

PERSONAL EXPENSE TRACKER APPLICATION

TEAM ID: PNT2022TMID38279

**NALAIYA THIRAN PROJECT BASED LEARNING ON
PROFESSIONAL READINESS FOR INNOVATION
EMPLOYMENT AND ENTREPRENEURSHIP.**

A PROJECT REPORT BY

Jagadesh M (412619104013)

Mohammed Ashik A (412619104021)

Kamal Raj S (412619104016)

Ajay Krishnakanth R V (412619104001)

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESWARAA COLLEGE OF TECHNOLOGY,

Pondur, Sriperumbudur, Tamil Nadu 602105

ANNA UNIVERSITY: CHENNAI 600025

INDEX

1. INTRODUCTION

1. Project Overview
2. Purpose

2. LITERATURE SURVEY

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

3. IDEATION & PROPOSED SOLUTION

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

4. REQUIREMENT ANALYSIS

1. Functional requirement
2. Non-Functional requirements

5. PROJECT DESIGN

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

6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule

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

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

8. RESULTS

1. Sample Code.

9. ADVANTAGES & DISADVANTAGES

10. CONCLUSION

11. FUTURE SCOPE

12. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

Personal Expense Tracker is a web application that allows you to track the daily expense of the user and help them to keep track of their expenses daily, monthly, weekly and yearly basis. It will also create a digital record for the user's income and various expenses spent by the user is calculated. It also gets the input from the user as how much income earned and the date of the income earned and further creates a transaction entry and sums up all the total income. It will be very helpful for the users to manage their needs and they can spend in a better way by keeping track of the application daily basis.

1.2 Purpose

The main purpose of personal expense tracker application is used to keep track of expenses based on the user income and how much they spent and they can keep track of their expenses daily, monthly, weekly and yearly basis.

2. LITERATURE SURVEY

2.1 Existing Problem

The problem of current generation population is that they can't remember where all of the money they earned have gone and ultimately have to live while sustaining the little money they have left for their essential needs. In this time there is no such perfect solution which helps a person to track their daily expenditure easily and efficiently and notify them about the money shortage they have. For doing so have to maintain long ledgers or computer logs to maintain such data and the calculation is done manually by the user, which may generate error leading to losses.

Not having a complete tracking.

2.2 Reference

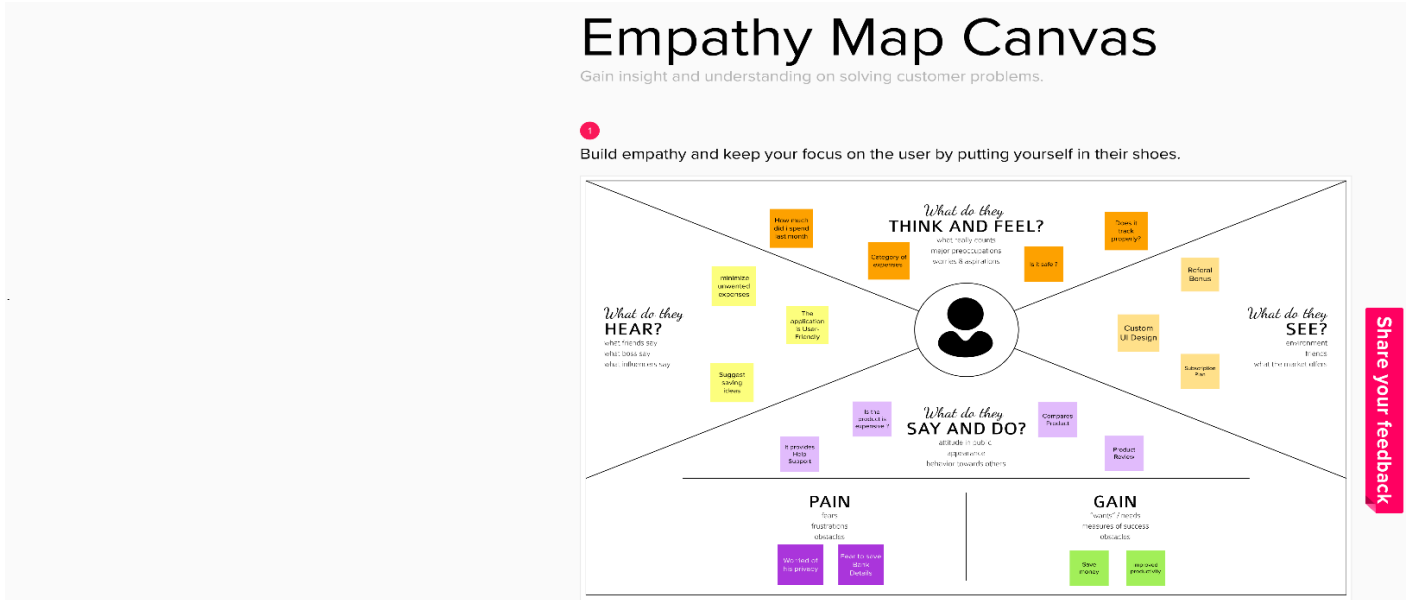
- <https://flask.palletsprojects.com/en/2.1.x/tutorial/>
- <https://www.ibm.com/docs/en/db2-for-zos/11?topic=zos-db2-basics-tutorial-working-db2-interactively>
- <https://www.ibm.com/cloud/kubernetes-service/kubernetes-tutorials>
- <https://www.ibm.com/in-en/cloud/learn/containerization>
- https://www.researchgate.net/publication/347972162_Expense_Manager_Application
- <https://www.spendee.com/>

2.3 Problem Statement Definition

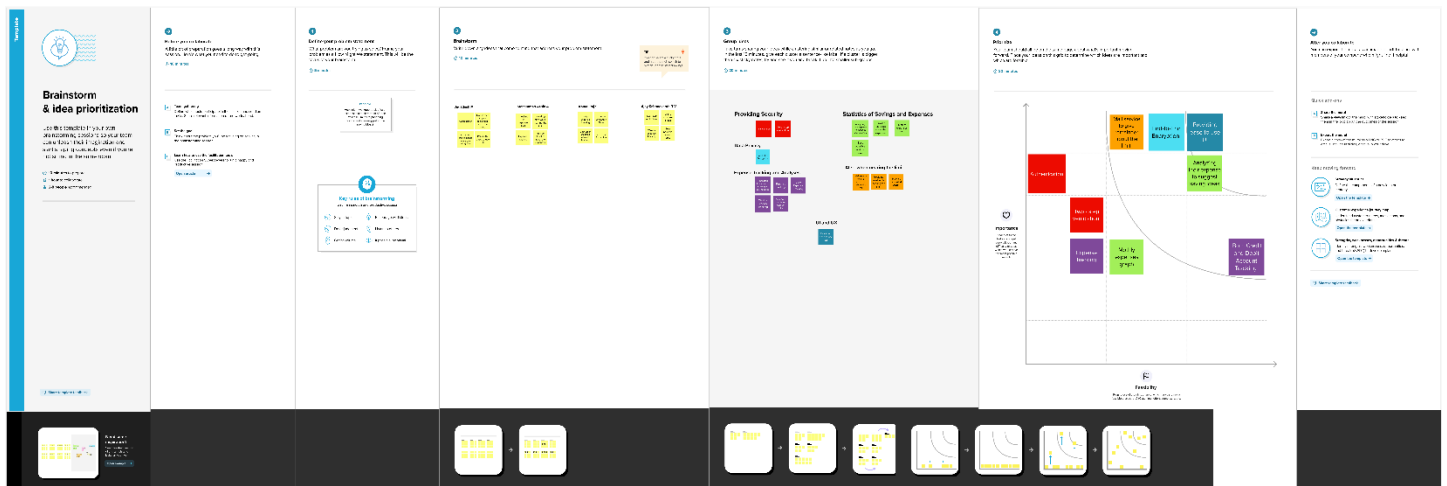
This Expense Tracker is a web application that facilitates the users to keep track and manage their personal as well as business expenses. This application helps the users to keep a digital diary. It will keep track of a user's income and expenses on a daily basis. The user will be able to add his/her expenditures instantly and can review them anywhere and anytime with the help of the internet. He/she can easily import transactions from his/her mobile wallets without risking his/her information and efficiently protecting his/her privacy. This expense tracker provides a complete digital solution to this problem. Excel sheets do very little to help in tracking. Furthermore, they don't have the advanced functionality of preparing graphical visuals automatically. Not only it will save the time of the people but also it will assure error free calculations. The user just has to enter the income and expenditures and everything else will be performed by the system. Keywords: Expense Tracker, budget, planning and savings.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Earlier, we use to track all their expenses by writing down in a small notebook and calculating it on their own Even still many of them follow the same to maintain their financial expenses even some of them don't care of their expenses and spendings. Not only in our homes ,Expenses are need to be tracked in many large scale and small scale sectors such as in many schools, colleges, marketing companies , departmental stores , etc So in order to optimize their work and make peoples life easier our expense tracker application will be much helpful for financial management The outcome of the application will be much useful for them to acknowledge the daily expenses and track the monthly expenses from their income with a limit to spend. They can easily track and view their expenses with a statistical data. In short, tracking our financial expenses is a great deal especially in this scenario so making those tracking easier is the job of this application
2.	Idea / Solution description	Due to the busy and hectic lifestyle people tend to overlook their budget and end up spending an excessive amount of money since they usually didn't plan their budget wisely. user cannot predict future expenses. While they can write down their expenses in a excel spreadsheet, their lack of knowledge in managing finances will be a problem
3.	Novelty / Uniqueness	This application tracks your every expenses anywhere and anytime without using the paper work. Just click and enter your expenditure. to avoid data loss, quick settlements and reduce human error. To provide the pie chart or graph lines in this application.
4.	Social Impact / Customer Satisfaction	Using this application one can track their personal expenses and frame a monthly/annual budget. If your expense exceeded than specified limit, the application will show you an alert message in form of a pie chart.
5.	Business Model (Revenue Model)	Business people can use subscription/premium feature of this application to gain revenue.
6.	Scalability of the Solution	IBM cloud will automatically allocate the storage for the users.

3.4 Problem Solution Fit

Project Title : Personal Expense Tracker Application

Project Design Phase-I - Solution Fit

Team ID PNT2022TMD38279

<p>1. CUSTOMER SEGMENT(S) Who is your customer? i.e. working parents of 0-5 y.o. kids</p> <p>-person who make budget -person who needs to track their daily expenses -person who makes weekly or monthly budget -persons who makes more expenses than their income</p>	<p>6. CUSTOMER CONSTRAINTS What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.</p> <p>-Stable Network Connection -Customers must provide their expense information -subscriptions</p>	<p>5. AVAILABLE SOLUTIONS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking</p> <p>- User friendly interface. - Avoiding misleading ads</p>
<p>2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.</p> <p>- User Data security - Backup and Restore all information - Alerts given based expense limit</p>	<p>9. PROBLEM ROOT CAUSE What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.</p> <p>- People need to have internet to access the Application. - People need to mention expense and income everytime they make a transaction. - While they enter the data, they must remember all the payment they made for the day</p>	<p>7. BEHAVIOUR What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)</p> <p>- User may get confused of their transactions They made. - People skip half of the payment they made for day</p>

3. TRIGGERS

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

- Seeing good financial management and their benefits.
- Guide them and make them aware about their daily expenses

TR**4. EMOTIONS: BEFORE / AFTER**

How do customers feel when they face a problem or a job and afterwards?

i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

- Before using this application they were spending their expenses lavishly
- After using this application they are more focused and concentrated about their expenses they make

EM**10. YOUR SOLUTION**

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.

If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

- Generate monthly report and forecast budget for the users
- we will provide two modes
- Manual mode -- were the user needs to enter the data manually.
- Automatic mode -- were the data will gets updated automatically

SL**8. CHANNELS of BEHAVIOUR****ONLINE**

What kind of actions do customers take online? Extract online channels from #7

- Their daily expense get update to the cloud

OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

- Their daily expense budget calculation will be done

CH

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration Form for collecting details.
FR-2	User Login	Enter Username and Password.
FR-3	Forgot Password	Resetting the password by sending an OTP to user's mail
FR-4	Calendar	Personal expense tracker application must allow user to add the add to their expenses.
FR-5	Dashboard	User can add the expense and can evaluate them using the provided options.
FR-6	Result Page	Show the user result.

4.2 Non-Functional requirement

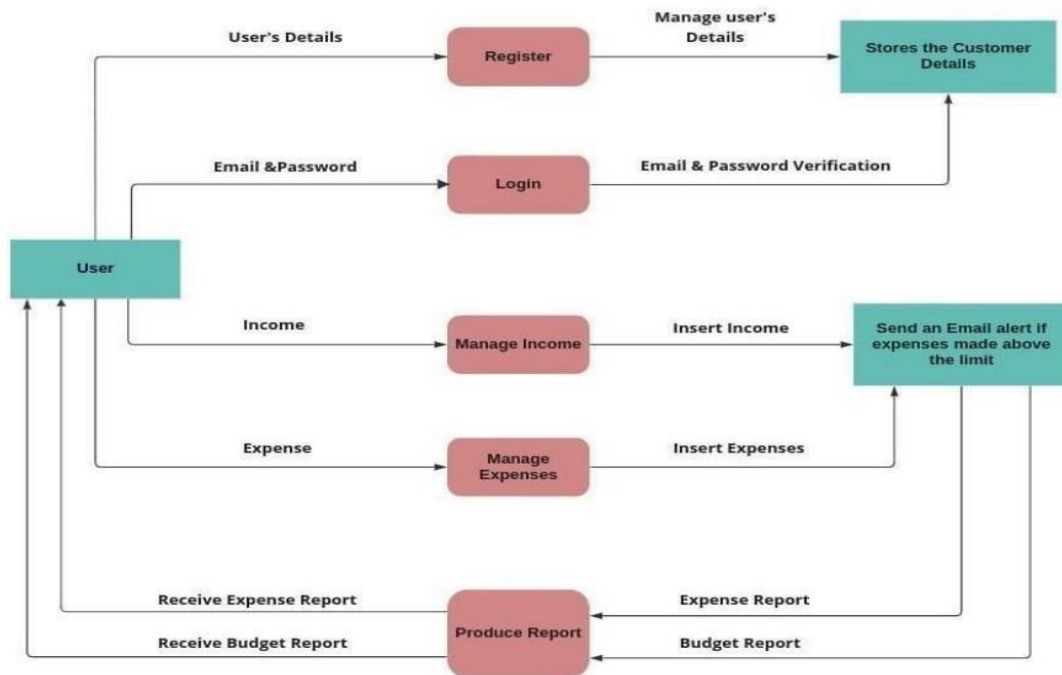
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Customers can use the application in almost all the web browsers. Application is with good looking and detailed UI, Which makes it more friendly to use.
NFR-2	Security	Customers are asked to create an account for themselves using their email which is

		protected with an 8 character-long password , making it more secure.
NFR-3	Reliability	Each data record is stored on well built efficient database schema. There is no risk of data loss.
NFR-4	Performance	Customers will have a smooth experience while using the application, as it is simple and is well optimised.
NFR-5	Availability	Application is available 24/7 as it is hosted on IBM cloud.

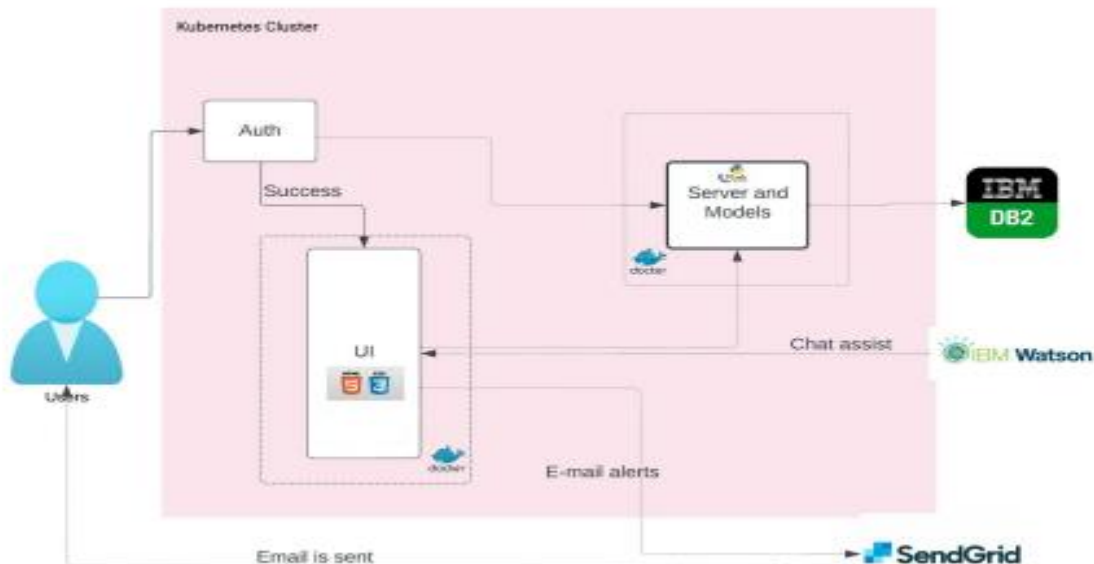
5. PROJECT DESIGN

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 store.



5.2 Solution & Technical Architecture



5.3 User Stories

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 Social media accounts	I can register & access the dashboard with Social media Login	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-5	Once logged in , based on user's expenses and data records , graphical representation is achieved	I can view my updated dashboard	High	Sprint-2
	Financial account	USN-6	As a user, I can add and remove any financial accounts	I can manage financial accounts	High	Sprint-2
	Notifications	USN-7	As a user, I can receive alerting notifications on untracked expenses	Untracked expenses are alerted	High	Sprint-3
		USN-8	As a user, I can receive suggesting notifications for saving and earning money	Suggestions are notified	Low	Sprint-3
	Security	USN-9	As a user, I am assured for linking my financial accounts securely	Security is assured	High	Sprint-4
	Customer care	USN-10	As a user, I can access the customer care for any queries and issues regarding the applications	24/7 customer care support is provided	Low	Sprint-4
Customer Care Executive						

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.	3	High	Jagadesh, Ashik, Kamal raj
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	3	Low	Jagadesh, Ashik, Kamal raj
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	2	High	Jagadesh, Ashik, Ajay
Sprint-1	Wallet page	USN-4	As a user, I can add my monthly income in the wallet section	4	High	Jagadesh, Kamal raj, Ajay
Sprint-2	Adding Expenses	USN-5	As a user, I will track my expenses by collecting expense information	4	High	Ajay, Kamal raj
Sprint-2	Computing Available Balance	USN-6	As a user, The income and expenses get computed and available balance is shown	4	High	Jagadesh, Ashik
Sprint-2	Setting limit balance	USN-7	As a user, I can set the limit of my expenses or balance	4	Low	Kamal raj, Ajay
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Alert Mail	USN-8	As a user, If my balance or expenses goes beyond the set limit a email will be sent to alert the user.	3	High	Jagadesh, Ashik, Kamal raj
Sprint-3	Storing information in cloud database	USN-9	As a user, My data will get stored in cloud database and can be accessed anywhere and anytime	3	High	Jagadesh, Kamal raj, Ajay
Sprint-3		USN-10	As a user, I can give the description of my income and expenses	2	Low	Jagadesh, Ashik
Sprint-3	Retrieving data	USN-11	As a user, I can view my previous transactions in history.	4	High	Jagadesh, Ashik, Ajay
Sprint-4	Update and graph view	USN-12	As a user, I spend the money in the wallet and the wallet will be updated periodically and can see the monthly expenses as a graph	4	Low	Kamal raj, Ajay
Sprint-4	Monthly Report	USN-13	As a user, I can get monthly reports.	4	Low	Ajay, Ashik
Sprint-4	Logout or Delete	USN-14	As a user, I can logout or delete my account	4	Low	Jagadesh, Ashik

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

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

7.1 Feature

Feature 1: Add Expense

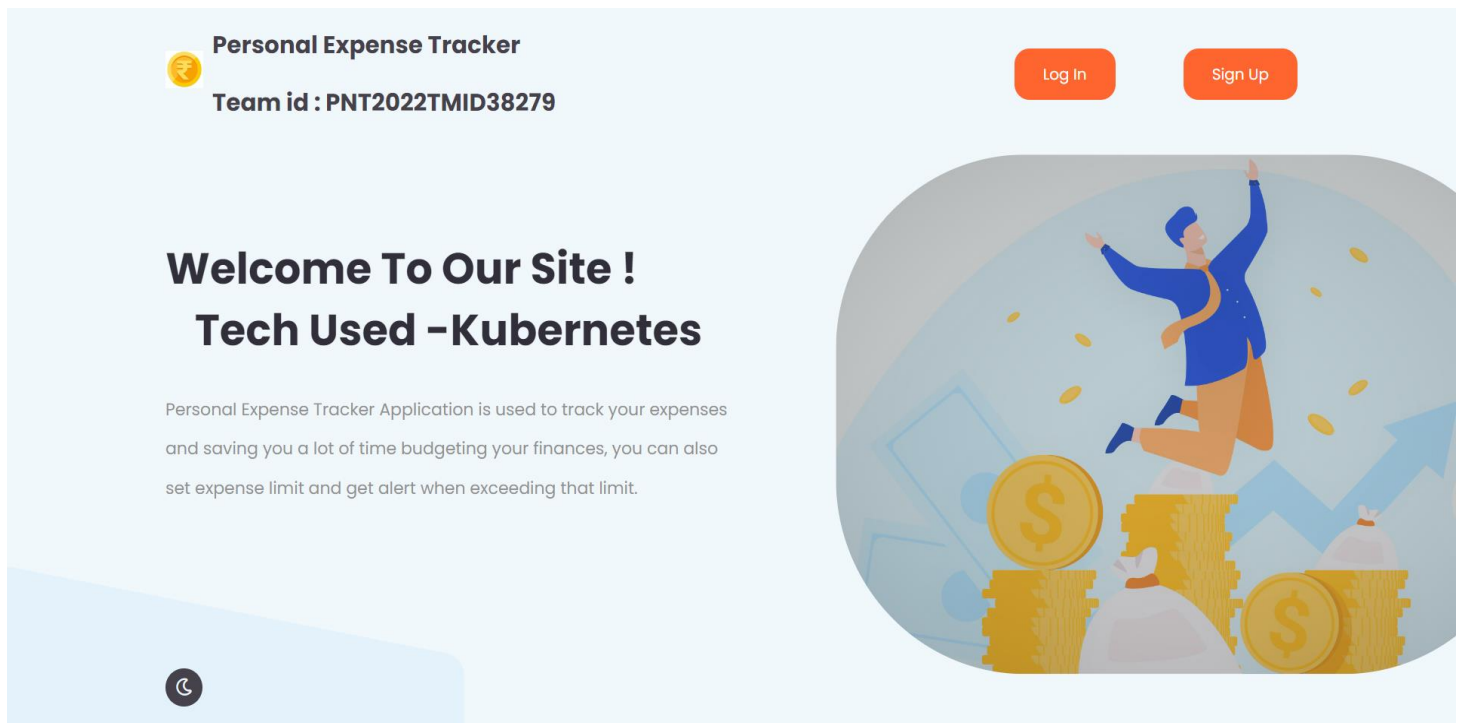
Feature 2: Update Expense

Feature 3: Set Limit

Feature 4: Send Alert Emails to users

7.2 Feature

Track your expenses anywhere, anytime. Seamlessly manage your money and budget without any financial paperwork. Just click and submit your invoices and expenditures. Access, submit, and approve invoices irrespective of time and location. Avoid data loss by scanning your tickets and bills and saving in the app. Approval of bills and expenditures in real-time and get notified instantly. Quick settlement of claims and reduced human errors with an automated and streamlined billing process.



Please fill out the form !

Register

User Name

Email

Password

SUBMIT

Already have an account? [Login Here](#)

Login

User Name

Password

LOGIN

[Forgot Password?](#)

Don't have an account yet? Click here to [register!](#)

GO HOME

Expense Tracker

Log Out

YOUR BALANCE

₹40000

INCOME	EXPENSE
₹50000	₹10000

[Click here](#) to see Transaction History

Add New Transaction

For your reference Enter Date along with Description inside (DD-MM-YY).

Amount

(negative (-) - expense , positive (+) - income)

Submit

Set/Change Expense limit

When limit is set you will be notified when Expenses are made beyond the limit.

User	Transaction	Income	Expense
jagadesh	Salary (14-11-2022)	50000	0
jagadesh	Rent (25-11-2022)	0	10000

[Go Back To Dashboard](#)

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

Find schemas or tables

Refresh

Schemas

<input checked="" type="checkbox"/>	Name	Type	Tables ▲
<input checked="" type="checkbox"/>	RRX76661	User	3

Total: 1, selected: 1

Tables

New table +

<input type="checkbox"/>	Name ▼	Schema	Properties
<input type="checkbox"/>	EXPENSE_TABLE	RRX76661	...
<input type="checkbox"/>	TRANSACTION_TABLE	RRX76661	...
<input type="checkbox"/>	USERS	RRX76661	...

Total: 3, selected: 0



Personal Expense Tracker

Team id : PNT2022TMID38279

Log In

Sign Up

Welcome To Our Site ! Tech Used - Docker

Personal Expense Tracker Application is used to track your expenses and saving you a lot of time budgeting your finances, you can also set expense limit and get alert when exceeding that limit.



8 CODE:

```
from flask import Flask, flash, session, render_template, url_for, request, redirect, abort
import ibm_db
import re
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

app = Flask(__name__)

app.secret_key = ''

mail_api = ""

hostname = ""
uid = ""
pwd = ""
driver = ""
db = ""
port = ""
protocol = ""
cert = ""

dsn = (
    "DATABASE={0};"
    "HOSTNAME={1};"
    "PORT={2};"
    "UID={3};"
    "SECURITY=SSL;"
    "SSLServerCertificate={4};"
    "PWD={5};"
).format(db, hostname, port, uid, cert, pwd)

conn = ibm_db.connect(dsn, '', '')

@app.route('/')
def home():
    return render_template('landing_page.html')

@app.route('/login', methods=['POST', 'GET'])
def login():
    global userid
    msg = ''

    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
```

```

sql = "SELECT * FROM users WHERE username =? AND password =?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,username)
ibm_db.bind_param(stmt,2,password)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
#print(account)
if account:
    session['loggedin'] = True
    session['id'] = account['EMAIL']
    userid = account['USERNAME']
    session['username'] = account['USERNAME']
    #print(session)

    return redirect(url_for('Dashboard'))

else:
    msg = 'Incorrect username or password !'

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

```

```

@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']
        sql = "SELECT * FROM users WHERE username =? OR EMAIL = ?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        #print(account)
        if account:
            msg = 'Account already exists username or mail already taken!'
        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 !"
        elif re.search(r'[!@#$$%&]', password) is None:
            msg = "Password must contain atleast one special character"
        elif re.search(r'\d', password) is None:
            msg = "Password must contain atleast one digit"
        elif re.search(r'[A-Z]', password) is None:
            msg = "Password must contain atleast one uppercase letter"
        elif not re.match(r'[A-Za-z0-9 !@#$$%&]{6}',password):
            msg = "Password length must be atleast 6 characters long"

```

```

else:
    insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, username)
    ibm_db.bind_param(prepare_stmt, 2, email)
    ibm_db.bind_param(prepare_stmt, 3, password)
    ibm_db.execute(prepare_stmt)

    insert_data = "INSERT INTO EXPENSE_TABLE VALUES (0, 0, 0, 0, 0, ?, 0)"
    data_stmt = ibm_db.prepare(conn, insert_data)
    ibm_db.bind_param(data_stmt, 1, username)
    ibm_db.execute(data_stmt)

    msg = 'You have successfully registered ! , Please click login to continue.'

elif request.method == 'GET':
    msg = 'Please fill out the form !'

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

```

```

@app.errorhandler(500)
def internal_error(error):
    return render_template('log_err.html'), 500

```

```

@app.route('/dashboard', methods = ['POST', 'GET'])
def Dashboard():
    global bal , Inc , exp

```

```

    if session['loggedin'] == True:

        sql3 = "SELECT * FROM Expense_Table WHERE USERID = ?;"
        stmt2 = ibm_db.prepare(conn, sql3)
        ibm_db.bind_param(stmt2, 1, session['username'])
        ibm_db.execute(stmt2)
        record = ibm_db.fetch_assoc(stmt2)
        #print(record)

        bal = "₹" + str(record['BALANCE'])
        Inc = "₹" + str(record['INCOME'])
        exp = "₹" + str(record['EXPENSE'])

    if request.method == 'POST':
        amount = int(request.form['rupee'])

```

```

limit = request.form['bal_limit']
detail = request.form['transaction_for']
if amount > 0:
    sql = "UPDATE Expense_Table SET inc = ? , income = income + ? WHERE USERID = ?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,amount)
    ibm_db.bind_param(stmt,2,amount)
    ibm_db.bind_param(stmt,3,session['username'])
    ibm_db.execute(stmt)

    transact_sql = "INSERT INTO TRANSACTION_TABLE VALUES (?, ?, ?, 0)"
    transact_stmt = ibm_db.prepare(conn,transact_sql)
    ibm_db.bind_param(transact_stmt,1,session['username'])
    ibm_db.bind_param(transact_stmt,2,detail)
    ibm_db.bind_param(transact_stmt,3,amount)
    ibm_db.execute(transact_stmt)

elif amount < 0:
    sql = "UPDATE Expense_Table SET exp = ? , expense = expense + ? WHERE USERID = ?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,abs(amount))
    ibm_db.bind_param(stmt,2,abs(amount))
    ibm_db.bind_param(stmt,3,session['username'])
    ibm_db.execute(stmt)

    transact_sql = "INSERT INTO TRANSACTION_TABLE VALUES (?, ?, 0, ?)"
    transact_stmt = ibm_db.prepare(conn,transact_sql)
    ibm_db.bind_param(transact_stmt,1,session['username'])
    ibm_db.bind_param(transact_stmt,2,detail)
    ibm_db.bind_param(transact_stmt,3,abs(amount))
    ibm_db.execute(transact_stmt)

if bool(limit) != False and int(limit) > 0:
    sql_limit = "UPDATE EXPENSE_TABLE SET BAL_LIMIT = ? WHERE USERID = ?"
    stmt_limit = ibm_db.prepare(conn,sql_limit)
    ibm_db.bind_param(stmt_limit,1,limit)
    ibm_db.bind_param(stmt_limit,2,session['username'])
    ibm_db.execute(stmt_limit)

sql2 = "UPDATE Expense_Table SET balance = income - expense WHERE USERID = ?"
stmt = ibm_db.prepare(conn,sql2)
ibm_db.bind_param(stmt,1,session['username'])
ibm_db.execute(stmt)

up_sql = "SELECT * FROM Expense_Table WHERE USERID = ?;"
up_stmt = ibm_db.prepare(conn,up_sql)
ibm_db.bind_param(up_stmt,1,session['username'])
ibm_db.execute(up_stmt)
limit_info = ibm_db.fetch_assoc(up_stmt)

```

```
program continues ...
```


9 ADVANTAGES & DISADVANTAGES

ADVANTAGES:

One of the major pros of tracking spending is always being aware of the state of one's personal finances. Tracking what you spend can help you stick to your budget, not just in a general way, but in each category such as housing, food, transportation and gifts. While a con is that manually tracking all cash that is spent can be irritating as well as time consuming, a pro is that doing this automatically can be quick and simple. Another pro is that many automatic spending tracking software programs are available for free. Having the program on a hand-held device can be a main pro since it can be checked before spending occurs in order to be sure of the available budget.

DISADVANTAGES:

A con with any system used to track spending is that one may start doing it then taper off until it's forgotten about all together. Yet, this is a risk for any new goal such as trying to lose weight or quit smoking. If a person first makes a budget plan, then places money in savings before spending any each new pay period or month, the tracking goal can help. In this way, tracking spending and making sure all receipts are accounted for only needs to be done once or twice a month. Even with constant tracking of one's spending habits, there is no guarantee that financial goals will be met. Although this can be considered to be a con of tracking spending, it could be changed into a pro if one makes up his or her mind to keep trying to properly manage all finances.

10 CONCLUSION

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

11 FUTURE SCOPE

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

- This application does not provide higher decision capability.
- Does not provide best saving ideas.

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

- Multiple language interface.
- Provide backup and recovery of data.
- Provide better user interface for user.
- Mobile apps advantage.
- Provide Graphical presentation.
- Better Saving ideas.

12 APPENDIX

Source Code Git hub Link :

<https://github.com/IBM-EPBL/IBM-Project-3558-1658579678>

Project Demo Link :

<http://169.51.205.212:30786/>