=0.0
sql='select budget from users where id= '+str(session['id'])
  stmt = ibm_db.exec_immediate(connection, sql)
budget_list= ibm_db.fetch_assoc(stmt)
# budget = budget_list.values()
 for key,value in budget_list.items():
```

```
pass
rem = value - sum
 per=0
 if value !=0:
    per = (sum/value)*100
return render_template('dashboard.html',
ex_list=expense_list,esum=sum,budget=value,rem=rem,per=per,name=session['usernam e'])
 @app.route('/addexpense')
def add():
        return render_template('addexpense.html')
@app.route('/addbudget')
def addb():
sql='select budget from users where id= '+str(session['id'])
stmt = ibm_db.exec_immediate(connection, sql)
budget_list= ibm_db.fetch_assoc(stmt)
 # budget = budget_list.values()
 for key,value in budget_list.items():
 pass
 return render_template('addbudget.html',budget=value,id=session['id'])
@app.route('/addbudget/',methods=['POST'])
def addbudget(id):
budget = request.form['budget']
sql = 'update users set budget = ? where id = '+str(id)
# Add to the database here
pstmt = ibm_db.prepare(connection, sql)
```

```
ibm_db.bind_param(pstmt, 1, budget)
ibm_db.execute(pstmt)
return redirect('/dashboard')
@app.route('/addexpense', methods=['POST'])
def addexpense():
date = request.form['date']
expensename = request.form['expensename']
amount = request.form['amount']
category = request.form['category']
sql = 'INSERT INTO expenses(edate,ename,eamount,ecategory,cid) VALUES(?,?,?,?,?)'
# Add to the database here
pstmt = ibm_db.prepare(connection, sql)
ibm_db.bind_param(pstmt, 1, date)
ibm_db.bind_param(pstmt, 2, expensename)
ibm_db.bind_param(pstmt, 3, amount)
ibm_db.bind_param(pstmt, 4, category)
ibm_db.bind_param(pstmt, 5, session['id'])
ibm_db.execute(pstmt)
flash('Expense added Successfully')
return redirect('/dashboard')
@app.route('/expense/update/')
def update(id):
# Get from the database
sql = 'select * from expenses where id = '+str(id)
# Add to the database here
```

```
pstmt = ibm_db.prepare(connection, sql)

ibm_db.execute(pstmt)

acc = ibm_db.fetch_assoc(pstmt)

return render_template('updateexpense.html',acc=acc)

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

def edit():

    id = request.form["id"]

    date = request.form['date']

    expensename = request.form['expensename']

    amount = request.form['amount']

    category = request.form['category']

   sql = 'update expenses set edate = ?,ename = ? ,eamount = ? ,ecategory = ? where id = '+str(id)
# Add to the database here

    pstmt = ibm_db.prepare(connection, sql)

    ibm_db.bind_param(pstmt, 1, date)

    ibm_db.bind_param(pstmt, 2, expensename)

    ibm_db.bind_param(pstmt, 3, amount)

    ibm_db.bind_param(pstmt, 4, category)

    ibm_db.execute(pstmt)

    return redirect('/dashboard')

@app.route('/expense/delete/', methods=['GET'])

def delete(id):

    # Database operation

    # flash(str(id))

    sql = 'delete from expenses where id = '+str(id)   #check if user is already registered
```

```
    pstmt = ibm_db.prepare(connection, sql)
    ibm_db.execute(pstmt)
    return redirect('/dashboard')
@app.route('/graph')
def graph():
        sql = 'select * from expenses where cid = '+str(session['id'])
       stmt = ibm_db.exec_immediate(connection, sql)
      expense_list = {}
      i=0
     while (res:=ibm_db.fetch_assoc(stmt)) != False:
            expense_list[i] = res
            i+=1
       total = 0
       household = 0
       food = 0
       entertainment = 0
       business = 0
       other = 0
       for key,value in expense_list.items():
            total += value['EAMOUNT']
             if value['ECATEGORY'] == 'household':
                household += value['EAMOUNT']
              elif value['ECATEGORY'] == 'food':
                food += value['EAMOUNT']
              elif value['ECATEGORY'] == 'entertainment':
```

```
            entertainment += value['EAMOUNT']
        elif value['ECATEGORY'] == 'business':
            business += value['EAMOUNT']
        elif value['ECATEGORY'] == 'other':
            other += value['EAMOUNT']
return render_template('graph.html',  total=total, household=household, food=food,
entertainment=entertainment, business=business, other=other) @app.route('/signout')
def signout():
session.pop('username', None)
# remove user session upon signing out
return redirect('/')
@app.route('/signup')
def register():
if 'username' in session:
#inform user if they're already signed in the same session
flash('You are already signed in! Sign out to login with a different account')
 return redirect(url_for('dashboard'))
else:
return render_template('signup.html')
#take user to the registration page
@app.route('/signup', methods=['POST'])
def regform():
uname = request.form['uname']
#get user id and password from the form
 email = request.form['email']
```

```
 pwd = request.form['pass']
print(uname,email,pwd)
sql = 'SELECT * from users WHERE email=?'
#check if user is already registered
pstmt = ibm_db.prepare(connection, sql)
ibm_db.bind_param(pstmt, 1, email)
ibm_db.execute(pstmt)
acc = ibm_db.fetch_assoc(pstmt)
if acc:
#inform user to sign in if they have an existing account
flash('You are already a member. Please sign in using your registered credentials')
else:
sql = 'INSERT INTO users(username,password,email) VALUES(?,?,?)'
#insert credentials of new user to the database
pstmt = ibm_db.prepare(connection, sql)
ibm_db.bind_param(pstmt, 1, uname)
ibm_db.bind_param(pstmt, 2, pwd)
ibm_db.bind_param(pstmt, 3, email)
ibm_db.execute(pstmt)
flash('Registration Successful! Sign in using the registered credentials to continue')
return redirect(url_for('signin'))
#ask users to sign in after registration @app.route('/signin')
def signin():
if 'username' in session:
# inform user if they're already signed in the same session
```

```
flash('You are already signed in! Sign out to login with a different account')
return redirect(url_for('dashboard'))
return render_template('login.html')
# take user to the sign in page
@app.route('/signinform', methods=['POST'])
def signinform():
uid = request.form['email']
# get user id and password from the form
pwd = request.form['pass']
# check user credentials in the database
sql = 'SELECT * from users WHERE email=? AND password=?'
pstmt = ibm_db.prepare(connection, sql)
ibm_db.bind_param(pstmt, 1, uid)
ibm_db.bind_param(pstmt, 2, pwd)
ibm_db.execute(pstmt)
acc = ibm_db.fetch_assoc(pstmt)
if acc: #if the user is already registered to the application
session['username'] = acc['USERNAME']
session['id'] = acc['ID']
flash(session['username'] + str(session['id'])+'Signed in successfully!')
return redirect(url_for('dashboard'))
else:
#warn upon entering incorrect credentials
flash('Incorrect credentials. Please try again!')
return render_template('login.html')
```

```
if __name__ == '__main__':
app.run(debug=True)
```

addbudget.html

```
{% extends 'base.html' %}
{% block body %}
 <div class="container">
 <div class="row">
     <div class="col-md-8">
        <h3 class="mt-5" style="margin-left: 200px;">Add Budget :</h3>
        <form method='post' action='/addbudget/{{id}}' style="margin-left: 200px;">
          <div class="form-group">
            <label>Amount</label>
          <input class="form-control" type="text" name="budget"
id="budget"
value="{{budget}}">
                </div>
                <input class="btn btn-danger" type="submit"
value="Add">
            </form>
        </div>
    </div>
</div>
{% endblock %}
```

addexpense.html

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

```
<div class="container">

    <div class="row">

        <div class="col-md-8">

            <h3 class="mt-5" style="margin-left: 200px;">Add expense
:</h3>

            <form method='post' action='/addexpense' style="margin-
left: 200px;">

                <div class="form-group">

                    <label>Date of expense</label>

                    <input class="form-control" type="date"
name="date" id="date">

                </div>

                <div class="form-group">

                    <label>Expense name</label>

                    <input class="form-control" type="text"
name="expensename"

id="expensename">

                </div>

                <div class="form-group">

                    <label>Expense amount</label>

                    <input class="form-control" type="text"
name="amount" id="amount">

                </div>

                <div class="form-group">

                    <label>Category</label>

                    <select class="form-control" name="category"
id="category">
```

```
                        <option value="household">Household</option>
                        <option value="food">Food</option>
                        <option
value="entertainment">Entertainment</option>
                        <option value="business">Business</option>
                        <option value="other">Other</option>
                    </select>
                </div>
                <input class="btn btn-danger" type="submit"
value="Add">
            </form>
        </div>
    </div>
</div>
{% endblock %}
```

base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    {% block head %} {% endblock %}
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font▫awesome/6.2.0/css/al
l.min.css">
```

```
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.
min.css"

rel="stylesheet"

        integrity="sha384-

EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"

crossorigin="anonymous">

    <script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bu
ndle.min.js"

        integrity="sha384-

MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"

        crossorigin="anonymous"></script>

    <script
src="https://cdn.jsdelivr.net/npm/chart.js@3.2.0/dist/chart.min.js"></
script>

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

    <link

href="https://fonts.googleapis.com/css2?family=Roboto:wght@700&display
=swap"

rel="stylesheet">

    <link rel="stylesheet" href="{{url_for('static',
filename='custom.css')}}">

</head>

<body>

    <nav class="navbar navbar-expand-lg sticky-top navbar-light bg-
light">

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

```
            <a class="navbar-brand fw-bold" href="/">

                Expense Tracker

            </a>

            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-

target="#navbarNav"

                aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle

navigation">

                <span class="navbar-toggler-icon"></span>

            </button>

            <div class="collapse navbar-collapse" id="navbarNav">

                <ul class="navbar-nav">

                    <li class="nav-item">

                        <a class="nav-link" href="/"
href="#">Dashboard</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/addbudget">Set
Budget <span class="sr-only">(current)</span></a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/addexpense">Add
Expenses <span class="sr-only">(current)</span></a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/graph">Graph</a>
```

```
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/signout">
                            <i class="fa-solid fa-right-from-
bracket"></i>
                            Sign Out
                        </a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    {% with messages = get_flashed_messages() %}
    {% if messages %}
    {% for message in messages %}
    <script>alert('{{ message }}')</script>
    {% endfor %}
    {% endif %}
    {% endwith %}
    {% block body %} {% endblock %}
</body>
</html>
```

dashboard.html

{% extends 'base.html' %}

```
{% block head %}
<title>Expense Tracker - Dashboard</title>
{% endblock %}
{% block body %}
<div class="container" >
    <h3 class="mt-5">Hello {{name}} , Your Expenses</h3>
    <div class="d-flex justify-content-between">
        <h3 class="mt-5">Budget = {{budget}} </h3>
        <h3 class="mt-5">Total spent = {{esum}}</h3>
        <h3 class="mt-5">Remaining = {{rem}}</h3>
    </div>
    {% for key,value in ex_list.items() %}
    <div class="row" >
        <div class="col-md-12">
            <div class="card shadow mb-2 bg-white rounded">
                <div class="card-body" style="background: linear-
gradient(#FFEBEE,#FAFAFA)">
                    <div class="row">
```

```
                    <div class="col-md-2">
                        <a href="expense/update/{{value['ID']}}" class="btn
btn-sm btn   success">Edit</a>
                    </div>
                    <div class="col-md-2">
                        <a href="expense/delete/{{value['ID']}}" class="btn
btn-sm btn   danger">Delete</a>
                    </div>
                    <div class="col-md-2" style="color: #000000">
                        {{value['EDATE']}}
                    </div>
                    <div class="col-md-2"style="color: #000000">
                        {{value['ECATEGORY']}}
                    </div>
                    <div class="col-md-2"style="color: #000000">
                        {{value['ENAME']}}
                    </div>
                    <div class="col-md-2"style="color: #000000">
                        ₹ {{value['EAMOUNT']}}
```

```
                    </div>
                </div>
            </div>
        </div>
    </div>
    {% endfor %}
    <div class="d-flex justify-content-between">
        <h4 class="mt-5">Spent % = {{per}} %    </h3>
        <h4 class="mt-5">Total expense = {{esum}}</h3>
    </div>
</div>
</body>
{% endblock %}
</html>
```

graph.html

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

```
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <h3 class="mt-5">EXPENSE BREAKDOWN</h3><br>
            <div class="card shadow bb-2 bg-dark rounded">
                <div class="card-body">
                    <div class="row">
                        <div class="col-md-6" style="color:
aliceblue;"><b>CATEGORY</b></div>
                        <div class="col-md-6" style="color: aliceblue;">TOTAL
EXPENSE</div>
                    </div>
                </div>
            </div>
            <div class="card shadow bb-2 bg-white rounded">
                <div class="card-body" style="color: #000000">
                    <div class="row">
                        <div class="col-md-6">Household</div>
                        <div id="thousehold" class="col-md-
```

```
6">{{household}}</div>
                </div>
            </div>
        </div>
        <div class="card shadow bb-2 bg-white rounded">
            <div class="card-body" style="color: #000000">
                <div class="row">
                    <div class="col-md-6">Food</div>
                    <div id="tfood" class="col-md-6">{{food}}</div>
                </div>
            </div>
        </div>
        <div class="card shadow bb-2 bg-white rounded">
            <div class="card-body" style="color: #000000">
                <div class="row">
                    <div class="col-md-6">Entertainment</div>
                    <div id="tentertainment" class="col-md-
6">{{entertainment}}</div>
                </div>
```

```
            </div>
        </div>
        <div class="card shadow bb-2 bg-white rounded">
            <div class="card-body" style="color: #000000">
                <div class="row">
                    <div class="col-md-6">Business</div>
                    <div id="tbusiness" class="col-md-
6">{{business}}</div>
                </div>
            </div>
        </div>
        <div class="card shadow bb-2 bg-white rounded">
            <div class="card-body" style="color: #000000">
                <div class="row">
                    <div class="col-md-6">Other</div>
                    <div id="tother" class="col-md-6">{{other}}</div>
                </div>
            </div>
```

```
        </div>
        <div class="card shadow bb-2 bg-white rounded">
            <div class="card-body" style="color: #000000">
                <div class="row">
                    <div class="col-md-6">TOTAL</div>
                    <div class="col-md-6">{{total}}</div>
                </div>
            </div>
        </div>
    </div>
    <div class="col-md-6">
        <h3 class="mt-5">EXPENSE CHART</h3>
        <canvas id="myChart" width="400" height="400"></canvas>
        <script>
            let household =
document.getElementById('thousehold').innerHTML
            let food = document.getElementById('tfood').innerHTML
            let entertainment =
document.getElementById('tentertainment').innerHTML
```

```
        let business =
document.getElementById('tbusiness').innerHTML
        let other = document.getElementById('tother').innerHTML
        var ctx =
document.getElementById('myChart').getContext('2d');
        var myChart = new Chart(ctx, {
            type: 'pie',
            data: {
                labels: ['Household','Food', 'Entertainment', 'Business',
'Other'],
                datasets: [{
                    label: 'Expense amount',
                    data: [household,food,entertainment,business,other],
                    backgroundColor: [
                        'rgba(255, 99, 132, 0.2)',
                        'rgba(54, 162, 235, 0.2)',
                        'rgba(255, 206, 86, 0.2)',
                        'rgba(75, 192, 192, 0.2)',
                        'rgba(153, 102, 255, 0.2)',
```

```
            'rgba(255, 159, 64, 0.2)'
        ],
        borderColor: [
            'rgba(255, 99, 132, 1)',
            'rgba(54, 162, 235, 1)',
            'rgba(255, 206, 86, 1)',
            'rgba(75, 192, 192, 1)',
            'rgba(153, 102, 255, 1)',
            'rgba(255, 159, 64, 1)'
        ],
        borderWidth: 1
    }]
},
options: {
    scales: {
        y: {
            beginAtZero: true
        }
```

```
                    }
                }
            });
        </script>
    </div>
</div>
</div>
{% endblock %}
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Log in</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.mi
```

```
n.css"
rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
</head>
<body>
    <!-- Section: Design Block -->
<section class="background-radial-gradient overflow-hidden vh-100">
    <style>
      .background-radial-gradient {
        background-color: hsl(218, 41%, 15%);
```

```
  background-image: radial-gradient(650px circle at 0% 0%,
      hsl(218, 41%, 35%) 15%,
      hsl(218, 41%, 30%) 35%,
      hsl(218, 41%, 20%) 75%,
      hsl(218, 41%, 19%) 80%,
      transparent 100%),
    radial-gradient(1250px circle at 100% 100%,
      hsl(218, 41%, 45%) 15%,
      hsl(218, 41%, 30%) 35%,
      hsl(218, 41%, 20%) 75%,
      hsl(218, 41%, 19%) 80%,
      transparent 100%);
}
#radius-shape-1 {
  height: 220px;
  width: 220px;
  top: -60px;
  left: -130px;
```

```
    background: radial-gradient(#44006b, #ad1fff);
    overflow: hidden;
  }
  #radius-shape-2 {
    border-radius: 38% 62% 63% 37% / 70% 33% 67% 30%;
    bottom: -60px;
    right: -110px;
    width: 300px;
    height: 300px;
    background: radial-gradient(#44006b, #ad1fff);
    overflow: hidden;
  }
  .bg-glass {
    background-color: hsla(0, 0%, 100%, 0.9) !important;
    backdrop-filter: saturate(200%) blur(25px);
  }
</style>
<div class="container px-4 py-5 px-md-5 text-center text-lg-start my-
```

```
5">
      <div class="row gx-lg-5 align-items-center mb-5">
        <div class="col-lg-6 mb-5 mb-lg-0" style="z-index: 10">
          <h1 class="my-5 display-5 fw-bold ls-tight" style="color: hsl(218, 81%, 95%)">
            The best application <br />
            <span style="color: hsl(218, 81%, 75%)">to track your expenses</span>
          </h1>
        </div>
        <div class="col-lg-6 mb-5 mb-lg-0 position-relative">
          <div id="radius-shape-1" class="position-absolute rounded-circle shadow-5-
strong"></div>
          <div id="radius-shape-2" class="position-absolute shadow-5-strong"></div>
          <div class="card bg-glass">
            <div class="card-body px-4 py-5 px-md-5">
              <form action="/signinform" method="POST">
                <!-- Email input -->
```

```
<div class="form-outline mb-4">
  <input type="email" id="form3Example3" class="form-control" name="email"
required/>
  <label class="form-label" for="form3Example3">Email address</label>
</div>
<!-- Password input -->
<div class="form-outline mb-4">
  <input type="password" id="form3Example4" class="form-control"
name="pass" required/>
  <label class="form-label" for="form3Example4">Password</label>
</div>
<!-- Submit button -->
<button type="submit" class="btn btn-primary btn-block mb-4">
  Log in
</button>
```

```
            <!-- Register buttons -->
            <div class="text-center">
              <a href="/signup">No account? Sign up here</a>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
 </section>
 <!-- Section: Design Block -->
</body>
</html>
```

signup.html

```
<!DOCTYPE html>
<html lang="en">
<head>
```

<meta charset="UTF-8">

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

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>SignUp</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"

rel="stylesheet" integrity="sha384-

iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"

crossorigin="anonymous">

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"

integrity="sha384-

u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"

crossorigin="anonymous"></script>

</head>

```
<body>
  <!-- Section: Design Block -->
<section class="background-radial-gradient overflow-hidden vh-100">
  <style>
    .background-radial-gradient {
      background-color: hsl(218, 41%, 15%);
      background-image: radial-gradient(650px circle at 0% 0%,
        hsl(218, 41%, 35%) 15%,
        hsl(218, 41%, 30%) 35%,
        hsl(218, 41%, 20%) 75%,
        hsl(218, 41%, 19%) 80%,
        transparent 100%),
      radial-gradient(1250px circle at 100% 100%,
        hsl(218, 41%, 45%) 15%,
        hsl(218, 41%, 30%) 35%,
        hsl(218, 41%, 20%) 75%,
        hsl(218, 41%, 19%) 80%,
        transparent 100%);
```

```
}
#radius-shape-1 {
  height: 220px;
  width: 220px;
  top: -60px;
  left: -130px;
  background: radial-gradient(#44006b, #ad1fff);
  overflow: hidden;
}
#radius-shape-2 {
  border-radius: 38% 62% 63% 37% / 70% 33% 67% 30%;
  bottom: -60px;
  right: -110px;
  width: 300px;
  height: 300px;
  background: radial-gradient(#44006b, #ad1fff);
  overflow: hidden;
}
```

```
      .bg-glass {
        background-color: hsla(0, 0%, 100%, 0.9) !important;
        backdrop-filter: saturate(200%) blur(25px);
      }
    </style>
  <script>
    function validate(){

if(document.getElementById("pass").value.trim()==document.getEleme
ntById("cpass").v
alue.trim())
          return(true);
        alert("Password Mismatch");
        return false;
    }
  </script>
    <div class="container px-4 py-5 px-md-5 text-center text-lg-start my-
5">
      <div class="row gx-lg-5 align-items-center mb-5">
```

```
<div class="col-lg-6 mb-5 mb-lg-0" style="z-index: 10">

  <h1 class="my-5 display-5 fw-bold ls-tight" style="color: hsl(218, 81%, 95%)">

    The best application <br />

    <span style="color: hsl(218, 81%, 75%)">to track your expenses</span>

  </h1>

</div>

<div class="col-lg-6 mb-5 mb-lg-0 position-relative">

  <div id="radius-shape-1" class="position-absolute rounded-circle shadow-5-

strong"></div>

  <div id="radius-shape-2" class="position-absolute shadow-5-strong"></div>

  <div class="card bg-glass">

    <div class="card-body px-4 py-5 px-md-5">

      <form onsubmit="return validate()" action="/signup" method="POST">

        <!-- 2 column grid layout with text inputs for the first and last names -->
```

```
            <div class="form-outline mb-4">
              <input type="text" id="form3Example1"
placeholder="Enter your name"
class="form-control" name="uname" required/>
              <label class="form-label"
for="form3Example1">Name</label>
            </div>
            <!-- Email input -->
            <div class="form-outline mb-4">
             <input type="email" id="form3Example3"
placeholder="Enter your email
address" class="form-control" name="email" required/>
              <label class="form-label"
for="form3Example3">Email</label>
            </div>
            <!-- Password input -->
            <div class="form-outline mb-4">
             <input type="password" id="pass" placeholder="Enter a
password"
class="form-control" name="pass" required/>
```

```
            <label class="form-label" for="pass">Password</label>
          </div>
          <div class="form-outline mb-4">
            <input type="password" id="cpass" placeholder="Confirm password"
class="form-control" required/>
            <label class="form-label" for="cpass">Confirm Password</label>
          </div>
          <!-- Submit button -->
          <button type="submit" class="btn btn-primary btn-block mb-4">
            Sign up
          </button>
          <!-- Register buttons -->
          <div class="text-center">
            <a href="/signin">Already have an account? Log in here</a>
          </div>
        </form>
```

```
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>
  <!-- Section: Design Block -->
</body>
</html>
```

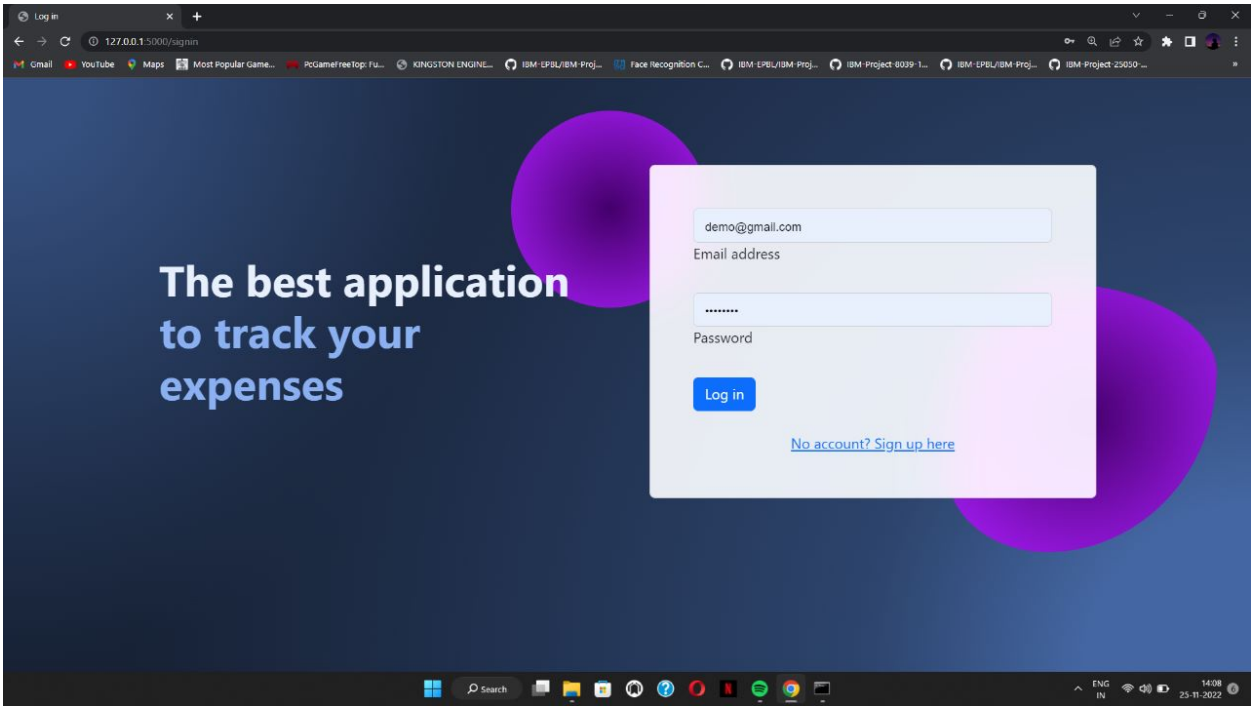
updateexpense.html

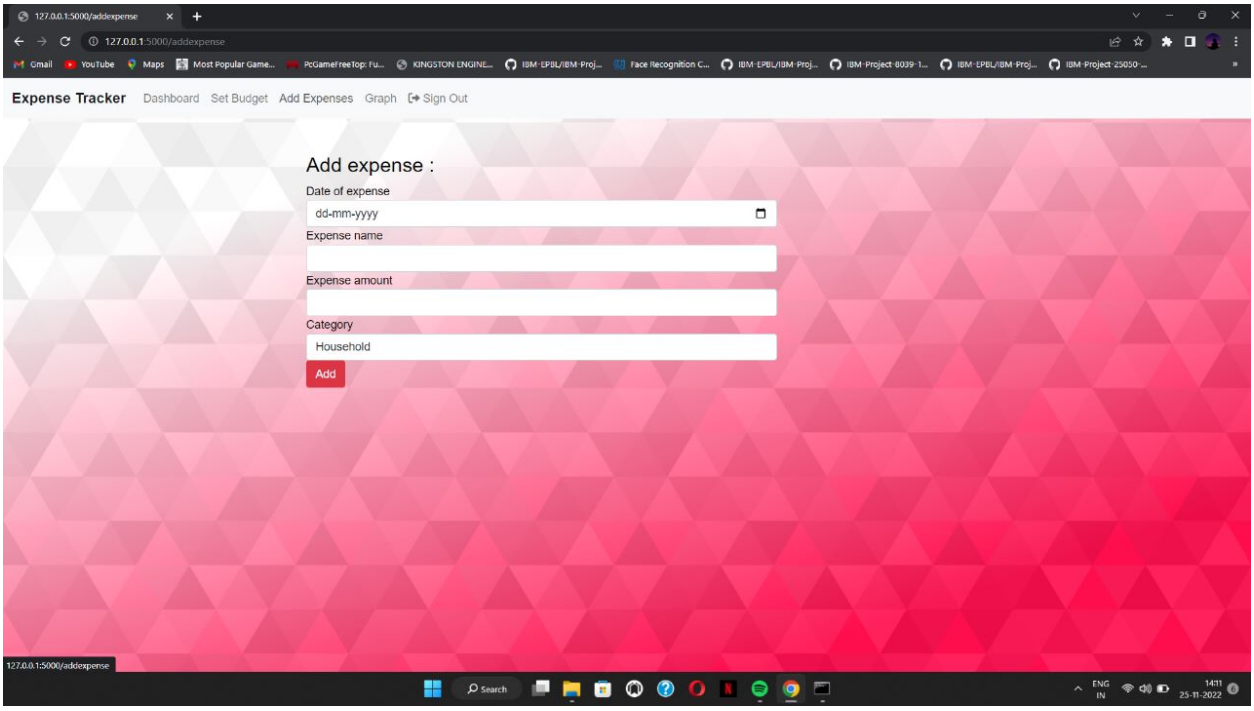
```
{% extends 'base.html' %}
{% block body %}
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <h3 class="mt-5"><i>Edit expense:</i></h3>
            <form method='POST' action = '/edit'>
                <input type="hidden" name="id" value="{{acc['ID']}}" >
```

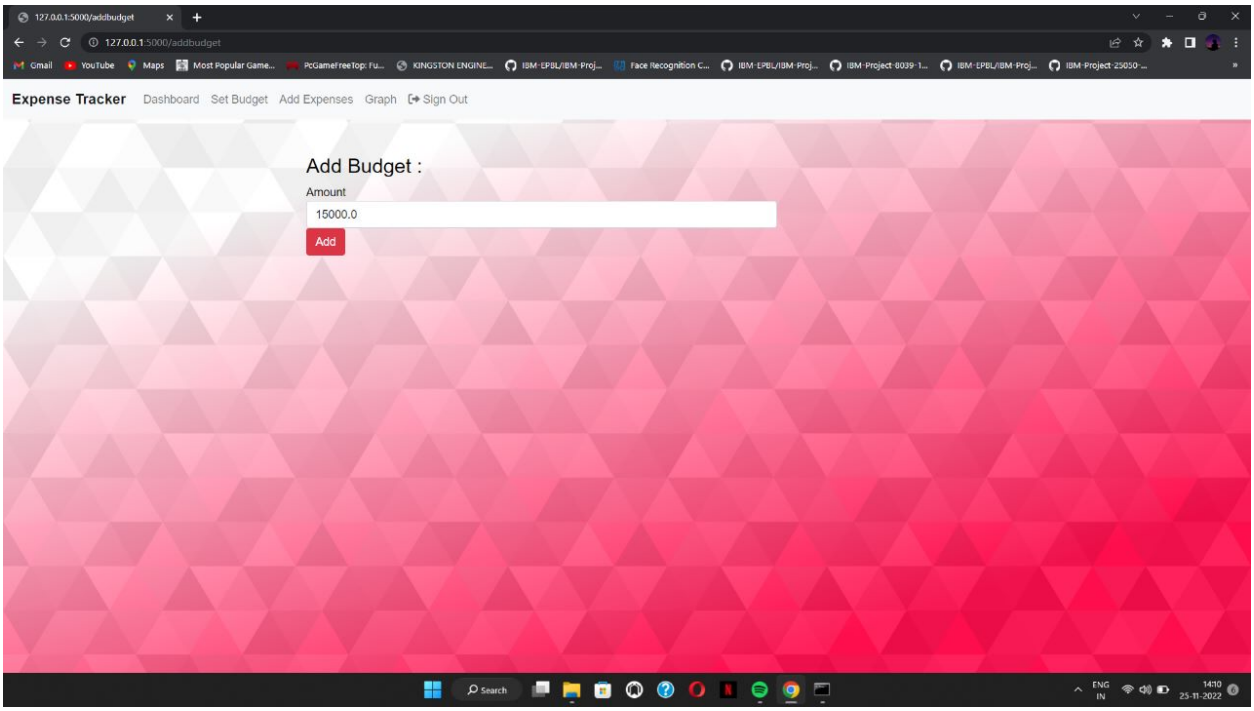
```
            <div class="form-group">
                <label>Date of expense</label>
                <input class="form-control" type="date" name="date"
id="date"
value="{{acc['EDATE']}}">
            </div>
            <div class="form-group">
                <label>Expense name</label>
                <input class="form-control" type="text"
name="expensename"
id="expensename" value="{{acc['ENAME']}}">
            </div>
            <div class="form-group">
                <label>Expense amount</label>
                <input class="form-control" type="text" name="amount"
id="amount"
value="{{acc['EAMOUNT']}}">
            </div>
            <div class="form-group">
```

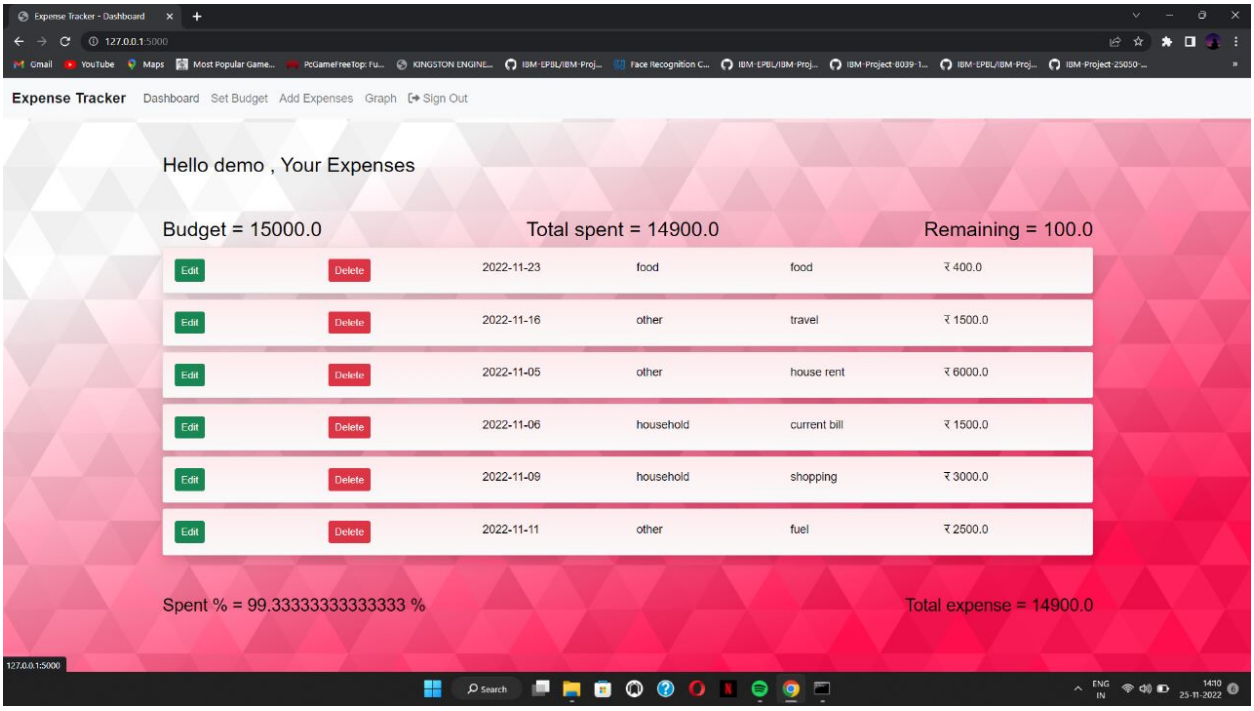
```
                <label>Category</label>
                <select class="form-control" name="category"
id="category">
                    <option value="household">Household</option>
                    <option value="food">Food</option>
                    <option value="entertainment">Entertainment</option>
                    <option value="business">Business</option>
                    <option value="other" selected>Other</option>
                </select>
            </div>
            <input class="btn btn-danger" type="submit"
value="Update">
        </form>
    </div>
  </div>
</div>
{% endblock %}
```

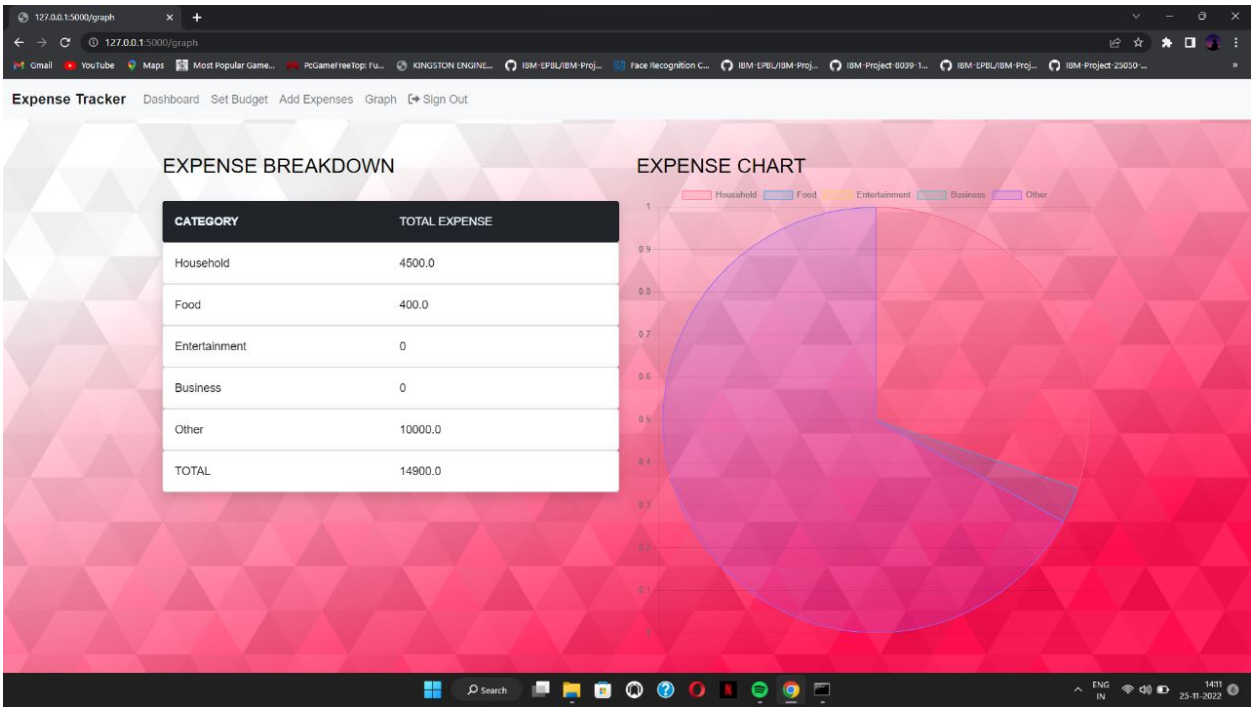
RESULTS:-


Log in
127.0.0.1:5000/signin
The best application to track your expenses
demo@gmail.com
Email address
Password
Log in
No account? Sign up here


127.0.0.1:5000/addexpense
Expense Tracker
Dashboard
Set Budget
Add Expenses
Graph
Sign Out
Add expense :
Date of expense
dd-mm-yyyy
Expense name
Expense amount
Category
Household
Add


127.0.0.1:5000/addbudget
Expense Tracker
Dashboard
Set Budget
Add Expenses
Graph
Sign Out
Add Budget :
Amount
15000.0
Add


Expense Tracker - Dashboard
127.0.0.1:5000
Expense Tracker
Dashboard
Set Budget
Add Expenses
Graph
Sign Out
Hello demo , Your Expenses
Budget = 15000.0
Total spent = 14900.0
Remaining = 100.0
Edit
Delete
2022-11-23
food
food
₹ 400.0
Edit
Delete
2022-11-16
other
travel
₹ 1500.0
Edit
Delete
2022-11-05
other
house rent
₹ 6000.0
Edit
Delete
2022-11-06
household
current bill
₹ 1500.0
Edit
Delete
2022-11-09
household
shopping
₹ 3000.0
Edit
Delete
2022-11-11
other
fuel
₹ 2500.0
Spent % = 99.33333333333333 %
Total expense = 14900.0
127.0.0.1:5000



ADVANTAGES & DISADVANTAGES:-

ADVANTAGES

The best organizations have a way of tracking and handling these reimbursements. This ideal practice guarantees that the expenses tracked are accurately and in a timely manner. From a company perspective, timely settlements of these expenses when tracked well will certainly boost employees' morale Financially Aware and Improve Money Management tracking your expenditures ensures you achieve your project financial targets. How is that? By clearly understanding your project spending using project budget limits, you can aptly make the necessary changes to complete your project within time and budget. 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. Tracking the project expenses by asking team members to provide receipts is helpful to avoid conflict and maintain compliance also. An excellent reporting mechanism is extremely helpful to support the amount to be reimbursed to your team and also invoicing to your customer

DISADVANTAGES

Spending more than necessary

Of course, if you have a large budget, certain departments may feel like they need to spend all the money you allocated to them. This can lead to overspending over time and cause you to inflate future budgets. Rather than implementing a use it or lose it policy, make it clear that each time you make a new budget, you'll consider the current needs of your team rather than historical trends. This can help ensure people are only spending what they need, rather than spending more to ensure their budget stays big.

Finding the time for it

Creating a budget can take a lot of time. You need to make sure you're looking at every aspect of each of your teams' spending.

You'll also need to consider things like upcoming projects, staffing needs, potential promotions or bonuses, travel costs and any other things that might fluctuate throughout your budget cycle. Since this process takes a lot of time, make sure you are devoting enough of your schedule to creating a budget. Your company may even want to hire a contractor or employee who can devote all of their time to creating budgets.

Overlooking important factors

When creating a budget, you may primarily focus on what's going to make your company the most profitable. While this is important, there are other factors you must consider when creating a budget. For instance, your budget can have a significant impact on employee satisfaction. When employees feel like their resources at work are limited, this could lead to high employee turnover rates at your company, which can ultimately lead to more spending to find new employees. By considering more factors than pure profit, you may actually be able to end up increasing your profits.

CONCLUSION:-

The expense tracker application is done succesfully by using the text recognition algorithm and the image uploading basis the expenses of thr person is done and it's achieving the accuracy .In conclusion, developing a personal budget and tracking all expenses and spending is acrucial aspect of personal finances. Set aside a fixed amount in a savings account, they say youshould always have three months work of your living expenses in a savings account in case ofemergencies. Lastly, educating the children early in life about personal finances should be amandatory class in every school. Parents need to proactive in teaching their children aboutbanking, credit card, interest rates, and credit. The importance of actually seeing my spending onmy budget sheet was enlightening.

FUTURE SCOPE:-

Expense management software solutions have come a long way from being mere reporting tools. They have evolved because of the advances in the spheres of automation and artificial intelligence. Expense management software that employ the latest technology are an absolute must-have for every organization, irrespective of the size or industry.

Companies have been gradually adopting expense management software in a bid to mitigate the losses associated with expense

frauds and wasted time. The speed of adoption and change, however, needs to keep pace with emerging trends in the industry.

Technavio analysts believe that the travel and expense management (T&E) software market will grow at a CAGR of 12% in the period 2019 – 2023.

In 2018, companies made advances in expense management practices such as

- digital transformation and moving ERP systems to the cloud
- adopting software with user-centric design and interface
- accepting spending outside of preferred channels.

2019 will see a continuing upswing in the sophistication of software capabilities and maturity in expense management practices.

APPENDIX:-

➤Source code github : **Personal expense tracker**

➤Project demo video : **Personal expense tracker**