

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Summary

1.2 Purpose

2. A SURVEY OF THE BOOKS

2.1 Current issue

2.2 References

2.3 Definition of Problem Statement

3. ORIGINAL IDEA & PROPOSED SOLUTION

3.1 Canvas Empathy Map

3.2 Brainstorming and Idea Generation

3.4 Problem Solution Fit

4. REQUIREMENT ASSESSMENT

4.1 Functional necessity

4.2 Non-functional specifications

5.PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT SCHEDULING & PLANNING

6.1 Planning & Estimation for the Sprint

6.2 Delivery schedule for Sprint

7.CODING AND SOLUTIONING

7.1 Feature 1

7.2 Feature 2

8. TESTING

8.1 Test Examples

8.2 User Acceptance Testing (UAT)

9. RESULTS

9.1 Performance Measures

9.2 Output

10. BENEFITS AND DISADVANTAGES

11. CONCLUSION

12. PERSPECTIVE

13.APPENDIX

CHAPTER 1

INTRODUCTION

Expense Tracker is a web application that allows users to easily track and manage personal and business expenses. This application helps users to run a digital dairy farm. It is designed for non-employed and office workers who want to track their daily expenses. Track user income and expenses daily. A user who adds spending and is updated based on the spending wallet balance. He can see exactly how long a particular product has been in use. Expenses can be analyzed graphically. Users can also easily access stored data. This is a simple and effective method with computerized systems that tend to eliminate manual record keeping. Users can instantly add expenses and view details with an internet connection. Modern life offers many options for consumer services and goods. As a result, people's spending increased dramatically. Compared to ten years ago, the cost of living is increasing by the day. Therefore, it is important to keep an eye on your spending to live a good life within a reasonable budget.

1.1 Project Overview

In today's world, financial well-being is everyone's dream, and managing and tracking spending plays a key role in that goal. The Daily Expense Tracker System is designed to track your organization's income and expenses daily. When people cannot control their spending, they are more likely to be in crisis. Money management is an essential part of life. A proper balance between income and expenditure is essential to lead a comfortable life. You want some savings that you can use later in life if you want. But without proper money management, there would be no savings at all. Some people have a good habit of writing down all their problems. Expense tracking software can help you cut down on unnecessary spending. Using an online expense tracker has several advantages and benefits. Expense tracking makes any business run faster and smoother. People misjudge how much money to spend on contingencies and buy from their mental budget. So money movements only matter when they come in and are often forgotten when they go out. The Daily Expense Tracker System is

designed to monitor users' earnings and expenses on a daily basis increase. With this approach, money is distributed based on daily expenses. The system will deduct the surplus of your daily spending from your earnings. In this case, a new daily spending quota will be provided. If the day costless, the system saves it. The daily expense tracking system's monthly report shows you the curve of your income and expenses. All economic pawns and players should learn personal money management skills as the economy is struggling and full of temptations to spend money. However, because financial information comes in many forms, it can be difficult to manage funds (invoices, bank statements, invoices, pay slips, receipts, etc.). As a result, consumers are often unable to find these financial records while searching, let alone manage and monitor their cash flow.

1.2 Purpose

An expense tracker gives you a better understanding of your financial situation and can help identify potential problems before they arise. This way, you'll be able to build changes and be ready once the issue arises. Also called expense manager and cash manager, an expense tracker could be a software or application that helps to stay a correct record of your money flow and outflow. many folks in India continue to exist a set income, and that they realize that towards the tip of the month they do not have decent money to satisfy their needs.

CHAPTER 2 LITERATURE SURVEY

AUTHOR: Miriam Thomas, Lekshmi P, and Dr.mahalekshmi

YEAR OF PUBLICATION: 2020

This System divides the Income based on daily expenses. If exceed day's expense, system will calculate income and will provide new daily expense allowed amount. The receipts can be defined as a document record which, there is a transfer of ownership over a good or some commercial activity between people and companies. But in general, how important are those receipts? They are intended to record the transfer of a monetary value between the parties and can be used also in broader contexts and yet may be used to cancel the validity of another already existing receipt. With the aim of proving that the transaction is really valid, the receipts appeared around the 18th century and until the present days they are evidence that a purchase has been made. It is worth noting that, for each transaction carried out, at least one paper receipt will be issued proving, therefore, that the transaction occurred. In general, most people from a user perspective tend to have the habit of keeping track of what was spent and for that reason, they choose to keep the invoice in order to be in control of what was spent on that specific purchase. The receipts can be defined as a document record which, there is a transfer of ownership over a good or some commercial activity between people and companies. But in general, how important are those receipts.

DISADVANTAGE:

- Lack of proper explaining and not able to run Windows Operating system.

TITLE : EXPENSES MANAGER

APPLICATION AUTHOR: Velmurugan, Albet

Mayan, Niranjana, Richard Francis

YEAR OF PUBLICATION: 2020

The Expense Manager is a multi-purpose finance related android application intended to run on android devices. Expense tracker is a web application used to track user expenses and generates periodical reports about the savings and expenditure. In this project, we propose an application known as "Expense Tracker," which is helpful to manage our income and expense daily or periodically or else whenever we want to remind. It also acts as an indicator or reminder example in the fastest world in which we cannot remember what the things we must do for the end of the month are and the payments we must pay for the month. Due to some conflict or other stress, we sometimes forget what the income is, where the money must come from, or the payments we must pay. This app will assist you in keeping track of what you need to do at the end of the month. For instance, what are the monthly expenses, and how much do they cost? Food expenses, phone, power, taxation, and other personal charges are some of the expense features. If we are a businessman with a multi-business, we do not know from which part of the business income has come and how much income has come for us, but with the help of this application, we can divide and store all of the income and set a reminder for a specific date to remain so that we can manage and finalize the income for us.

DISADVANTAGE:

- Reduce the security that end-to-end privacy offers.

TITLE: A NOVEL EXPENSE TRACKER USING STATISTICAL

ANALYSIS AUTHOR: Muskaan Sharma, Ayush Bansal, Dr. Raju Rajan

YEAR OF PUBLICATION: 2021

Expense Tracker is used to maintain and manage data of daily expenditure in a more precise way it can give profound knowledge of their expenses. “Expense Tracker” to efficiently manage house-old budget. Our system will allow user to keep track of their expenses. Some statistical analysis has to be done to be able to give users correct information on their expenses and help them spend better. This helps the society to help them from issues like bankruptcy and save time from manual calculations. For using such application, a user needs to provide his/her total income or the amount he/she is spending per day and each user details or information are going to be stored in a unique way. Every user is required to register on the system to create a record unique to the user. This novel expense Tracker uses statistical analysis which is going to keep a track of your expenses and would even give you results accordingly. If you spend too much on a particular day this smart feature will help you to sort things out it will minus the expenses from the next day or if it lesser then it will add your saved amount for that day in the savings. Tracking application will generate a report accordingly on monthly or weekly basis and would generate a statistical analysis of your expenses in a more sorted and easier to understand way.

DISADVANTAGE:

- Your data is likely being used and sold, and it is less secure

TITLE: DAILY EXPENSES TRACKER

AUTHOR: Tamia Ruvimbo Masendu Aanajey Mani Tripath

YEAR OF PUBLICATION: 2022

Daily Expense Trackers lets in the person in accordance with hold a computerized diary. It offers a class dole on one's expenses. Budget Tracking involves recording and analyzing the incomes and expenses of a person or an organization over a particular period of time. Today, since we are living in a hurry up and get it done society, many people are looking forward to efficient ways to budget their time and money. During the recent years, some research has been carried out on household budget. It has been noted that in mostcases, budget management is being done mentally and never being put on paper which makes Budget Tracking very difficult. This application will help you to make a note for what or the things we have to do for the end of month. For example, like how much it expenses for monthly and what are the expenses for a month. Some of the expense features like food expenses billing expenses like phone, electricity, taxation and some other personal expenses. In this fast- moving world this web application will be very useful for a people who was a family and especially for a business people. Budgeting is an integral part of the society. Due to some conflict or some other stress we forget some times that what are the income or where the money has to come from or what the payments we have to pay. In this project we propose a web application known as “Online Income and Expense Tracker” which is helpful to manage out income and expense as a daily or periodically or else whenever we want to remind. It also acts as an indicator or reminder example in the fastest world which we can't able to remember what are the things we have to do for the end of month and what arethe payments we have to pay for the particular month.

DISADVANTAGE:

- To identify alerting notification through email

TITLE: STUDENTS EXPENSES TRACKER

APPLICATION AUTHOR: Saumya Dubey, Pragya Dubey, Rishabh Kumar

YEAR OF PUBLICATION: 2022

The mainly focuses on tracking the expenses of a student and it works like a digital dairy that will help the student to stay away from the manually calculating everything using a pen and paper. It is quite useful to have a tool that will really assess the situation with cash flow. You can't save money without knowing the real picture of your expenses. In any case, starting a habit of managing and controlling personal finances is a good choice for the first important step towards balanced costs and significant savings. Many people start keeping a budget and analyzing their expenses. Unfortunately, there are some limitations such as lack of time or lack of information. Sometimes, a person knows the approximate amount of his or her expenses and profits, but the real statistics of the data differ from the approximate figures. An increase in wages very often does not lead to an increase in free funds. Costs and requirements for services and goods are growing. It is quite useful to have a tool that will really assess the situation with cash flow. You can't save money without knowing the real picture of your expenses. This is because of the constant rise in prices, unstable exchange rates etc. Personal finances will not answer the question 'where can I get money from?', but it can help with a better understanding of where your money comes from and on what it is spent, and based on this information, everyone draws their conclusions. The usefulness of financial management is an understanding of how and for how much money we live, and the right conclusion from financial management is to live according to our capabilities. Want to live (spend more) better or save more - earn more or save more in the long run with control of useless costs. There are different approaches, and everyone benefits from personal finance.

DISADVANTAGE:

- A very common cause of personal budget tensions is the purchase of expensive

2.1 Existing Problems

The Expense tracker existing system does not provide the user portable device management level, existing system only used on desktop so unable to update anywhere expenses done and unable to update the location of the expense details disruptive that the proposed system provides. In existing, we need to maintain the Excel sheets, CSV files for the user daily, weekly and monthly expenses. In existing, there is no such complete solution to keep a track of its daily expenses. To do so a person as to keep a log in a diary or in a computer system, also all the calculations need to be done by the user which may sometimes results in mistakes leading to losses. The existing system is not user friendly because data is not maintained perfectly. This project would not have any reminder to remain a person in a specific date, so that is the only drawback in which the reminder is not present. This project will be an un populated information because it has some disadvantages by not remind a person for each and every month. But it can used to perform calculation on income and expenses to overcome this problem we propose the new project.

DISADVANTAGES:

- It is impossible to update the budget details regularly
- There is possibility of missing the notes in manual process
- It take more time to calculate the expenses by manually
- There is a problem for calculating the year

2.2 References

1. Velmurugan A, Albert Mayan J, Mohana Prasad R ,Yovan Felix A,"Implementing Health Care Center using Hadoop for Analysis and Prediction of Diseases", International Journal of Engineering and Advanced Technology (IJEAT), Vol.8, No.6, pp.3456-3459,2019.
2. <https://www.groupon.com/>
3. Eswar V.O.S, Vinil.B, Ankayarkanni B and Albert,"Integrated Collective Node Behavior Analysis with Onion Protocol for Best and Secured Data Transmission," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2018, pp. 0918- 0921.
4. Mayan, J.A., Priya, K.L., " Novel approach to reuse unused test cases in a GUI based application", IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2015
5. Mohan Prasad K, Sai Nagendra Goru Rajeev, Vamsi Desu, Albert Mayan M.J , "Automated Payroll Using GPS Tracking and Image Capture ", IOP Conference Series: Materials Science and Engineering, Vol.590, 012026 ,pp.1-6,2019 doi:10.1088/1757-899X/590/1/01202
6. Mary Posonia A, S. Vigneshwari, Albert Mayan J, D. Jamunarani,"Service Direct : Platform that Incorporates Service Providers and Consumers Directly",International Journal of Engineering and Advanced Technology (IJEAT) , Vol.8 ,No.6,pp.3301- 3304,2019.

2.3 Problem Statement Definition

There may be a server issue when a user wants to monitor the progress of their expenses. When utilizing the application, the user must be conscious. A person who cannot manage his expenses cannot successfully lead a household and fulfill his goals. In the current world where mobile phones and laptops have become a part of living, such an app would be handy to deal with all our expenses. We can readily track the details of our expenses. We can see how we are spending our money because it keeps accurate records of all of our expense

CHAPTER 3


IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas




3.2 Ideation & Brainstorming


Template




Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

 10 minutes to prepare

 1 hour to collaborate


 2-8 people recommended

Share template feedback

→

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

 10 minutes

A

Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B

Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

C

Learn how to use the facilitation tools


Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1


Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

 5 minutes


PROBLEM


How might we [your problem statement]?





Key rules of brainstorming


To run a smooth and productive session


 Stay in topic.


 Encourage wild ideas.

 Defer judgment.

 Listen to others.

 Go for volume.

 If possible, be visual.

 See a finished version of this template to inspire your work.

Open example →

Need some inspiration?

See a finished version of this template to inspire your work.

Open example →

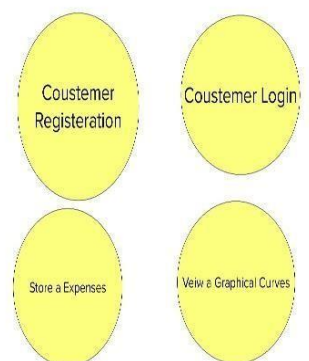
Brainstorm
 The team will brainstorm the features and functions of the system.
 The team will brainstorm the features and functions of the system.



Group Ideas
 The team will group the ideas and select the best ones to develop.
 The team will group the ideas and select the best ones to develop.

Create a account for customer Details

Whenever and where ever we can access the customer details using the internet connection



3.2 Proposed Solution:

The system we propose can send a message by email when the monthly/daily limit is exceeded. Created a user interface for interacting with the application. Future enhancements to the application may be supported on all future Android versions. History can be set to show all details in the app even if the respective data has been deleted from the database. Statistics can be generated based on user income and expense details. You can allow file sharing via Bluetooth. You can print details of each income or expense details. Some of the additional components will likely allow users to register with the application using their existing email or social network accounts, and sync the user's profile and data with the application. This website, called Expense Tracker, is always available. All data and financial information are stored in the database tier. This proposed system should allow users to communicate with the system and store information. The user should be able to select from different categories and enter the payment amount and method. Data analysis should be possible with this technique. Provides details about the categories in which users spent the most money. Users should be able to save and track previous costs through the system's user interface. This tracking money app was created with the aim to help users easily manage all their income and expenses and track all credits and debits of transactions in different categories. Users can also visualize all transactions in different categories and download an Excel report on the transactions. App users can better manage their spending thanks to push notifications with personalized suggestions.

ADVANTAGES:

- Easy to find and reduce the unnecessary of spending money
- We can save the by using this application
- It helps to more awareness for budget making and overcome the manual process
- Provide graphical pie-chart makes easy to know the Expensive Details

3.3 Problem Solution fit

Problem-Solution fit canvas 2.0		Purpose / Vision	
Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? i.e. working parents of 0-5 y.o. kids The personal expense tracking system can be used by individual users, Working people, students, office management other small organizations etc.	6. CUSTOMER CONSTRAINTS CC 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. The User should have internet access to store data as well as to sign up or sign in to access the application with their own account.	5. AVAILABLE SOLUTIONS AS 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 One of the available solutions is tracking the expenses manually using a note and a pen, a traditional way to keep track of your expenses. But this solution is not efficient since it is a time-consuming process.
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. The main problem is to provide an optimized and efficient personal expense tracking application to the users for better management of their expenses and savings.	9. PROBLEM ROOT CAUSE RC 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. People are concerned about spending their money on waste and struggling to save some money. Keep tracking of expenses will be a greater solution to save money at all costs	7. BEHAVIOUR BE 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) The primary reason to address the problem is the user who is the retailer, who wants track their expenses anytime. The user needs to keep updating about their expenses then and there
Identify strong TR & EM	3. TRIGGERS TR What triggers customers to act? i.e. seeing their neighbour install it Seeing their friends or co-workers who efficiently saves money will trigger them to use the application	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fit 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. This project is aimed at developing a desktop-based application named Personal expense tracker for managing the expenses. This system can be used to update the daily expenses of the user and set target/goals to save money.	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 The actions taken by the user is that storing the details of the expenses immediately after spending 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 The user can save their expense entries in the local storage when the device connected to internet the data will be sent to the cloud
	4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? Using manual method will consume lot of time. But using app will be easier to track expenses anytime anywhere		

CHAPTER 4

REQUIREMENT ANALYSIS

Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Add Expenses	Enter the everyday expenses Split into categories (example : food, petrol, movies)
FR-4	Reminder mail	Sending reminder mail on target (for ex : if user wants a reminder when his/her balance reaches some amount (5000)) Sending reminder mail to the user if he/she has not filled that day's expenses.
FR-5	Creating Graphs	Graphs showing everyday and weekly expenses. Categorical graphs on expenditure.
FR-6	Add salary	Users must enter the salary at the start of the month.

Income and Expense Tracking

This should be a must-have feature. The main function of the expense management app is to assess your income and find out the expenses incurred during the day.

Capturing and Organizing Receipts

It is possible that you may miss some transactions or forget some expenses. This issue is resolved with this feature. Simply take a photo of your receipt every time you make or receive a digital or cash payment. Placing it under the right category ensures that the expense tracker doesn't miss it.

Tracking Sales You can also use

Personal Expense Tracker app to do e-commerce calculations. You can authorize credit cards by synchronizing the mobile card feature.

4.1.3 Secure Access

You can't do all the bookkeeping yourself, so you can give the app privacy and secure access to your details. In addition, special access to some expense tracking software.

Non-Functional Requirements

The middle name was a quality attribute. For example, a continuing messaging platform might require the speed at which the system performs edits to meet user expectations as a non-functional requirement.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	A simple web application which is accessible across devices. You will able to allocate money to different priorities and also help you to cut down on unnecessary spending
NFR-2	Security	It employs the latest security and technology measures to keep customers personal and financial information safe
NFR-3	Reliability	Containerized service ensures that new instance cankick up when there is a failure
NFR-4	Performance	Help to gain control of your finance, pay down debt, grow your net worth, help to upload receipts, track mileage
NFR-5	Availability	With load balancing and multiple container instances, the service is always available
NFR-6	Scalability	To know where money goes and you can ensure that money is used widely

CHAPTER 5

PROJECT DESIGN

5.1 Data flow Diagrams

A two-dimensional diagram represents how data is processed and transmitted within the system. A graphical representation identifies each data source and how it interacts with other data sources to produce common output. Data Flow Someone who wants to design her diagram identifies the external inputs and outputs, determines how the inputs and outputs relate to each other, and uses graphics to show how these connections relate to each other. You can show what you are doing. I need to explain what to do. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

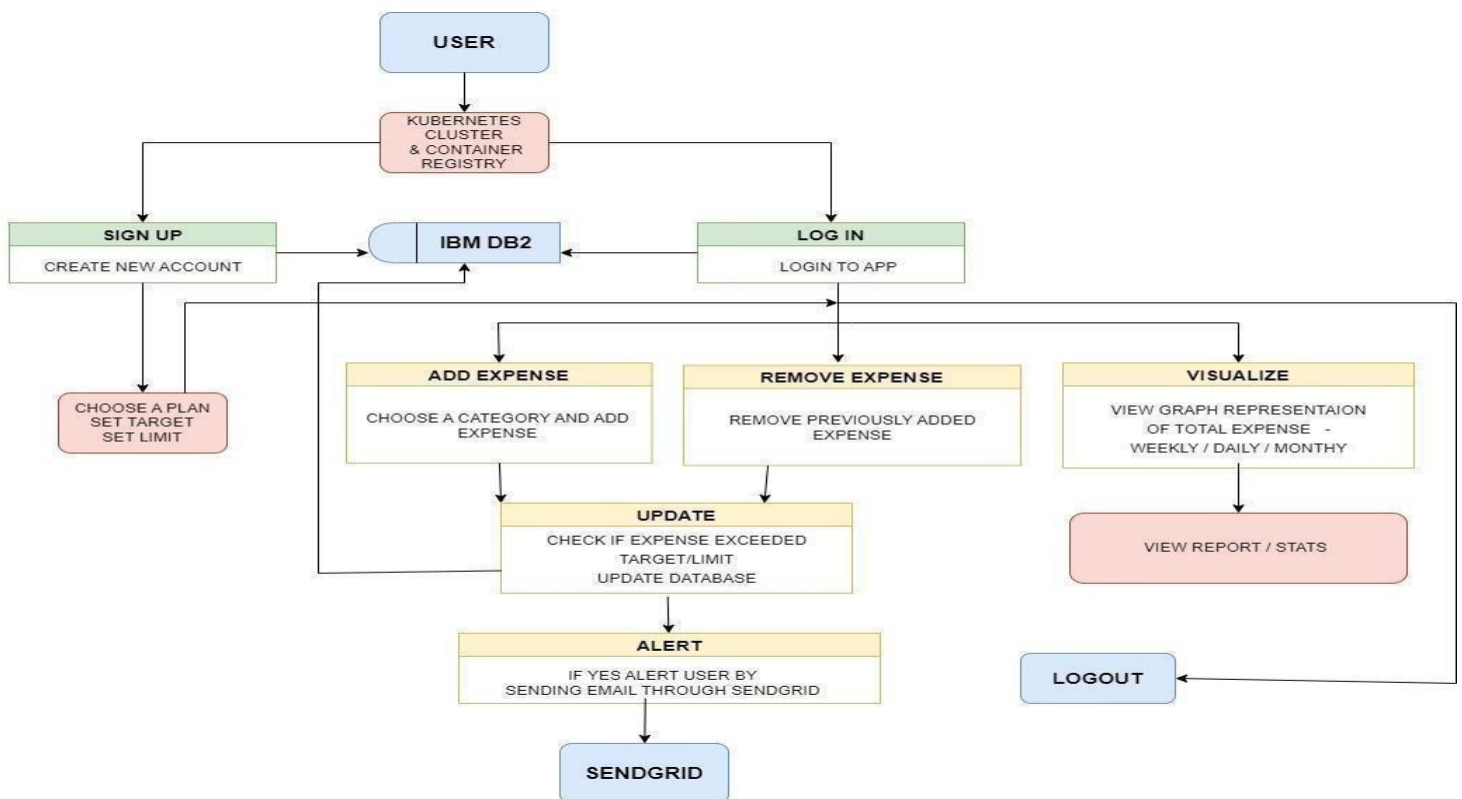


TABLE 5.1.1 DATA FLOW DIAGRAM

LEVEL 0

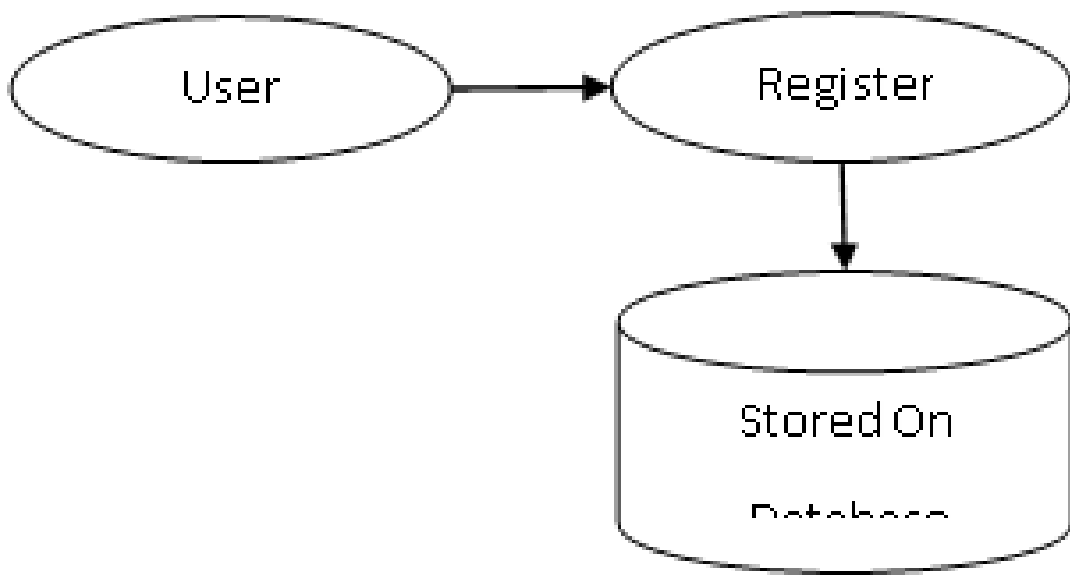


FIG 5.1.2 LEVEL 0

LEVEL 1

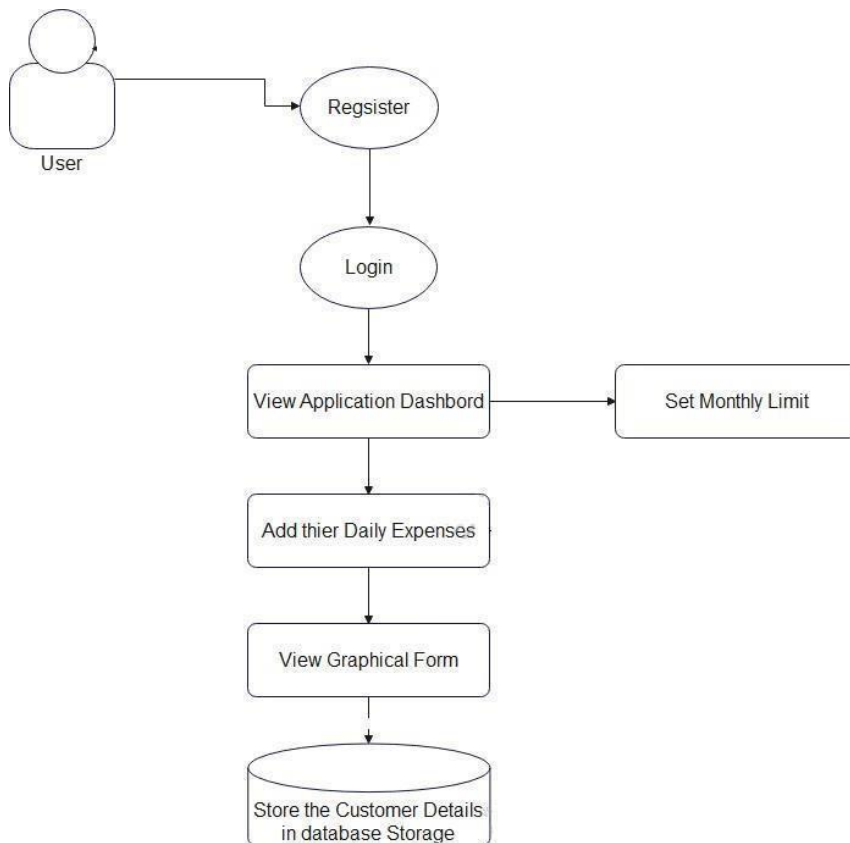


FIG 5.1.3 LEVEL 1

LEVEL 2

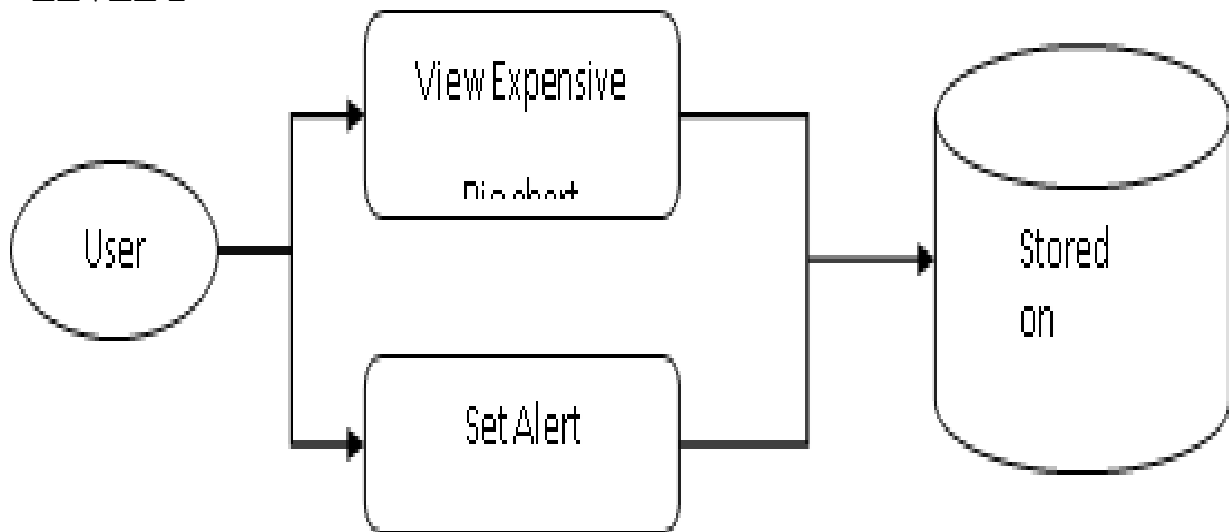
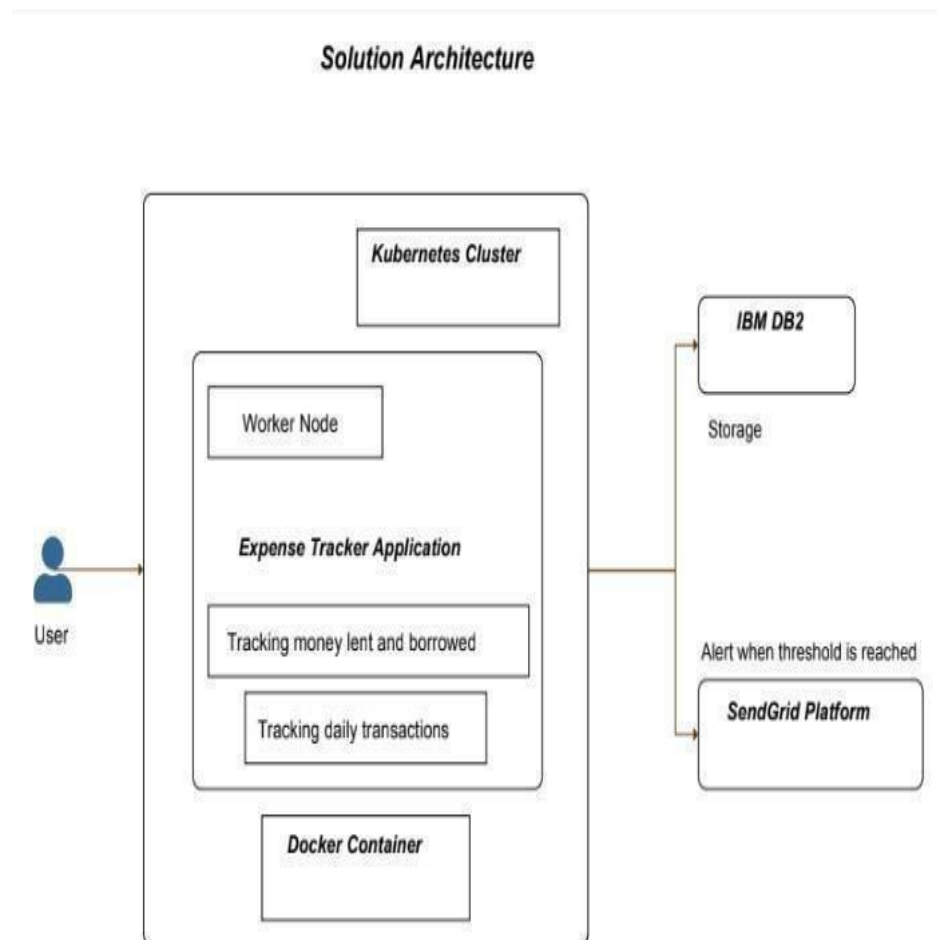


FIG 5.1.4 LEVEL 2

5.2 Solution and Technical architecture



5.2.1 Solution Architecture

5.3 User stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Akilan
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Arumugam
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Karthikeyan
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Athithiyan
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Karthikeyan
	Dashboard	USN-1	As a user, I can see my dashboard and go through the functions provided by the system.	2	High	Akilan
Sprint-2	Customer Care Executive	USN-1	Make a call to the customer care executive and rectify the queries.	1	High	Athithiyan
Sprint-3	Administrator	USN-4	Responsible for carrying out the administrative process.	2	Medium	Arumugam

CHAPTER - 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Overall Team
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Overall Team
Sprint-2		USN-3	As a user, I can register for the Application through Facebook	2	Low	Overall Team
Sprint-1		USN-4	As a user, I can register for the Application through Gmail	2	Medium	Overall Team
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Overall Team
	Dashboard	USN-1	As a user, I can see my dashboard and go through the functions provide by the system.	2	High	Overall Team
Sprint-2	Customer Care Executive	USN-1	Make a call to the customer care executive and rectify the queries.	1	High	Overall Team
Sprint-3	Administrator	USN-4	Responsible for carrying out the administration process.	2	Medium	Overall Team

6.2 Sprint Delivery Plan

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	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Chapter 7

Coding & Solution

```
from xml.etree.ElementTree import tostring
from flask import Flask, request, render_template, flash, session, redirect, url_for
import ibm_db

def connection():
    try:
        conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;PROTOCOL=TCPIP;UID=hsl71286;PWD=7idgN8Pz3PUtDrJG;SECURITY=SSL;SSLServiceCertificateDigiCertGlobalRootCA.crt;", "", "")
        print("Connected to Database")
        return conn
    except:
        print("Not Connected to Database")

app = Flask(__name__)
app.secret_key = '123'
conn = connection()
app.debug = True
app.config['SESSION_TYPE'] = 'filesystem'

@app.route('/', methods=["GET", "POST"])
def defaultpage():
    if request.method == "POST":
        try:
            sql = "INSERT INTO USER
VALUES('{}','{}','{}','{}').format(request.form['rollno'], request.form['email'],
request.form['username'],
request.form['password'])
            ibm_db.exec_immediate(conn, sql)
            flash("Successfully Registered!")
            print('registered')

            return render_template('login.html')
        except Exception as inst:
```

```

        print(inst)
        print(inst.args)
        flash("Account already exists!")
        return render_template('register.html')
    else:
        return render_template('register.html')

@app.route('/login', methods=["GET", "POST"])
def login():
    if request.method == "POST":
        username = request.form["username"]
        password = request.form["password"]
        sql = "SELECT COUNT(*) FROM USER 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)
        res = ibm_db.fetch_assoc(stmt)
        if res['1'] == 1:
            session['loggedin'] = True
            session['username'] = username
            return render_template('home.html')
        else:
            flash("Username Password is incorrect!")
            return render_template('login.html')
    else:
        return render_template('login.html')

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('username', None)
    return redirect(url_for('login'))

@app.route("/home",methods=["GET","POST"])
def render_home():
    return render_template('home.html',message="")

@app.route("/dashboard",methods=["GET","POST"])
def render_dashboard():
    return render_template('dashboard.html',message="")

@app.route("/dashboard",methods=["GET","POST"])
def dashboard():

```



```

if request.method == "POST":
    new_balance1 = request.form['new_balance']

    sql="INSERT INTO BALANCE(CURRENT_BALANCE)
VALUES(new_balance1)"
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,new_balance1)
    ibm_db.execute(stmt)

    return redirect(url_for('dashboard'))

return render_template('dashboard.html')


@app.route("/expenses")
def render_expenses():
    return render_template('expenses.html')

if __name__ == '__main__':

    app.run(debug=True)

```

CHAPTER 8

CONCLUSION AND FUTURE WORK

CONCLUSION

The majority of the shortcomings of the old system have been mostly resolved by the new one, which operates in accordance with the provided design specification. Design the project that is more effective than existing income and expense trackers. The manual calculation for determining the income and expense every month is successfully avoided by the project. The modules are created in a way that is both effective and appealing. The created methods eliminate the issue and satisfy the needs by giving accurate and thorough information. The system has complied with every criterion that the user envisaged. The newly created system takes less time to process, and all the information is updated and processed right away. Thus, we have developed such type of android application which help the users to reduces their effort of handling daily expenses. That the application will have various components of updating and viewing users' expenditures. As part of research, we considered adding certain components to the application to make it more useful to the user. Some of the extra Components are like enabling users to register to the application using existing email or social network account, it will synchronize the users profile data to the application

FUTURE ENHANCEMENT

As future work, History can be set to view all the details in the app even if the particular data is deleted from the database. Statistics could be prepared based on the Income, Expense details of the user. Sharing files via Bluetooth, WhatsApp can be allowed. Printing the details of the particular income or expense details can be made. Some of the extra components are like enabling users to register to the application using existing email or social network account, it will synchronize the users profile data to the application

APPENDIX

A1. SOFTWARE DESCRIPTION

FRONTEND: HTML and PYTHON

The common markup language for texts intended to be viewed in a web browser is called Hypertext Markup Language (HTML). Technologies like Cascading Style Sheets (CSS) and programming languages like JavaScript can help. HTML documents are downloaded from a web server or local storage by web browsers, who then turn them into multimedia web pages. HTML initially featured cues for the document's design and semantically explains the structure of a web page. The foundation of HTML pages are HTML components. Images and other objects, such as interactive forms, may be embedded within the produced page using HTML techniques. By indicating structural semantics for text like as headings, paragraphs, lists, links, and quotations, HTML offers a way to build organised texts. Tags, which are written in angle brackets, are used to distinguish HTML components. Input and image tags, for example, add content directly to the page. Other tags, like `p`, enclose and describe the content of the document. They may also include other tags as sub-elements. Browsers employ the HTML tags to decipher the page's content rather than displaying them. HTML allows for the insertion of scripts written in scripting languages like JavaScript that modify the appearance and content of web pages. CSS defines how material is presented and how it is organised. Since 1997, the World Wide Web Consortium (W3C), which previously oversaw the HTML and now oversees the CSS standards, has pushed for the use of CSS rather than explicit presentational HTML.

Python is a high-level, interpreted general-purpose programming language. Python, which was developed by Guido van Rossum and originally made available in 1991, stresses code readability and makes extensive use of whitespace. It offers building blocks that make it possible to programme clearly on both small and big dimensions. Van Rossum gave up his position as the community's leader in July 2018. Python has an autonomous memory management system and a dynamic type system. It includes a sizable and thorough standard library, supports a variety of programming paradigms, including imperative, functional, procedural, and object-oriented. There are Python interpreters for many different operating systems. The community-based development process is shared by virtually all of Python's other implementations, including CPython, the standard Python implementation. both

Python and CPython

It is especially well-liked for adding programmable interfaces to already-existing applications because of its versatility. Van Rossum's dissatisfaction with ABC, which advocated the opposite strategy, led to his concept of a tiny core language with a huge standard library and readily expandable interpreter. The Python ideology discourages flamboyant syntax (like that of Perl) in favour of a simpler, less-cluttered grammar while giving freedom in development style. "To label anything as 'clever' is not regarded a praise in the Python community," as Alex Martelli put it. Python's guiding principle is "there should be one—and preferably only one—obvious way to do it," as opposed to Perl's "there is more than one way to do it" approach to language design. The developers of Python try to prevent over-optimization and reject changes to non-critical portions of CPython that would result in slight speed improvements at the expense of readability. When speed is crucial, a Python programmer can utilise PyPy, a just-in-time compiler, or relocate time-critical functions to extension modules written in languages like C. Another option is CPython, which converts Python scripts into C and allows users to directly invoke the Python interpreter via C-level API calls.

The developers of Python prioritise keeping the language enjoyable to use. This is evident in the language's name, which pays homage to the British comedy group Monty Python, as well as in the language's somewhat lighthearted approach to tutorials and reference materials, like in the use of examples like "spam" and "eggs" (from a well-known song). In the Python community, the term "pythonic" is frequently used to refer to a variety of programming styles. Code is said to be pythonic if it adheres to Python's minimalist philosophy and emphasises readability, makes good use of Python idioms, and is natural or demonstrates proficiency in the language. Unpythonic code, on the other hand, reads like a clumsy translation from another programming language or is difficult to understand. Pythonists, Pythonistas, and Pythoneers are terms that are frequently used to refer to Python users and enthusiasts, particularly those who are seen to be competent or experienced. Python is an interpreted, object-oriented, high-level, dynamically semantic programming language.

It is very appealing due to its high-level built-in data structures, dynamic typing, and dynamic binding both for Rapid Application Development and as a scripting or glue language to join together already existing components. Python's straightforward syntax prioritises readability and makes it simple to learn, which lowers the cost of programme maintenance. Python's support for modules and packages promotes the modularity and reuse of code in programmes. For all popular systems, the Python interpreter and the comprehensive standard library are freely distributable and accessible in source or binary form. Python frequently causes programmers to fall in love due to the enhanced productivity it offers.

BACKEND:MySql

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, Word Press, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), Imagebook, Twitter, Flickr, Nokia.com, and YouTube.

Inter images

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design &

modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

SQL Server allows multiple clients to use the same database concurrently. As such, it needs to control concurrent access to shared data, to ensure data integrity—when multiple clients update the same data, or clients attempt to read data that is in the process of being changed by another client. SQL Server provides two modes of concurrency control: pessimistic concurrency and optimistic concurrency. When pessimistic concurrency control is being used, SQL Server controls concurrent access by using locks. Locks can be either shared or exclusive. Exclusive lock grants the user exclusive access to the data—no other user can access the data as long as the lock is held. Shared locks are used when some data is being read—multiple user scan read from data locked with a shared lock, but not acquire an exclusive lock. The latter would have to wait for all shared locks to be released. Locks can be applied on different levels of granularity—on entire tables, pages, or even on a per-row basis on tables. For indexes, it can either be on the entire index or on index leaves. The level of granularity to be used is defined on a per-database basis by the database administrator. While a fine-grained locking system allows more users to use the table or index simultaneously, it requires more resources, so it does not automatically yield higher performance. SQL Server also includes two more lightweight mutual exclusion solutions—latches and spinlocks—which are less robust than locks but are less resource intensive. SQL Server uses them for DMVs and other resources that are usually not busy. SQL Server also monitors all worker threads that acquire locks to ensure that they do not end up in deadlocks—in case they do, SQL Server takes remedial measures, which in many cases are to kill one of the threads entangled in a deadlock and roll back the transaction it started. To implement locking, SQL Server contains the Lock Manager. The Lock Manager maintains an in-memory table that manages the database objects and locks, if any, on them along with other metadata about the lock. Access to any shared object is mediated by the lock manager, which either grants access to the resource or blocks it. SQL Server also provides the optimistic concurrency control mechanism, which is similar to the multi version concurrency control used in other databases. Social database frameworks are the most critical database frameworks utilized as a part of the product business today. A standout amongst the most remarkable frameworks is Microsoft SQL Server. SQL Server is a database administration framework created and showcased by Microsoft. It runs solely under Windows NT and Windows 95/98.

➤ **The most critical parts of SQL Server 8 are:**

- SQL Server is anything but difficult to utilize.
- SQL Server scales from a portable tablet to symmetric multiprocessor frameworks.
- SQL Server gives information warehousing elements that as of recently have just been accessible in Oracle and other more costly DBMSs.

A database framework is a general gathering of distinctive database programming segments and databases containing the parts viz. Database application projects, Front-End segments, Database administration frameworks, and Databases.

➤ **A database framework must give the accompanying elements:**

- A mixture of client interfaces
- Physical information autonomy
- Logical information autonomy
- Query advancement
- Data honesty
- Concurrency control
- Backup and recuperation
- Security and approval

SQL Server is a Relational Database Management System. The SQL Server social dialect is called Transact-SQL. SQL is resource arranged dialect. This implies that SQL can inquiry numerous lines from one or more tables utilizing only one announcement. This component permits the utilization of this dialect at a coherently larger amount than procedural dialects. Another vital property of SQL is its non-procedurally. SQL contains two sub dialects DDL and DML.

SQL Server functions as a characteristic augmentation of Windows NT and windows 95/98. SQL Server is generally simple to oversee through the utilization of a graphical registering environment for each undertaking of framework and database organization. SQL

Server uses administrations of Windows NT to offer new or expanded database capacities, for example, sending and accepting messages and overseeing login security.

The SQL Server chairman's essential device for connecting with the framework is Enterprise Manager. The Enterprise Manager has two primary purposes: Administration of the database server and Management of database items.

- SQL Server Query Analyzer gives a graphical presentation of the execution arrangement of a question and a programmed segment that recommends which list ought to be utilized for a chose inquiry. This intelligent segment of SQL Server performs the assignments like:
- Generating and executing Transact-SQL explanations
- Putting away the produced Transact-SQL explanations in a document
- Analyzing execution gets ready for produced inquiries
- Graphically representing the execution arrangement for a chose question.

A put away method is an exceptional sort of clump written in Transact-SQL utilizing the SQL dialect and SQL augmentations. It is saved money on the database server to enhance the execution and consistency of monotonous undertakings. SQL Server backings put away methods and framework techniques. Put away techniques can be utilized for the accompanying purposes: to control access approval, to make a review trial of exercises in database tables, to discrete information definition & information control articulations concerning a database & every single comparing application.

The database article perspective can be utilized for:

- Restricting the utilization of specific sections and lines of tables - that is to control access to a specific piece of one or more tables,
- To shroud the points of interest of confounded inquiries, to limit embedded & redesigned qualities to certain extents.

A1 Appendix Coding

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Login</title>
    <link
      rel="stylesheet"
      type="text/css"
      href={{ url_for('static',filename="Style/bootstrap.css") }}>
    <link
      rel="stylesheet"
      type="text/css"
      href={{ url_for('static',filename="Style/fontawesome.css") }}
    />
    <link
      rel="stylesheet"
      type="text/css"
      href={{ url_for('static',filename="Style/iofrm.css") }}
    />
    <link
      rel="stylesheet"
      type="text/css"
      href={{ url_for('static',filename="Style/iofrm-theme4.css") }}
    />
    <script>
      function loadRegisterPage() {
        window.location.assign("http://localhost:5000/");
      }
    </script>
  </head>
  <body>
    <div class="form-body">
```

```

<div class="row">
  <div class="img-holder">
    <div class="bg"></div>
    <div class="info-holder">
      
    </div>
  </div>
  <div class="form-holder">
    <div class="form-content">
      <div class="form-items">
        { % for message in get_flashed_messages() % }
        <div class="message-flash">
          <strong>{{ message }}</strong>
        </div>
        { % endfor % }
      <div class="page-links">
        <a href="http://localhost:5000/login" class="active">Login</a>
        ><a href="http://localhost:5000/">Register</a>
      </div>
      <form
        action="http://localhost:5000/login"
        method="post"
        id="studentform"
      >
        <input
          class="form-control"
          type="text"
          name="username"
          placeholder="Username"
          required
        />
        <input
          class="form-control"
          type="password"
          name="password"
          placeholder="Password"
          required

```



```

    type="text/css"
    href={ { url_for('static',filename="Style/fontawesome.css") } }
/>
<link
    rel="stylesheet"
    type="text/css"
    href={ { url_for('static',filename="Style/iofrm.css") } }
/>
<link
    rel="stylesheet"
    type="text/css"
    href={ { url_for('static',filename="Style/iofrm-theme4.css") } }
/>
<script>
    function loadLoginPage() {
        window.location.assign("http://localhost:5000/login");
    }
</script>
</head>
<body>
    <div class="form-body">
        <div class="row">
            <div class="img-holder">
                <div class="bg"></div>
                <div class="info-holder">
                    <img src={ { url_for('static',filename="images/graphic1.svg") } } alt="" />
                </div>
            </div>
            <div class="form-holder">
                <div class="form-content">
                    <div class="form-items">
                        <div class="page-links">
                            <a href="http://localhost:5000/login">Login</a>
                            ><a href="http://localhost:5000/" class="active">Register</a>
                        </div>
                        {% for message in get_flashed_messages() %}
                        <div class="message-flash">

```

```

    <strong>{ { message } }</strong>
</div>
{ % endfor % }
<form
  action="http://localhost:5000/"
  method="post"
  id="studentform"
>
<input
  class="form-control"
  type="text"
  name="rollno"
  placeholder="Enter the rollnumber"
  required
/>

<input
  class="form-control"
  type="email"
  name="email"
  placeholder="E-mail"
  required
/>
<input
  class="form-control"
  type="text"
  name="username"
  placeholder="UserName"
  required
/>
<input
  class="form-control"
  type="password"
  name="password"
  placeholder="Password"
  required
/>

```

```

    <div class="form-button">
      <button
        id="submit"
        type="submit"
        class="ibtn"
        onclick="loadLoginPage();"
      >
        Register
      </button>
    </div>
  </form>
</div>
</div>
</div>
</div>
</div>
<script src="/static/js/jquery.min.js"></script>
<script src="/static/js/popper.min.js"></script>
<script src="/static/js/bootstrap.min.js"></script>
<script src="/static/js/main.js"></script>
</body>
</html>

```

```

<!DOCTYPE html>

```

```

<html>

```

```

<head>

```

```

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

```

```

<!-- CSS only -->

```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"

```

```

integrity="sha384-

```

```

Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">

```

```

<!-- JavaScript Bundle with Popper -->

```

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-

```

```
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
<link href={ { url_for('static',filename="/stylesheets/styles.css" )} } rel="stylesheet">
```

```
<style>
.main {
margin-left: 300px; /* Same as the width of the sidenav */
font-size: 28px; /* Increased text to enable scrolling */
padding: 0px 10px;

}
.container{
margin-left: 300px;

}
.sidenav a {
text-decoration: none;
color: #818181;
display: block;
}
.sidenav a:hover {
color: #ff0000;
}
```

```
#mySidenav a {
position: absolute;
left: -60px;
margin-top:50px;
transition: 0.3s;
padding: 15px;
width: 100px;
text-decoration: none;
font-size: 20px;
color: white;
```

```

    border-radius: 0 5px 5px 0;
}
.fs-4{
    color: white;
}
#mySidenav a:hover {
    left: 0;
    right: 100;
}

#home {
    top: 20px;
    background-color: #04AA6D;
}

#dashboard {
    top: 80px;
    background-color: #2196F3;
}

#expenses {
    top: 140px;
    background-color: #f44336;
}

#product {
    top: 200px;
    background-color: #555
}
</style>
<script>
    function loadLoginPage() {
        window.location.assign("http://localhost:5000/logout");
    }
</script>
</head>
<body>

```



```

<nav class="navbar bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">
      <img src={{ url_for('static', filename="logo.png" )}} alt="Logo" width="30" height="24"
        class="d-inline-block align-text-top">
        PERSONAL EXPENSES TRACKER APPLICATION
      </a>
      <button type="button" class="btn btn-primary accountButton" onclick="loadLoginPage();"
id="logoutbutton">Logout</button>
    </div>
  </nav>

```

```

<div class="d-flex flex-column flex-shrink-0 p-3 text-white bg-dark sidenav" style="width:
280px; position: fixed; height: 95%; top: 1; bottom: 5; left: 0;">
  <a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-white text-
decoration-none">
    <svg class="bi me-2" width="40" height="32"><use
xlink:href="#bootstrap"></use></svg>
    <span class="fs-4">MENU</span>
  </a>
  <hr>

```

```

<ul class="nav nav-pills flex-column mb-auto">
  <li class="nav-item">
<div id="mySidenav" class="sidenav">
  <a href="/home" id="home">HOME</a>
  <a href="/dashboard" id="dashboard">DASHBOARD</a>
  <a href="/expenses" id="expenses">EXPENSES</a>
  <a href="/product" id="product">PRODUCT</a>
</div>
</ul>
<hr>

```

```

  <div class="dropdown">
    <a href="#" class="d-flex align-items-center text-white text-decoration-none dropdown-
toggle" id="dropdownUser1" data-bs-toggle="dropdown" aria-expanded="false">
      

```

```

    <strong>mdo</strong>
  </a>
  <ul class="dropdown-menu dropdown-menu-dark text-small shadow" aria-
labelledby="dropdownUser1">
    <li><a class="dropdown-item" href="#">New project...</a></li>
    <li><a class="dropdown-item" href="#">Settings</a></li>
    <li><a class="dropdown-item" href="#">Profile</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" href="#">Sign out</a></li>
  </ul>
</div>
</div>

<div style="margin-left:50px; margin-top:10px; margin-right: 0px; position: fixed;">
  <center><h1>Welcome to the Home, {{session.username}}!</h1></center>
  <div class="main">
    <h2>Introduction About Our Application</h2>
    <p>If you want to track your expenses in a simple and straightforward way, then all you
need to do is collect your receipts once a month, organize them and then run your system.
Other expense tracking systems could provide you with a little more information (listing
expenses in a spreadsheet, using money management software, or even using an online
application), however all methods have one thing in common: you have to get in the habit of
thinking about your expenses from the beginning. Misplacing a receipt or forgetting to keep
track of the cash you have spent is one of the most common mistakes people make. It is
important to keep track of even the most small expenses, such as coffee or trips to the vending
machine, as they can mount up very quickly, even though you might not think it is necessary
to do so. There are all sorts of opportunities to throw a kink in your plan to track expenses.
You have to get in the habit of doing so, to reduce those lapses, and make sure that the data
you're basing financial decisions on is solid.</p>
  </div>
</div>

<!--Footer Code Begins-->
<div class="container" style="margin-top:610px; position: fixed;">
  <footer class="d-flex flex-wrap justify-content-between align-items-center py-3 my-4
border-top">
    <div class="col-md-4 d-flex align-items-center">
      <a href="/" class="mb-3 me-2 mb-md-0 text-muted text-decoration-none lh-1">
        <svg class="bi" width="30" height="24">
          <use xlink:href="#bootstrap"></use>

```

```

    </svg>
  </a>
  <span class="mb-3 mb-md-0 text-muted">© 2022 Company, Inc</span>
</div>

<ul class="nav col-md-4 justify-content-end list-unstyled d-flex">
  <li class="ms-3"><a class="text-muted" href="#"><svg class="bi" width="24"
height="24">
    <use xlink:href="#twitter"></use>
  </svg></a></li>
  <li class="ms-3"><a class="text-muted" href="#"><svg class="bi" width="24"
height="24">
    <use xlink:href="#instagram"></use>
  </svg></a></li>
  <li class="ms-3"><a class="text-muted" href="#"><svg class="bi" width="24"
height="24">
    <use xlink:href="#facebook"></use>
  </svg></a></li>
</ul>
</footer>
</div>
<!--Footer Code Ends-->
</body>
</html>

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"

```

```
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
<link href={ { url_for('static',filename="/stylesheets/styles.css" ) } } rel="stylesheet">
```

```
<style>
.main {
margin-left: 300px; /* Same as the width of the sidenav */
font-size: 28px; /* Increased text to enable scrolling */
padding: 0px 10px;

}
.container{
margin-left: 300px;

}
.sidenav a {
text-decoration: none;
color: #818181;
display: block;
}
.sidenav a:hover {
color: #ff0000;
}
```

```
#mySidenav a {
position: absolute;
left: -60px;
margin-top:50px;
transition: 0.3s;
padding: 15px;
width: 100px;
text-decoration: none;
font-size: 20px;
```

```

    color: white;
    border-radius: 0 5px 5px 0;
}
.fs-4{
    color: white;
}
#mySidenav a:hover {
    left: 0;
    right: 100;
}

#home {
    top: 20px;
    background-color: #04AA6D;
}

#dashboard {
    top: 80px;
    background-color: #2196F3;
}

#expenses {
    top: 140px;
    background-color: #f44336;
}

#product {
    top: 200px;
    background-color: #555
}
</style>
<script>
    function loadDashboardPage() {
        window.location.assign("http://localhost:5000/dashboard");
    }
</script>
</head>

```

```

<body>

<nav class="navbar bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">
      <img src={{ url_for('static', filename="logo.png" )}} alt="Logo" width="30" height="24"
      class="d-inline-block align-text-top">
        PERSONAL EXPENSES TRACKER APPLICATION
    </a>
    <button type="button" class="btn btn-primary accountButton" onclick="loadLoginPage();"
id="logoutbutton">Log out</button>
  </div>
</nav>

<div class="d-flex flex-column flex-shrink-0 p-3 text-white bg-dark sidenav" style="width:
280px; position: fixed; height: 95%; top: 1; bottom: 5; left: 0;">
  <a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-white text-
decoration-none">
    <svg class="bi me-2" width="40" height="32"><use
xlink:href="#bootstrap"></use></svg>
    <span class="fs-4">MENU</span>
  </a>
  <hr>

<ul class="nav nav-pills flex-column mb-auto">
  <li class="nav-item">
<div id="mySidenav" class="sidenav">
  <a href="/home" id="home">HOME</a>
  <a href="/dashboard" id="dashboard">DASHBOARD</a>
  <a href="/expenses" id="expenses">EXPENSES</a>
  <a href="/product" id="product">PRODUCT</a>
</div>
</ul>
<hr>
<div class="dropdown">
  <a href="#" class="d-flex align-items-center text-white text-decoration-none dropdown-
toggle" id="dropdownUser1" data-bs-toggle="dropdown" aria-expanded="false">
    
    <strong>mdo</strong>
</a>
<ul class="dropdown-menu dropdown-menu-dark text-small shadow" aria-
labelledby="dropdownUser1">
    <li><a class="dropdown-item" href="#">New project...</a></li>
    <li><a class="dropdown-item" href="#">Settings</a></li>
    <li><a class="dropdown-item" href="#">Profile</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" href="#">Sign out</a></li>
</ul>
</div>
</div>

<div style="margin-left:300px; margin-top:10px; margin-right: 300px; position: fixed;">
    <center><h1>Welcome to the Wallet, {{session.username}}!</h1></center>
    <div class="main">
        </div>
</div>
</div>
<form method="POST">
<div class="col py-3" style="background-color:#00AD83">
    <h3 style="color:white; text-align: center;">Update Balance</h3>
    <div class="container mt-3" style="width: 600px;">
        <div class="card shadow-lg bg-white rounded">
            <form action="/Updatebalance" method="POST">
                <div class="card-header" style="text-align: center;">
                    <span style="display:inline-flex"><h4>Wallet Balance</h4></span>
                </div>
                <div class="card-body">
                    <div class="mb-3">
                        <label for="category" class="form-label">Current Balance: </label>
                        <input type="text" value="{{ session.new_balance }}" readonly>
                    </div>
                    <div class="mb-3">
                        <label for="description" class="form-label">new balance: </label>
                        <input type="text" class="form-control" id="new_balance"></input>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

    </div>
    <div class="card-footer text-muted" style="text-align:center">
        <button type="button" style="background-color:#00AD83; border-color:#00AD83;
border-radius:5px;" onclick="loadDashboardPage()";>Update Balance</button>
    </div>
</form>
</div>
</div>
</div>
</div>
</form>

```

```

<!--Footer Code Begins-->
<div class="container" style="margin-top:610px; position: fixed;">
    <footer class="d-flex flex-wrap justify-content-between align-items-center py-3 my-4
border-top">
        <div class="col-md-4 d-flex align-items-center">
            <a href="/" class="mb-3 me-2 mb-md-0 text-muted text-decoration-none lh-1">
                <svg class="bi" width="30" height="24">
                    <use xlink:href="#bootstrap"></use>
                </svg>
            </a>
            <span class="mb-3 mb-md-0 text-muted">© 2022 Company, Inc</span>
        </div>

```

```

<ul class="nav col-md-4 justify-content-end list-unstyled d-flex">
    <li class="ms-3"><a class="text-muted" href="#"><svg class="bi" width="24"
height="24">
        <use xlink:href="#twitter"></use>
    </svg></a></li>
    <li class="ms-3"><a class="text-muted" href="#"><svg class="bi" width="24"
height="24">
        <use xlink:href="#instagram"></use>
    </svg></a></li>
    <li class="ms-3"><a class="text-muted" href="#"><svg class="bi" width="24"
height="24">
        <use xlink:href="#facebook"></use>

```



```

        </svg></a></li>
    </ul>
</footer>
</div>
<!--Footer Code Ends-->
</body>
</html>

<!doctype html>
<html lang="en">
    <head>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <!-- Bootstrap 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">

        <title>Add Expense</title>
        <style>
            .main {
margin-left: 300px; /* Same as the width of the sidenav */
font-size: 28px; /* Increased text to enable scrolling */
padding: 0px 10px;

}
.container{
margin-left: 300px;

}
.sidenav a {
text-decoration: none;
color: #818181;
display: block;
}

```

```
.sidenav a:hover {  
  color: #ff0000;  
}
```

```
#mySidenav a {  
  position: absolute;  
  left: -60px;  
  margin-top: 50px;  
  transition: 0.3s;  
  padding: 15px;  
  width: 100px;  
  text-decoration: none;  
  font-size: 20px;  
  color: white;  
  border-radius: 0 5px 5px 0;  
}
```

```
.fs-4{  
  color: white;  
}
```

```
#mySidenav a:hover {  
  left: 0;  
  right: 100;  
}
```

```
#home {  
  top: 20px;  
  background-color: #04AA6D;  
}
```

```
#dashboard {  
  top: 80px;  
  background-color: #2196F3;  
}
```

```
#expenses {  
  top: 140px;
```

```

background-color: #f44336;
}

#product {
  top: 200px;
  background-color: #555
}

</style>
</head>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
<script>
  function addGroup(e) {
    // e.preventDefault();
    group = window.prompt('Enter group name: ');
    console.log('PROMPT WINDOW SHOWN'+group);

    const formData = new FormData();
    formData.append("groupname", group);

    const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
      if (this.readyState == 4 && this.status == 200) {
        var groupid= JSON.parse(this.responseText);
        console.log(groupid);
        // create option using DOM
        const newOption = document.createElement('option');
        const optionText = document.createTextNode(groupid['groupname']);
        newOption.appendChild(optionText);
        newOption.setAttribute('value',groupid['groupID']);
        const selectDropdown = document.getElementById('group');
        selectDropdown.appendChild(newOption);
        console.log('GROUPID :'+ groupid['groupID']);
      }
    }
    xhttp.open("POST", "http://localhost:5000/addgroup");
  }
}

```

```

        xhttp.send(formData);
    }
</script>
<body>

    <nav class="navbar bg-light">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">
                <img src={{ url_for('static', filename="logo.png" )}} alt="Logo" width="30"
height="24"
                class="d-inline-block align-text-top">
                    PERSONAL EXPENSES TRACKER APPLICATION
            </a>
            <button type="button" class="btn btn-primary accountButton"
onclick="loadLoginPage();" id="logoutbutton">Log out</button>
        </div>
    </nav>

    <div class="d-flex flex-column flex-shrink-0 p-3 text-white bg-dark sidenav"
style="width: 280px; position: fixed; height: 95%; top: 1; bottom: 5; left: 0;">
        <a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-white text-
decoration-none">
            <svg class="bi me-2" width="40" height="32"><use
xlink:href="#bootstrap"></use></svg>
            <span class="fs-4">MENU</span>
        </a>
        <hr>

        <ul class="nav nav-pills flex-column mb-auto">
            <li class="nav-item">
                <div id="mySidenav" class="sidenav">
                    <a href="/home" id="home">HOME</a>
                    <a href="/dashboard" id="dashboard">DASHBOARD</a>
                    <a href="/expenses" id="expenses">EXPENSES</a>
                    <a href="/product" id="product">PRODUCT</a>
                </div>
            </li>
        </ul>
        <hr>

```

```

<div class="dropdown">
  <a href="#" class="d-flex align-items-center text-white text-decoration-none
dropdown-toggle" id="dropdownUser1" data-bs-toggle="dropdown" aria-expanded="false">
    
    <strong>mdo</strong>
  </a>
  <ul class="dropdown-menu dropdown-menu-dark text-small shadow" aria-
labelledby="dropdownUser1">
    <li><a class="dropdown-item" href="#">New project...</a></li>
    <li><a class="dropdown-item" href="#">Settings</a></li>
    <li><a class="dropdown-item" href="#">Profile</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" href="#">Sign out</a></li>
  </ul>
</div>
</div>

```

```

<div class="col py-3" style="background-color:#00AD83">
  <h3 style="color:white; text-align: center;">Add expense</h3>
  <div class="container mt-3" style="width: 600px;">
    <div class="card shadow-lg bg-white rounded">
      <form action="/addexpense" method="POST">
        <div class="card-header" style="text-align: center;">
          <span style="display:inline-flex"><h4>Expense Made</h4></span>
        </div>
        <div class="card-body">
          <div class="mb-3">
            <label for="amountspent" class="form-label">Amount Spent: (Rs)
          </label>
            <input type="number" class="form-control" name="amountspent"
id="amountspent" placeholder="100.00" required>
          </div>
          <div class="mb-3">
            <label for="expensecategory" class="form-label">Expense Category:
          </label>
            <select name="category" id="category" class="form-control"
placeholder="Select a category">

```

```

        <option value="">Select a category</option>
        {% for cat in categories %}
            <option value="{{ cat[0] }}">{{ cat[1] }}</option>
        {% endfor %}
    </select>
</div>
<div class="mb-3">
    <label for="date" class="form-label">Date of Expense: </label>
    <input type="date" class="form-control" name="date" id="date"
required></input>
</div>
<div class="mb-3">
    <label for="description" class="form-label">Description of Expense:
</label>
    <input type="text" class="form-control" name="description"
id="description"></input>
</div>
<div class="mb-3">
    <label for="group" class="form-label">Group(if needed): </label>
    <div title="New group" style="float:right" value="Create group"
onclick="addGroup()">ADD GROUP</div><br/>

    <select name="group" id="group" class="form-control">
        <option value="">Select existing group</option>
        {% for group in groups %}
            <option value="{{ group[0] }}">{{ group[1] }}</option>
        {% endfor %}
    </select>
</div>
</div>
<div class="card-footer text-muted" style="text-align:center">
    <button type="submit" value="submit" style="background-color:#00AD83;
border-color:#00AD83; border-radius:5px;">Submit Expense</button>
</div>
</form>
</div>
</div>
</div>

```

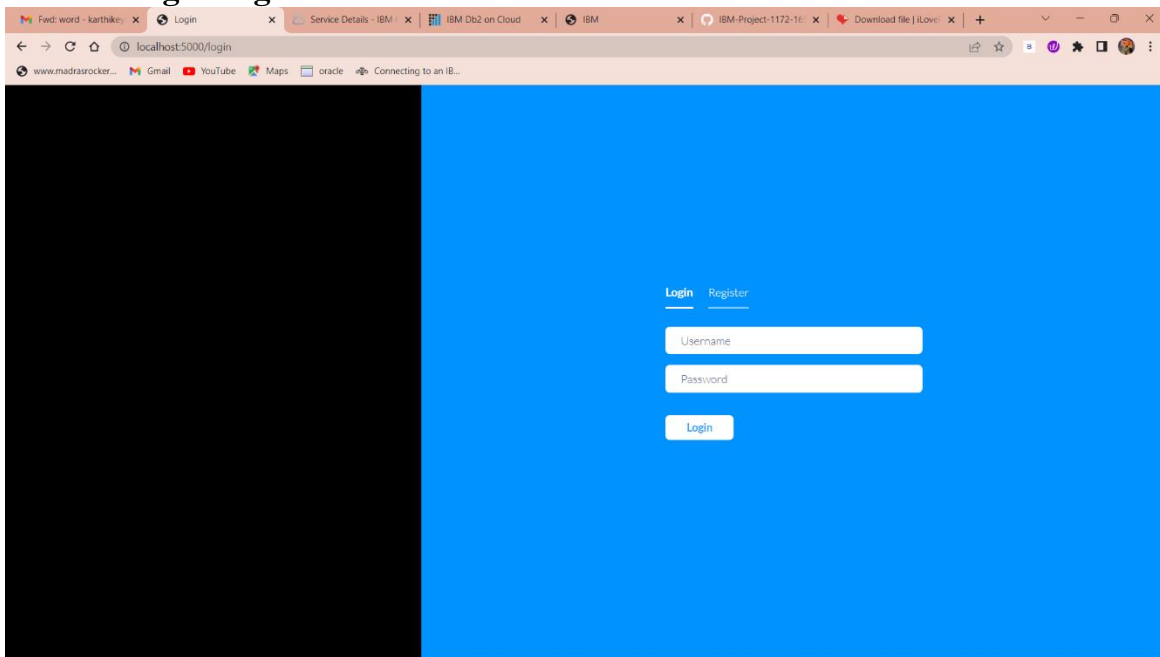
```

</div>
</div>
</body>
<script>
    document.querySelector('#date').valueAsDate = new Date();
</script>
</html>

```

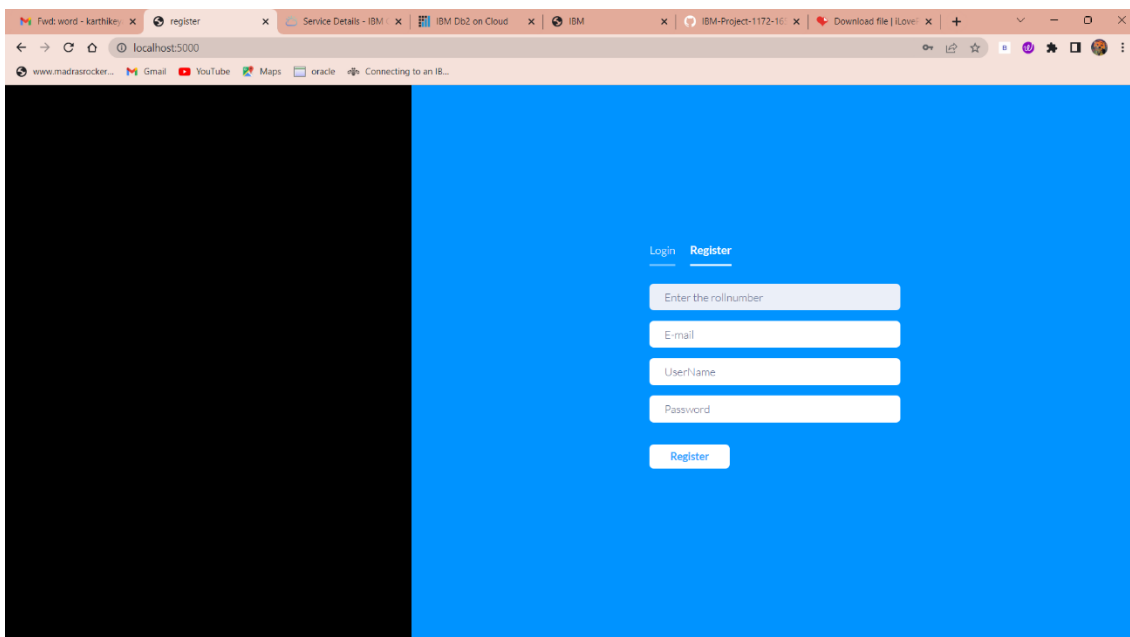
A3. SAMPLE OUTPUT

Login Page

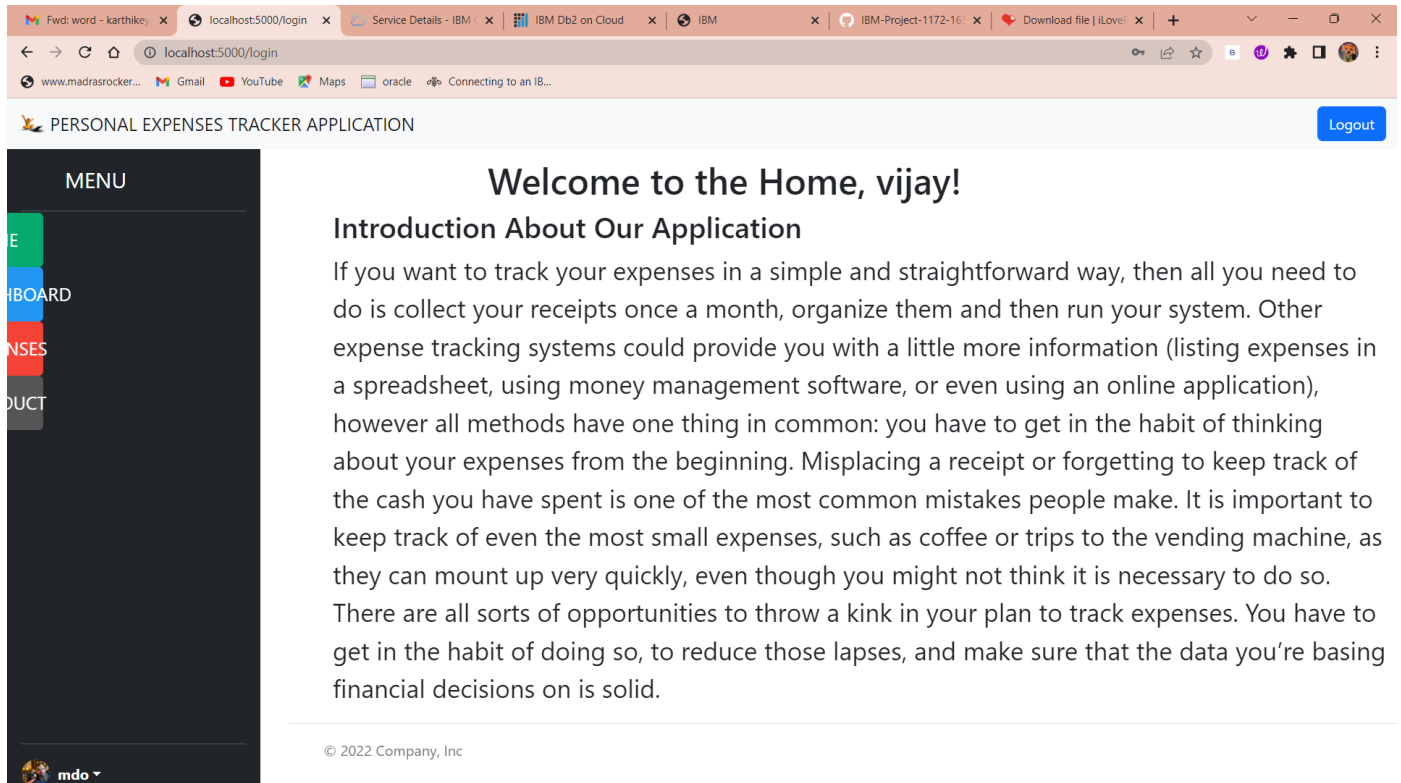


The screenshot shows a web browser window with the address bar displaying 'localhost:5000/login'. The page has a blue background on the right and a black background on the left. The login form is centered on the blue background and includes a 'Login' tab (selected) and a 'Register' tab. The form contains three input fields: 'Username', 'Password', and a 'Login' button.

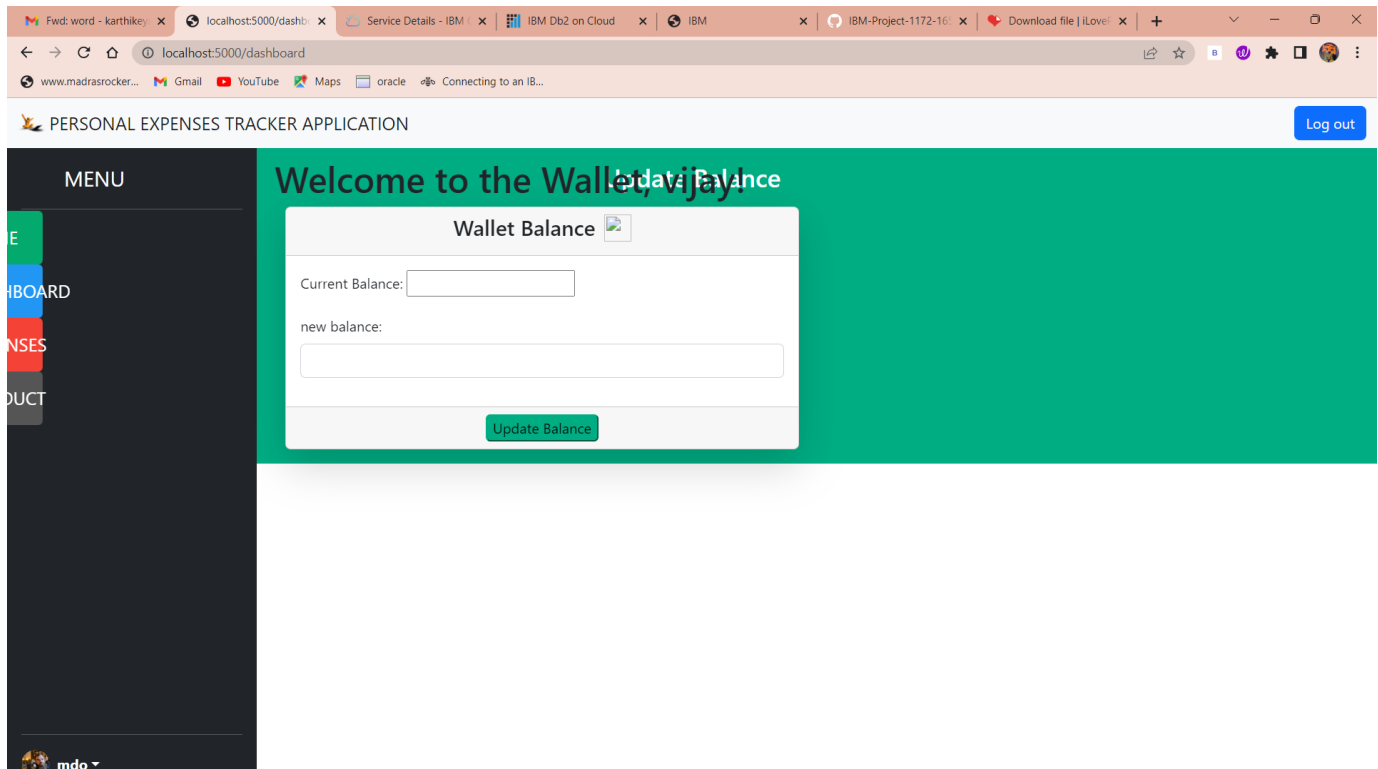
Register Page



The screenshot shows a web browser window with the address bar displaying 'localhost:5000'. The page has a blue background on the right and a black background on the left. The register form is centered on the blue background and includes a 'Login' tab and a 'Register' tab (selected). The form contains four input fields: 'Enter the rollnumber', 'E-mail', 'UserName', and 'Password', followed by a 'Register' button.



Wallet Page



REFERENCES

1. Velmurugan A, Albert Mayan J, Mohana Prasad R ,Yovan Felix A,"Implementing Health Care Center using Hadoop for Analysis and Prediction of Diseases", International Journal of Engineering and Advanced Technology (IJEAT),Vol.8, No.6, pp.3456-3459,2019.
2. <https://www.groupon.com/>
3. Eswar V.O.S, Vinil.B, Ankayarkanni B and Albert,"Integrated Collective Node Behavior Analysis with Onion Protocol for Best and Secured Data Transmission," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2018, pp. 0918-0921.
4. Mayan, J.A., Priya, K.L.," Novel approach to reuse unused test cases in a GUI based application", IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2015
5. Mohan Prasad K, Sai Nagendra Goru Rajeev, Vamsi Desu, Albert Mayan M.J , "Automated Payroll Using GPS Tracking and Image Capture ", IOP ConferenceSeries: Materials Science and Engineering, Vol.590, 012026 ,pp.1-6,2019 doi:10.1088/1757-899X/590/1/01202
6. Mary Posonia A, S. Vigneshwari, Albert Mayan J, D. Jamunarani,"Service Direct : Platform that Incorporates Service Providers and Consumers Directly",International Journal of Engineering and Advanced Technology (IJEAT) , Vol.8 ,No.6,pp.3301- 3304,2019.