TEAM ID - PNT2022TMID00673

PERSONAL EXPENSE TRACKER APPLICATION

A PROJECT REPORT

Submitted

by

SHIRLY.N [211419104250]

RATHIKA.C[211419104218]

SHANGEETHA .J [211419104246]

SUJITHRA.J [211419104272]

TEAM ID:PNT2022TMID00673

INDUSTRY MENTOR: Kusboo

FACULTY MENTOR: Maheswari.M

TEAM ID - PNT2022TMID00673

TABLE OF CONTENT

TEAM ID - PNT2022TMID00673

TEAM ID - PNT2022TMID00673

	4.1 Functional requirement	
	4.2 Non-Functional requirements	
5	**PROJECT DESIGN**	**15**

TEAM ID - PNT2022TMID00673

	5.1 Data Flow Diagrams	
	5.2 Solution & Technical Architecture	

TEAM ID - PNT2022TMID00673

1.Introduction

Personal finance entails all the financial decisions and activities that a finance app makes your life easier by helping you to manage your finances efficiently. A personal finance app will not only help you with budgeting and accounting but also give youhelpful insights about money management. Personal finance applications will ask usersto add their expenses and based on their expenses wallet balance will be updatedwhich will be visible to the user. Also, users can get an analysis of their expenditure ingraphical forms. They have an option to set a limit for the amount to be used for that particular month if the limit is exceeded the user will be notified with an email alert.

1.1 Project Overview:

Expenses Tracker is a way that can help us to keep up with our spending. Not only that, it can help us pinpoint areas that we have been spending and track upcoming bill payments.It is a web-based system that can keep track of their expenses and determine whether they are spending as per their set budget. Potential users need to input the required data such as the expense amount, merchant, category, and date when the expense was made. This mobile system is a full detailed expense tracker tool that will not only help users keep a check on their expenses, but also cut down the unrequired expenses, and thus will help provide a responsible lifestyle.

1.2 Purpose:

As of right now, there are many budget planner programmes online, but not all of them are effective at assisting users in actually creating and adhering to a budget. One of the negatives is the ongoing upkeep; several budgeting programmes provide the ease of linking with all users; bank accounts and condensing their activity into a single dashboard. However, a lot of the software that is now available has cumbersome, inefficient features. Additionally, because of their busy and chaotic lifestyles, people often neglect their budgets and wind up overspending since they didn’t carefully plan their spending. Last but not least, the user is unable to foresee future costs. While they can write down their expenses in a piece of paper or manage them in excel spreadsheet, their lack of knowledge in managing finances will be a problem. excel spreadsheet, their lack of financial management experience will be an issue.

2.Literature Survey

2.1 Existing Problem:

When someone receives their salary under the current system, they can add it to their daily cost manager. The user can then utilise the cost manager to store all the information after adding their salary information. The Expense Tracker allows the user to obtain all credit and debit detail if desired. After the salary is adjusted, the notification manager additionally reminds about the credit and debit details.

2.2 References:

[1] http://expense-manager.com/how-expense software/
[2] https://www.splitwise.com/terms
[3] http://code.google.com/p/socialauthandroid/wiki/Facebook
[4] http://code.google.com/p/socialauth-android
[5] https://ijirt.org/master/publishedpaper/IJIRT150860_PAPER.pdf
[6] http://www.appbrain.com/app/expensemanager/
com.expensemanager
[7] http://dspace.daffodilvarsity.edu.bd:8080/handle/123456789/4026
[8] http://expense-manager.com/how-expense software/
[9] Donn Felker, "Android Application Development for Dummies", published by For
Dummies, 2010.
[10] https://www.irjet.net/archives/V6/i3/IRJET-V6I31110.pdf .
[11]https://www.proquest.com/openview/a372033c3cafa03eaa5e18e68e99c86b/1.pdf?p
q-origsite=gscholar&cbl=2045096

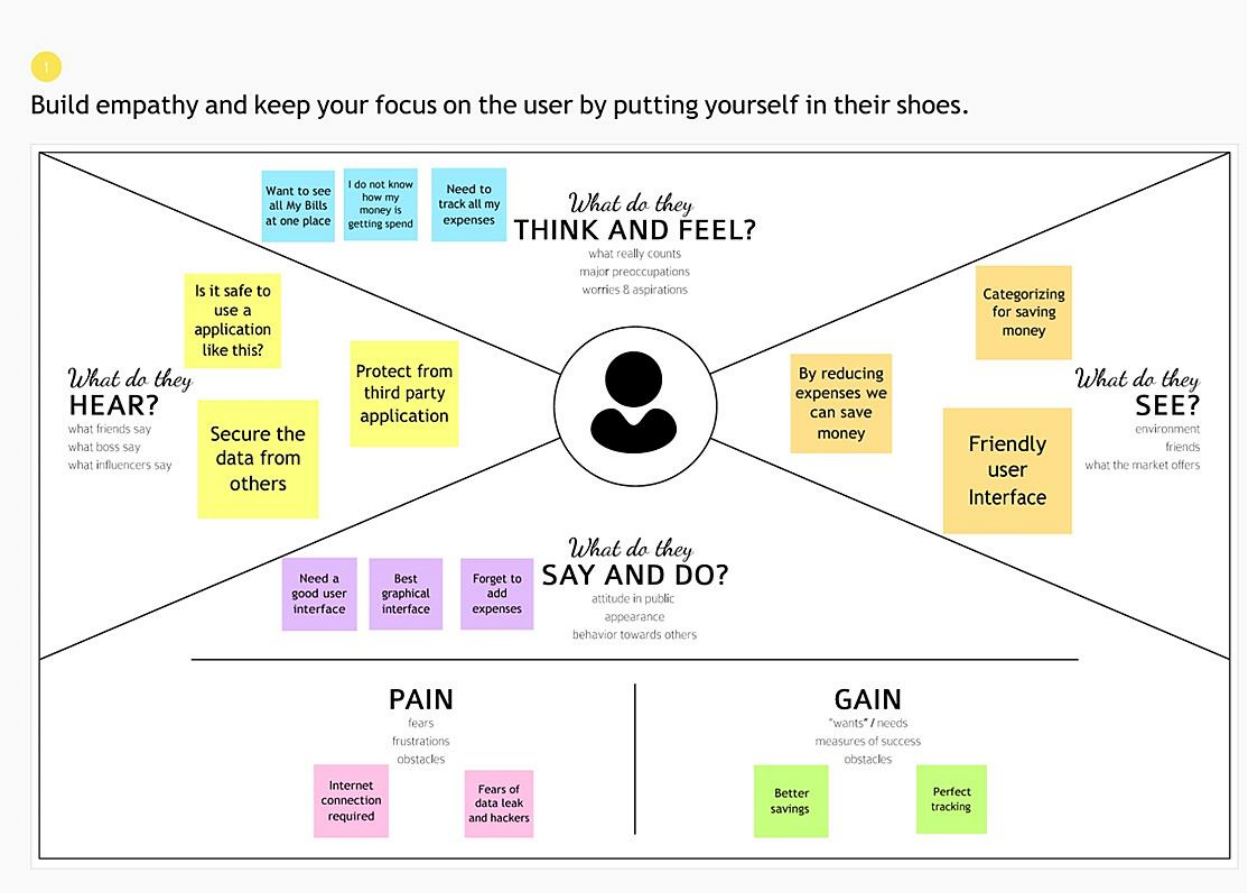
2.3 Problem statement defnition:

In simple words, personal finance entails all the financial decisions and activities that a Finance app makes your life easier by helping you to manage your finances efficiently. A personal finance app will not only help you with budgeting and accounting but also give you helpful \insights about money management. Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updated which will be visible to the user. Also, users can get an analysis of their expenditure in graphical forms. They have an option to set a limit for the amount to be used for that particular month if the limit is exceeded the user will be notified with an email alert.

3.Ideation and Proposed Solution

3.1 Empathy map canvas:

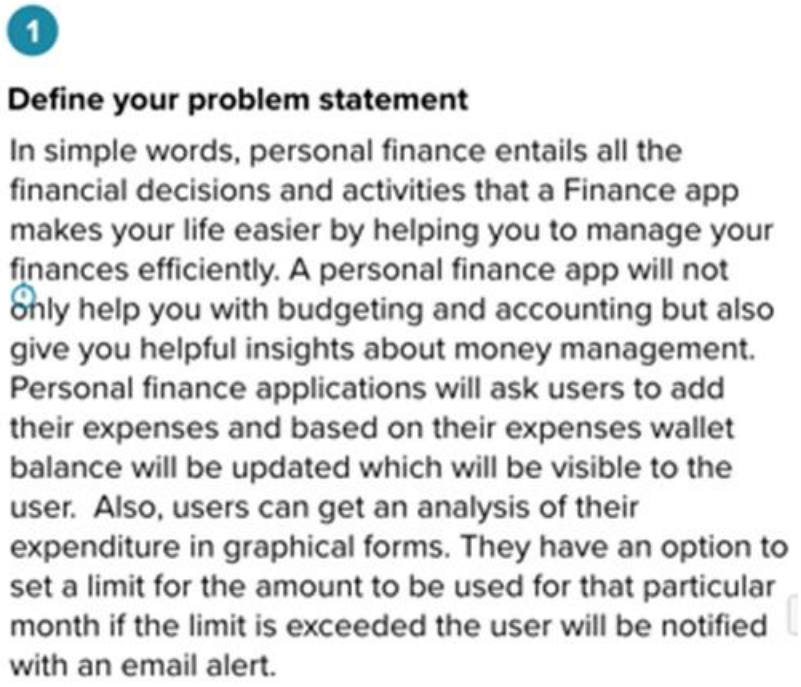
An empathy map is a straightforward, simple-to-understand picture that summarises information about a user's actions and views. It is a helpful tool that enables teams to comprehend their users more fully. It's important to comprehend both the actual issue and the individual who is experiencing it in order to develop a workable solution. Participants learn to think about situations from the user's perspective, including goals and problems, through the exercise of constructing the map.



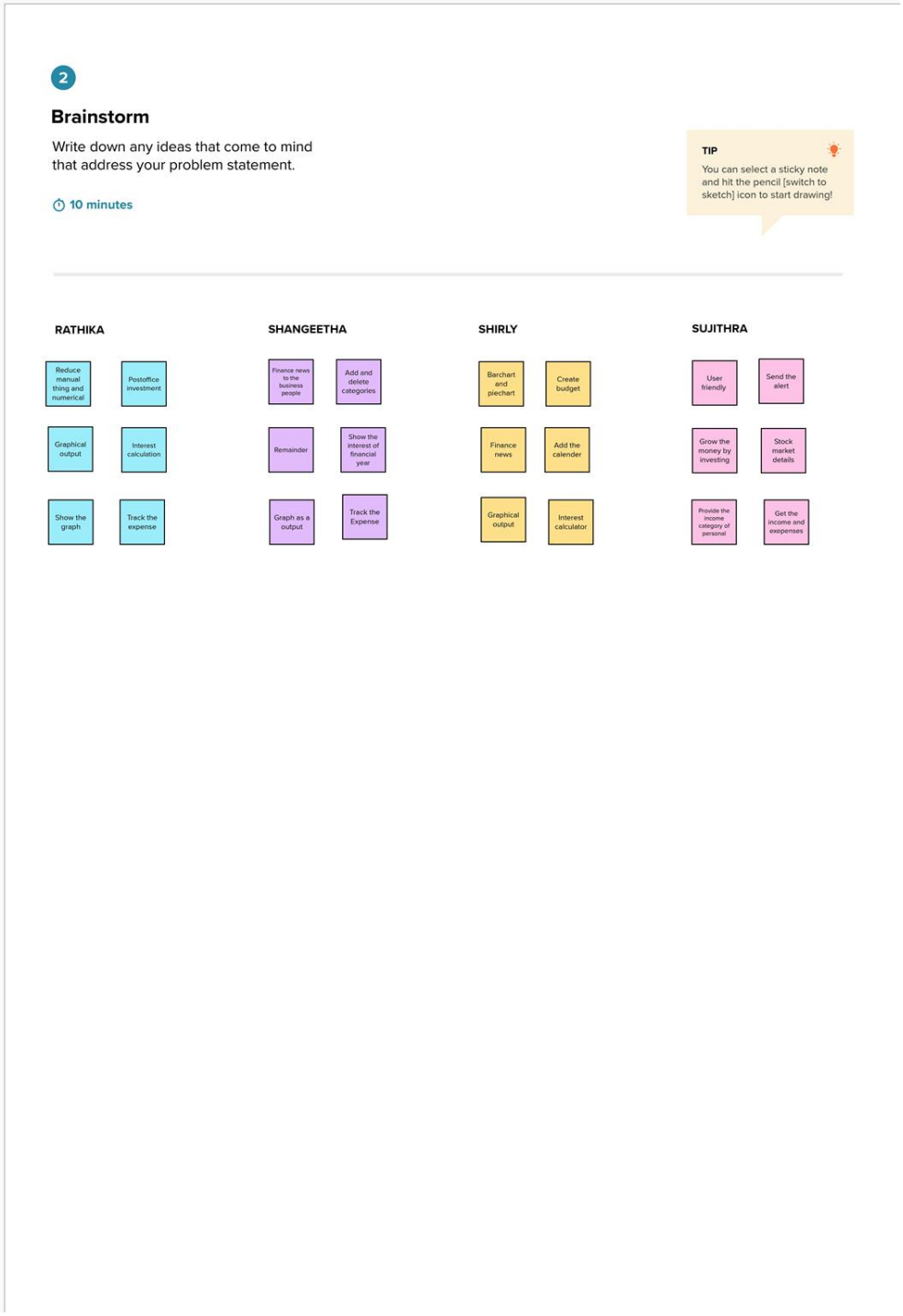
3.2 Ideation and Brainstroming:

TEAM ID - PNT2022TMID00673

In a free and open setting, brainstorming enables team members to engage in the innovative problem-solving process. Prioritizing quantity over quality, unconventional ideas are welcomed and expanded upon, and everyone is urged to cooperate in order to produce a wealth of creative solutions.



TEAM ID - PNT2022TMID00673



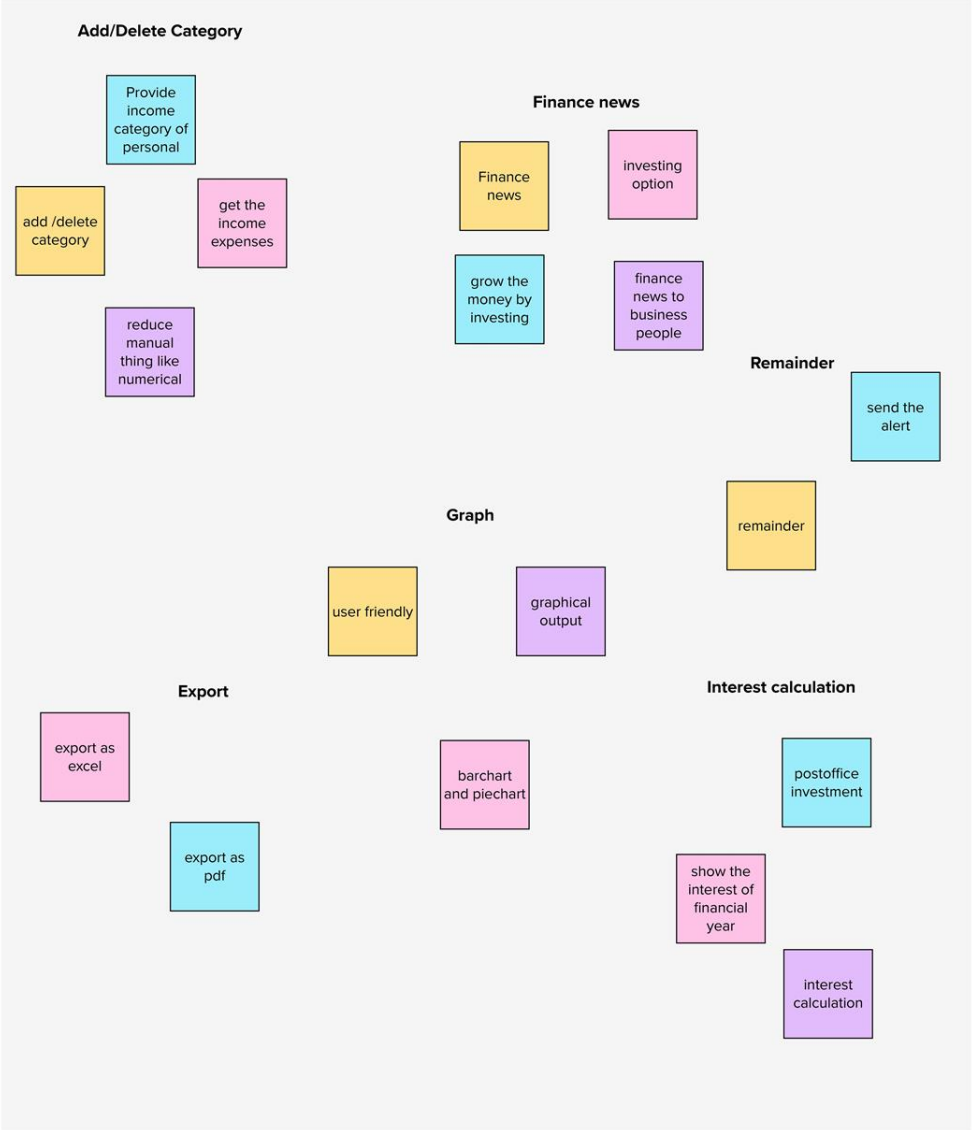
TEAM ID - PNT2022TMID00673

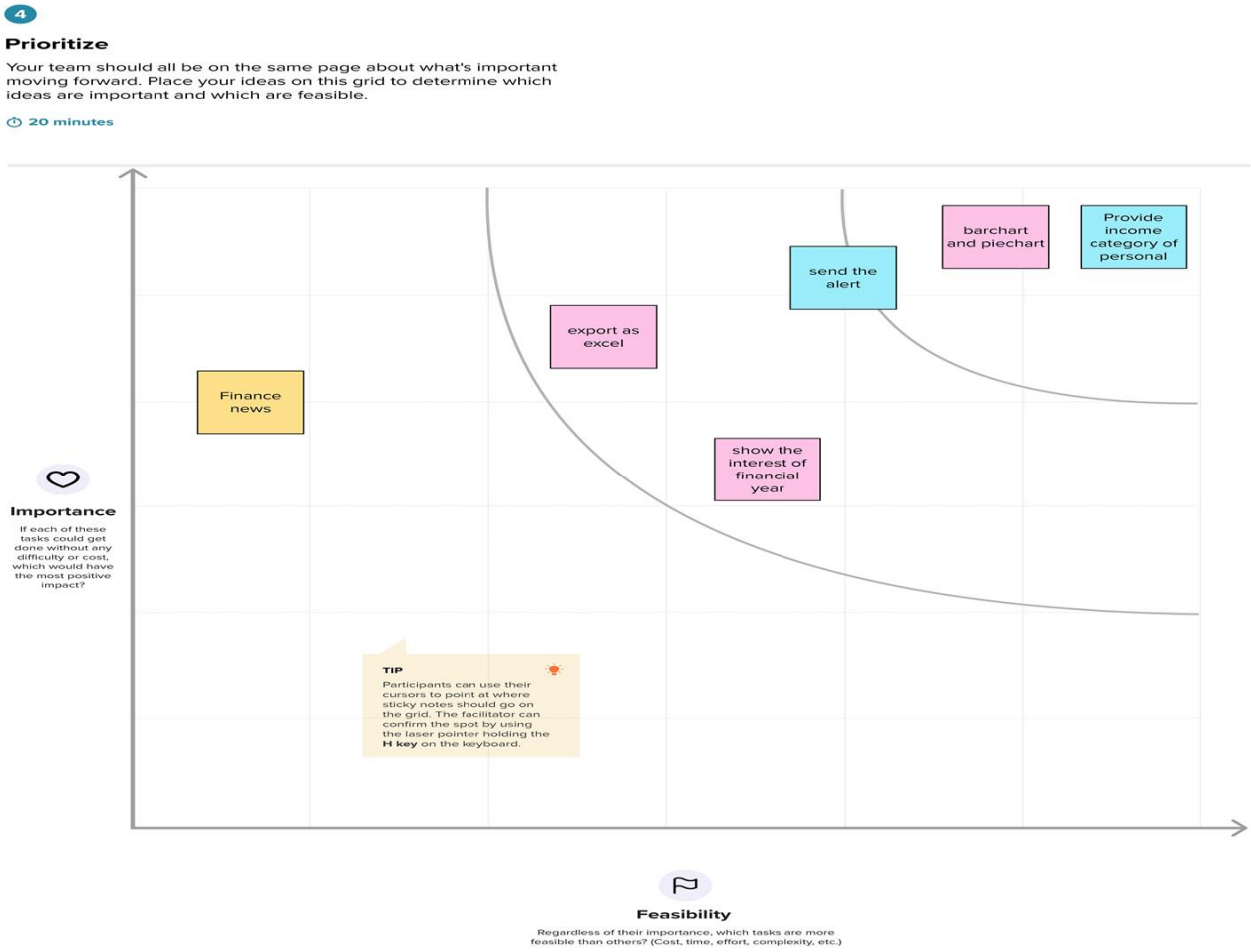
3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes





3.4 Proposed Solution:

The data in this personal expense tracker project is stored in the IBM DB2 cloud rather than on local storage. Here, the idea of "containerization," which replaced "virtualization," enables the user to run the application consistently and uniformly on any infrastructure by utilising the Docker programme. The user is guided by the IBM Watson Assistant chatbot, which also provides information on the application. This system stores project backup information in IBM Cloud Foundry so that, in the event of a failure, the data is immediately rolled back to the most recent checkpoint. Here, our project is constructed with Python Flask, which enhances its potential to scale. A warning email saying that the user has over his expense limit using Send Grid will be sent to him if he goes over the limit.

4.Requirement Analysis

4.1 Functional Requirment:

TEAM ID - PNT2022TMID00673

Following are the functional requirements of the proposed solution.

FRNo.	FunctionalRequirement(Epic)	SubRequirement(Story/Sub-Task)
FR-1	UserRegistration	Registration through Application Registration through Gmail
FR-2	UserConfirmation	Confirmation via EmailConfirmationvia OTP
FR-3	User monthly expense tentativedata	Datato beregisteredintheapp
FR-4	Usermonthlyincomedata	Datato beregisteredintheapp
FR-5	Alert/Notification	Alert through E- mailAlertthroughSM S
FR-6	UserBudgetPlan	PlanningandTrackingofuserexpensevsbudgetlimit

4.2 NON-FUNCTIONAL REQUIREMENTS:

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

FR No.	Non-Functional Requirement	Description

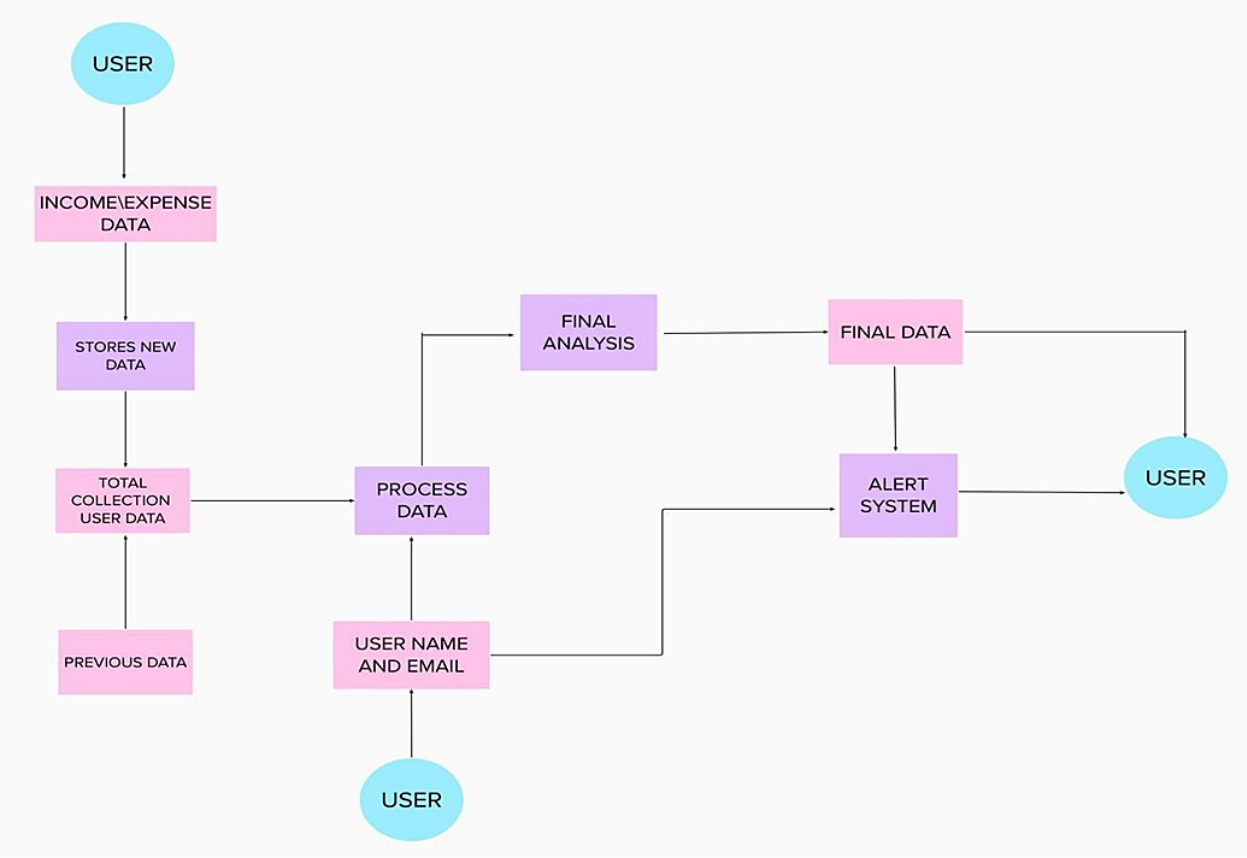
NFR-1	**Usability**	Effectiveness, efficiency and overall satisfaction of the user while interacting with our application.
NFR-2	**Security**	Authentication, authorization, encryption of the application.
NFR-3	**Reliability**	Probability of failure-free operations in a specified environment for a specified time.
NFR-4	**Performance**	How the application is functioning and how responsive the application is to the end-users.
NFR-5	**Availability**	Without near 100% availability, application reliability and the user satisfaction will affect the solution.
NFR-6	**Scalability**	Capacity of the application to handle growth, especially in handling more users.

5.Project Design

5.1 Dataflow diagram:

The information flows inside a system are traditionally shown visually in a data flow diagram (DFD). The appropriate amount of the system need can be graphically represented by a clean and unambiguous DFD. It demonstrates where data is stored, how it enters and leaves the system, and what modifies the data

TEAM ID - PNT2022TMID00673



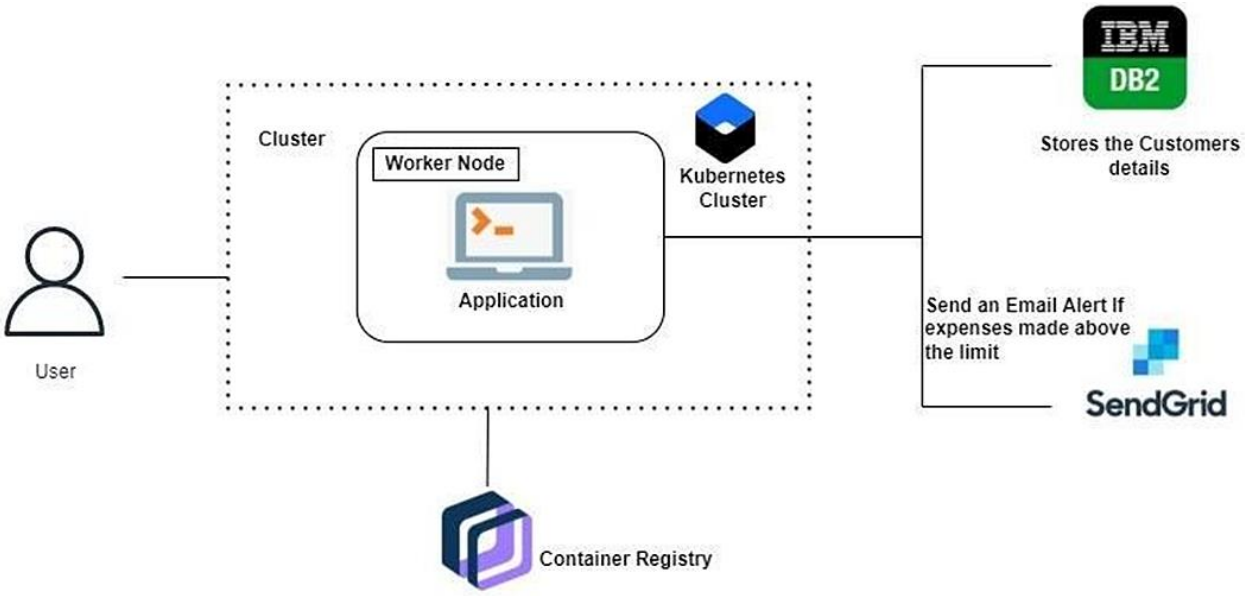
5.3 Solution and Technical Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

TEAM ID - PNT2022TMID00673



5.3 User stories:

User Type	Functional Requirement	User Story Number	User Story / Task	Acceptance criteria	Priority

TEAM ID - PNT2022TMID00673

Customer (web user)	Registration	USN-1	As a user, I can register for the application by entering my email and confirming my password.	I can access my account/dashboard	High
		USN-2	As a user, I will receive a I can receive a confirmation email once I confirmation have registered for the email & click application	I can receive a confirmation email	High
		USN-3	As a user, I can register access the Facebook	I can register through access the Facebook	Low
	Login	USN-4	As a user, I can log in application by entering application my email and password.	I can access the application	Medium

	Dashboard	USN-5	As a user, I can view my income and expenditure details	I can view my daily expenses.	High
Customer Care Executive		USN-6	As a customer executive, I can solve the login issue and other issues of the solution at any application	I can provide support	Medium
Administrator	Application	USN-7	As an administrator, I can upgrade or update the application	I can fix the bug	Medium

6.Project planning and Scheduling

TEAM ID - PNT2022TMID00673

6.1 Sprint planning and 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.	8	High	Shirly, Rathika
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	8	High	Sujithra Shangeetha
Sprint-1	Validating user	USN-3	Checking whether new user or existing user of the application	4	Medium	Shirly, Sujithra
Sprint-2	Add Expense	USN-4	As a user, I can add the day-to-day expense to the application	8	High	Rathika, Shangeetha
Sprint-2	Edit and Delete Expense	USN-5	As a user, I can edit and delete the previously created expense	8	High	Shirly, Rathika
Sprint-2	Creating time-based filters in history.	USN-6	As a user, I can see the time-based history of expenses.	4	Medium	Sujithra Shangeetha
Sprint-3	Integrating with pie charts for analysis	USN-7	As a user, I can view diagrammatic representation of expenses	8	High	Shirly, Sujithra
Sprint-3	Enabling limit feature	USN-8	As a user, I can set monthly limit to expenses	4	Medium	Rathika, Shangeetha
Sprint-3	Sending Email Alerts	USN-9	As a user, I will receive a mail if I cross a limit	8	High	Shirly, Sujithra
Sprint-4	Testing	USN-9	Testing the application with various tools	10	High	Rathika, Shangeetha
Sprint-4	Deployment	USN-9	Deployment of the application	10	High	Shirly, Sujithra

6.2 Sprint delivery Schedule:

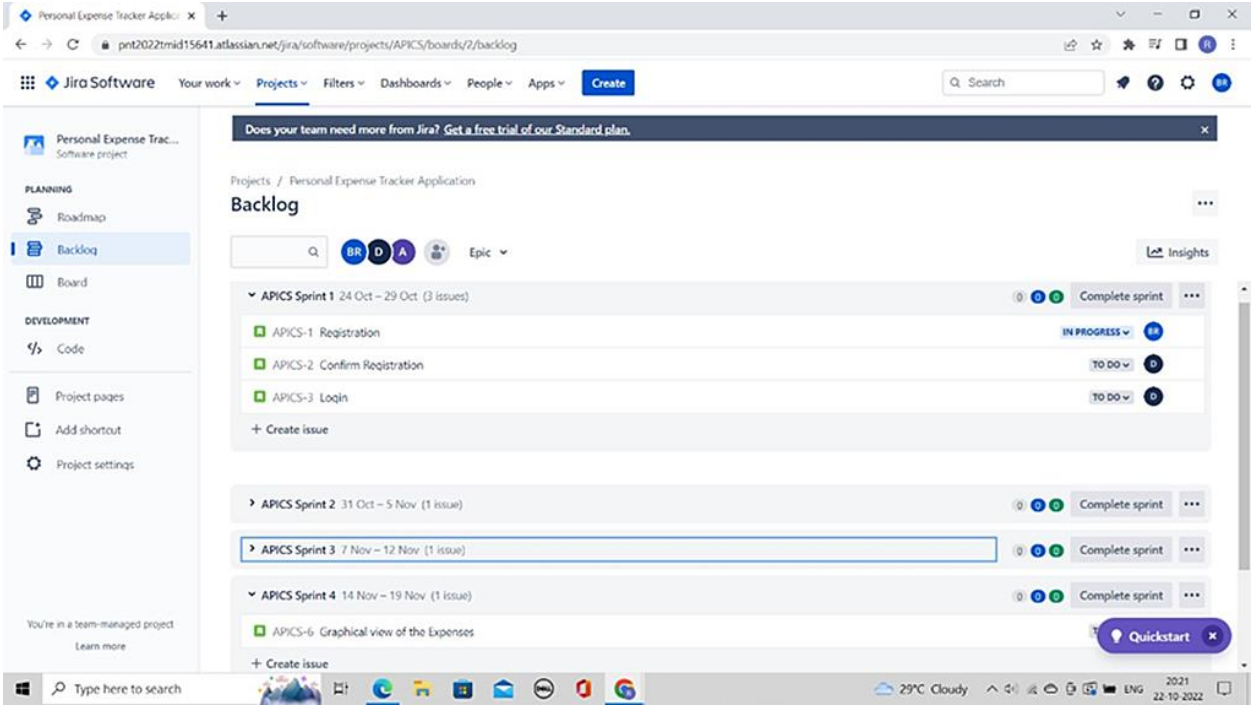
TEAM ID - PNT2022TMID00673

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

6.3 Reports from JIRA:

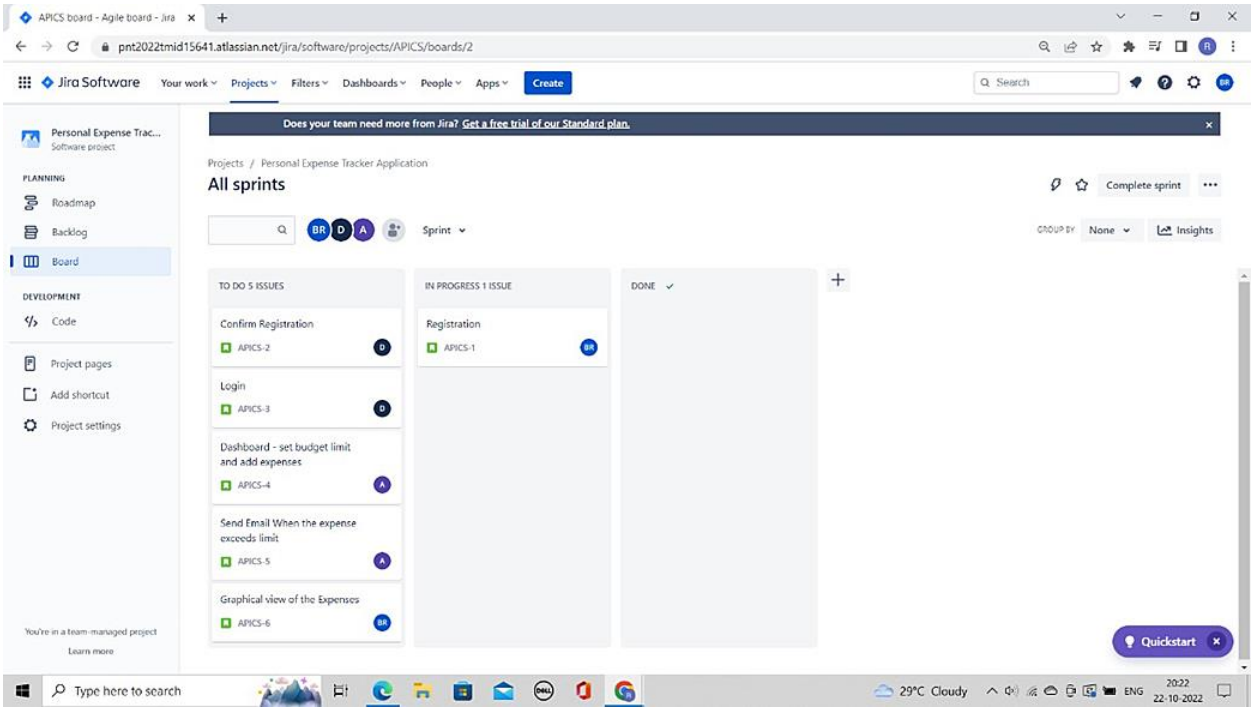
JIIRA is a software application used for issue tracking and project management.The tool, developed by the Australian software company Atlassian, has becomewidely used by agile development teams to track bugs, stories, epics, and othertasks

6.3.1 Backlog:

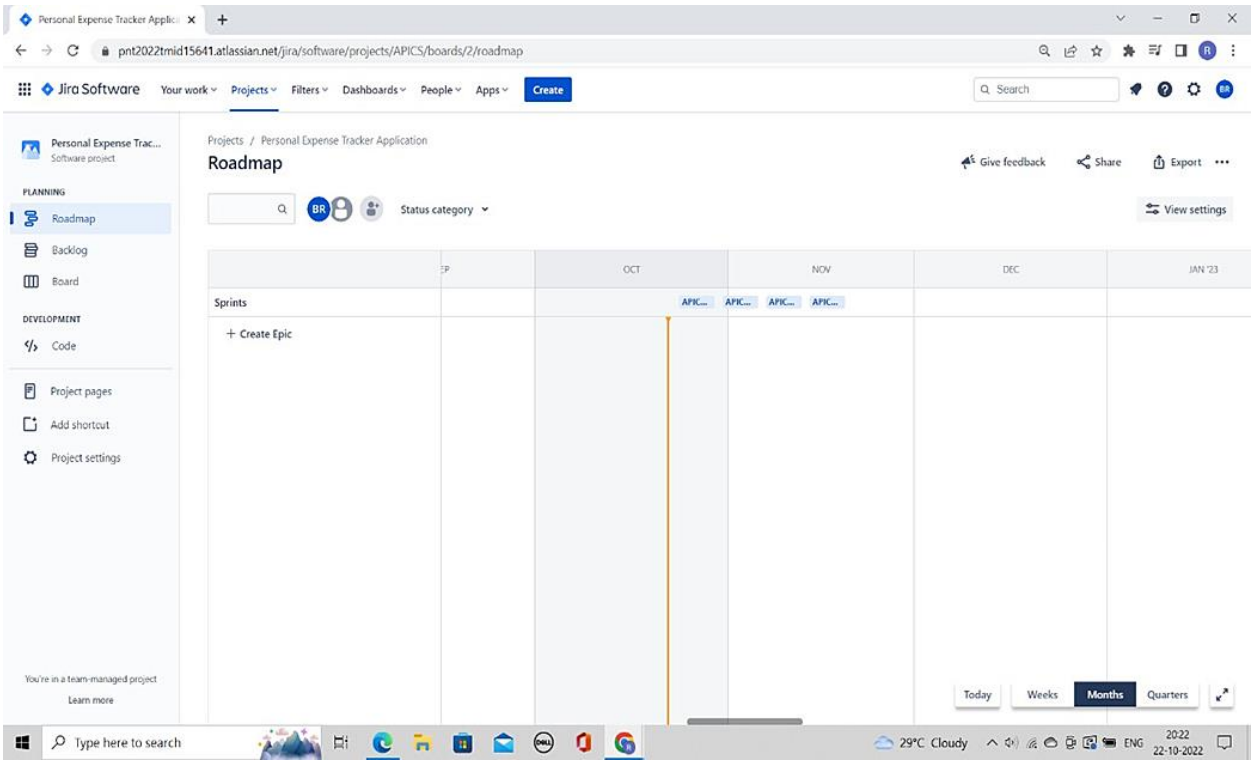


TEAM ID - PNT2022TMID00673

6.3.2 Board:



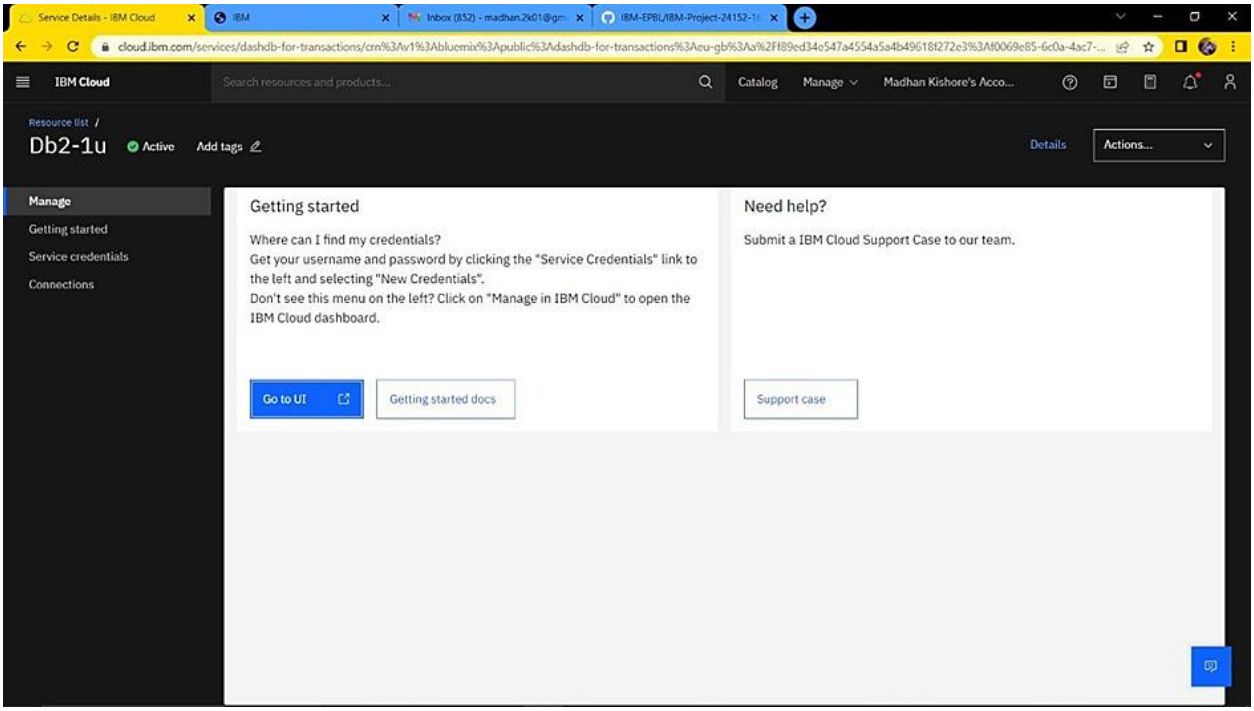
6.3.3 Road Map:



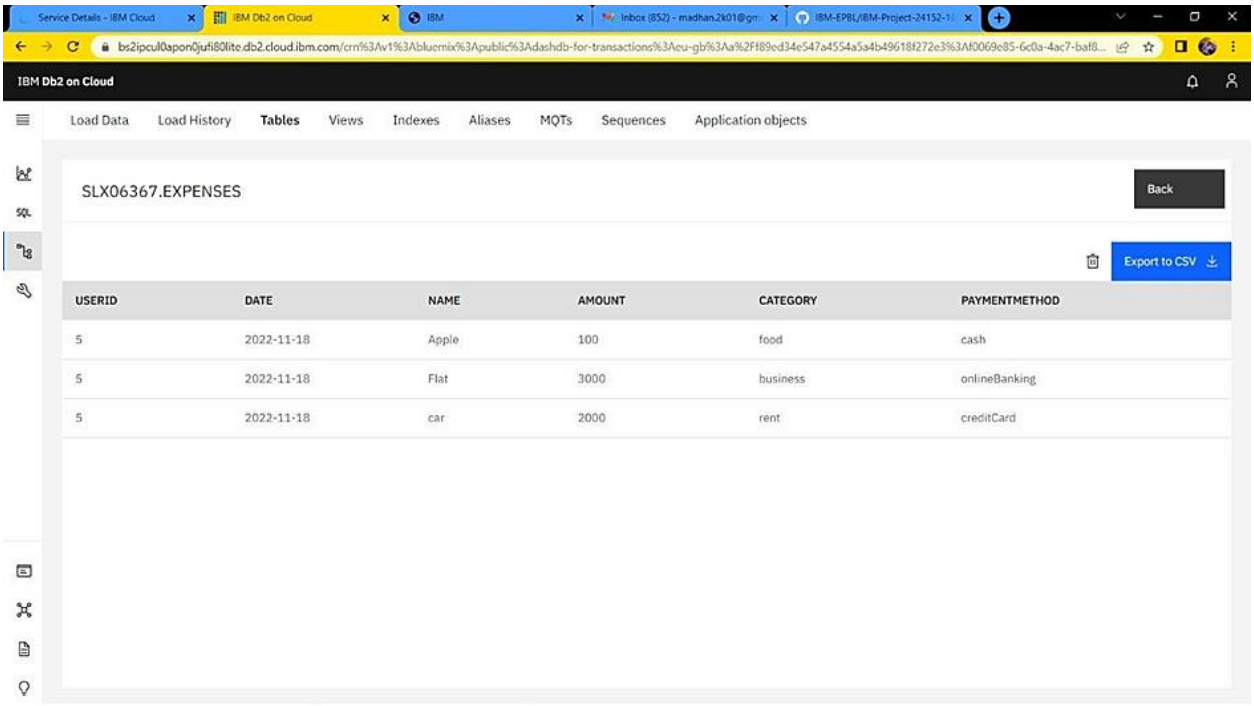
TEAM ID - PNT2022TMID00673

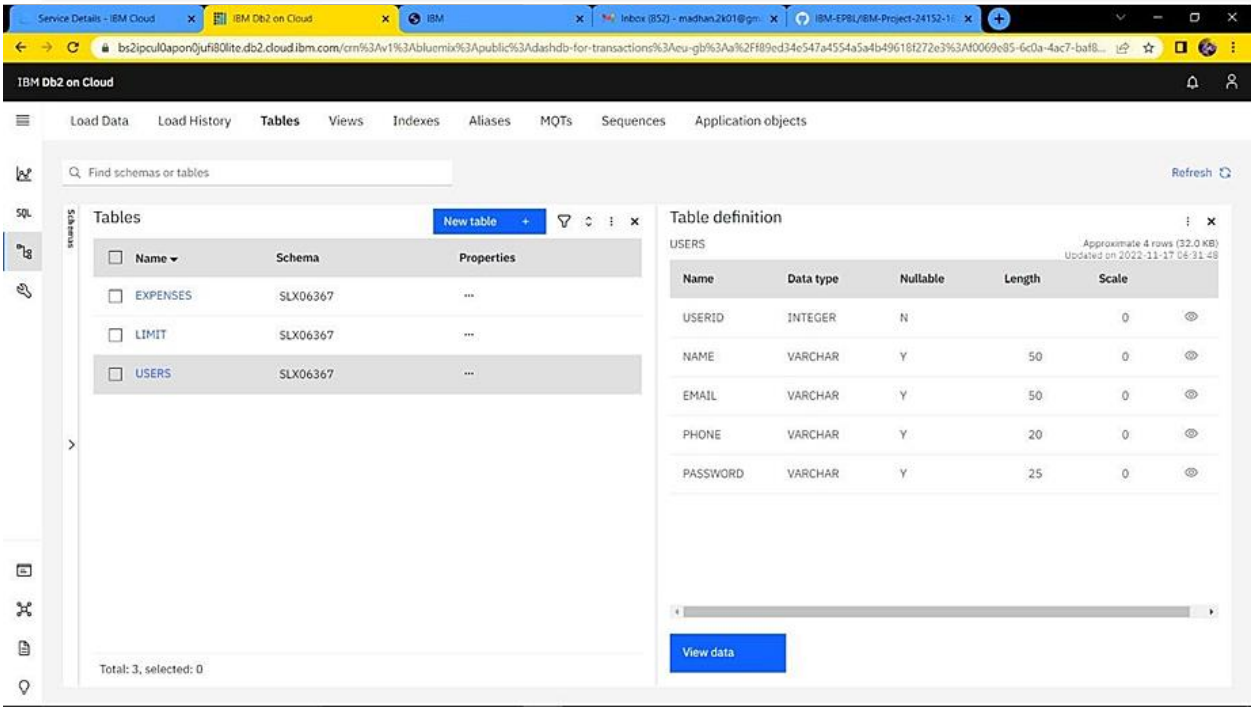
7.Coding and Solutioning

7.1 Database Schema:

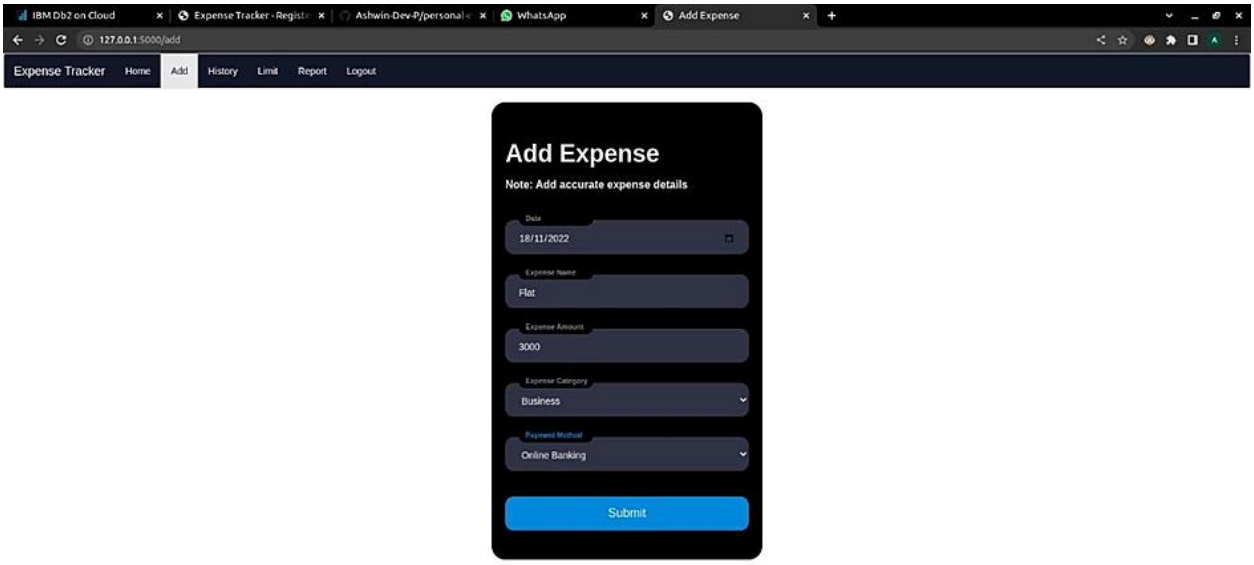


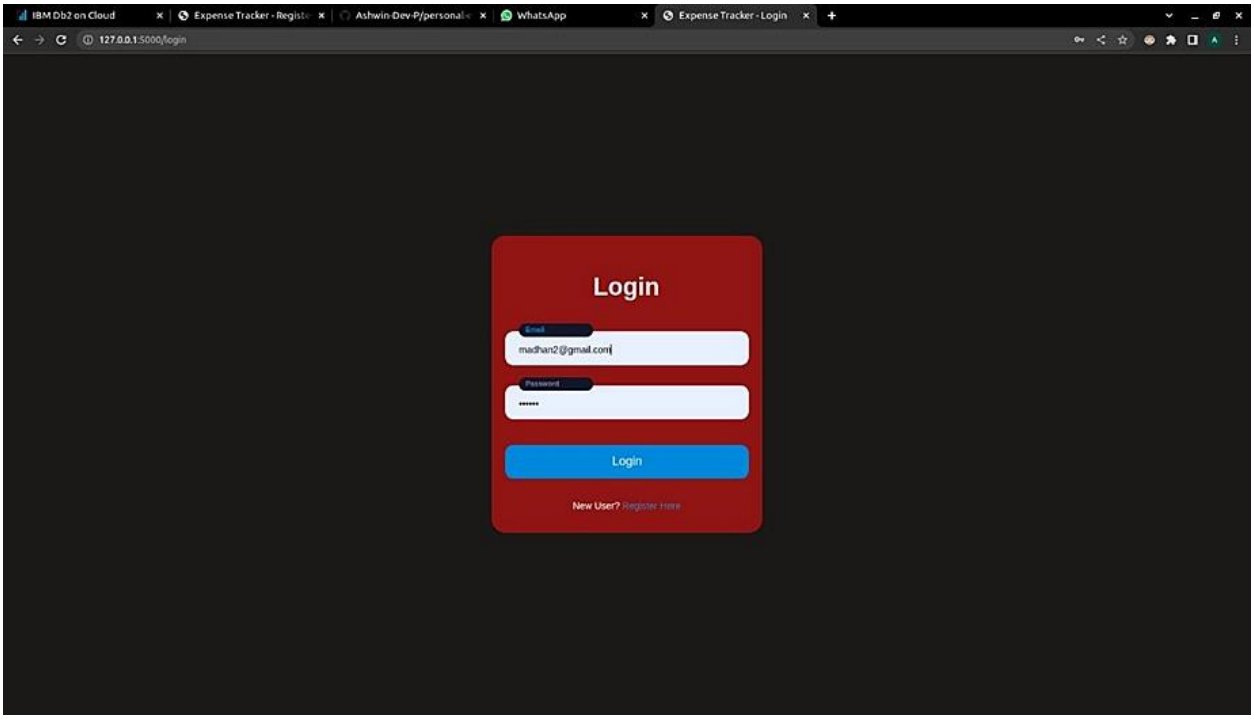
TEAM ID - PNT2022TMID00673



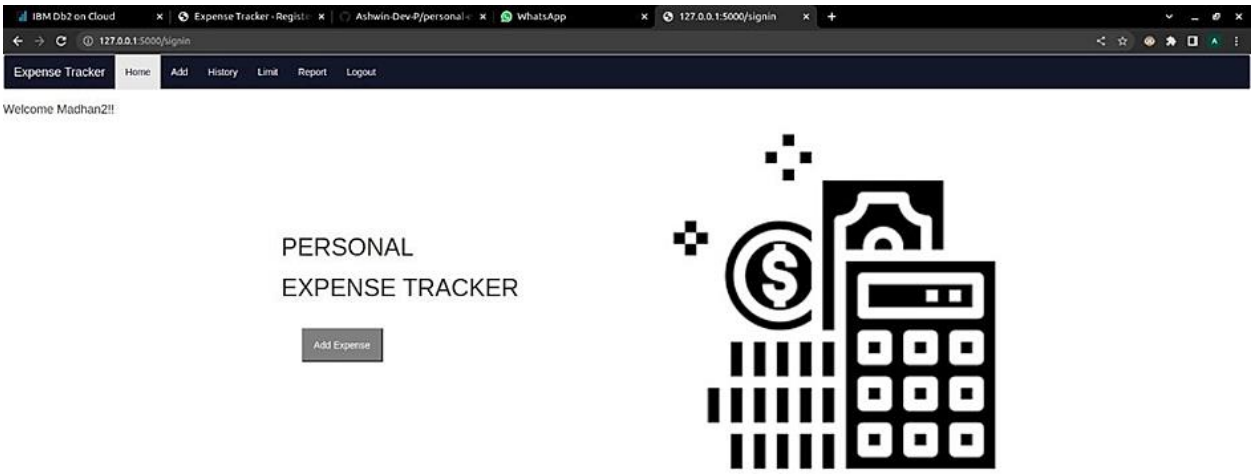


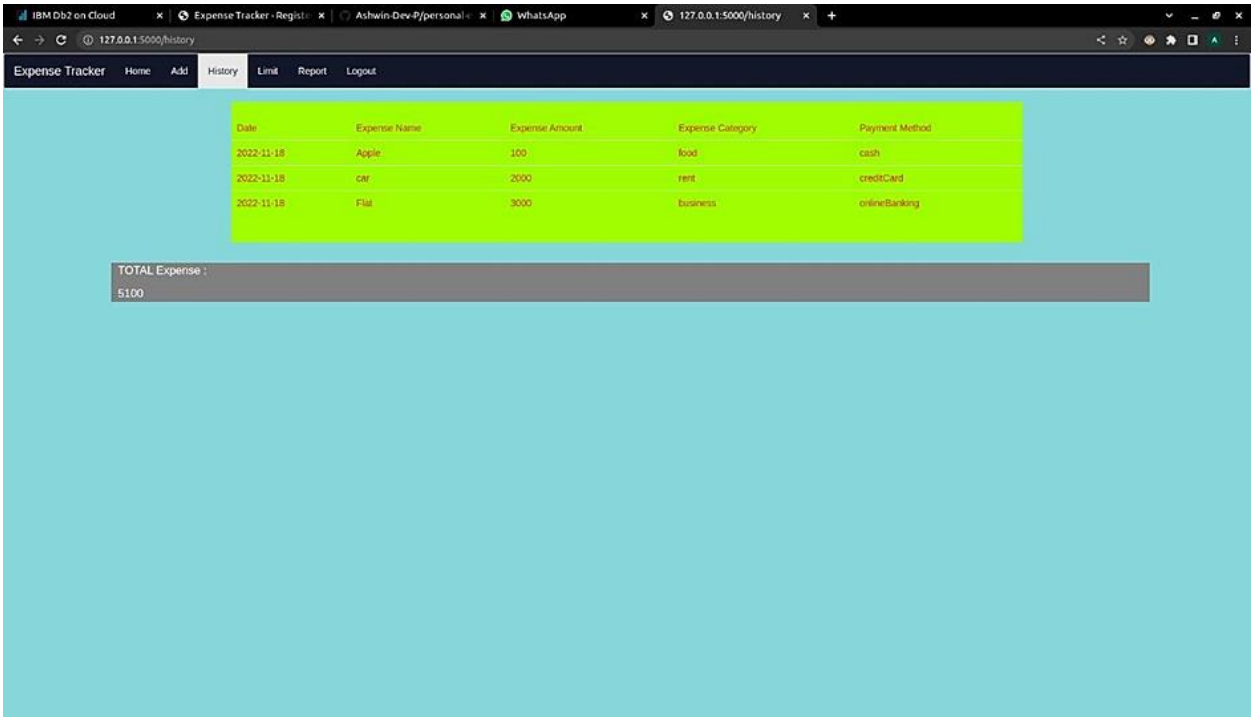
TEAM ID - PNT2022TMID00673



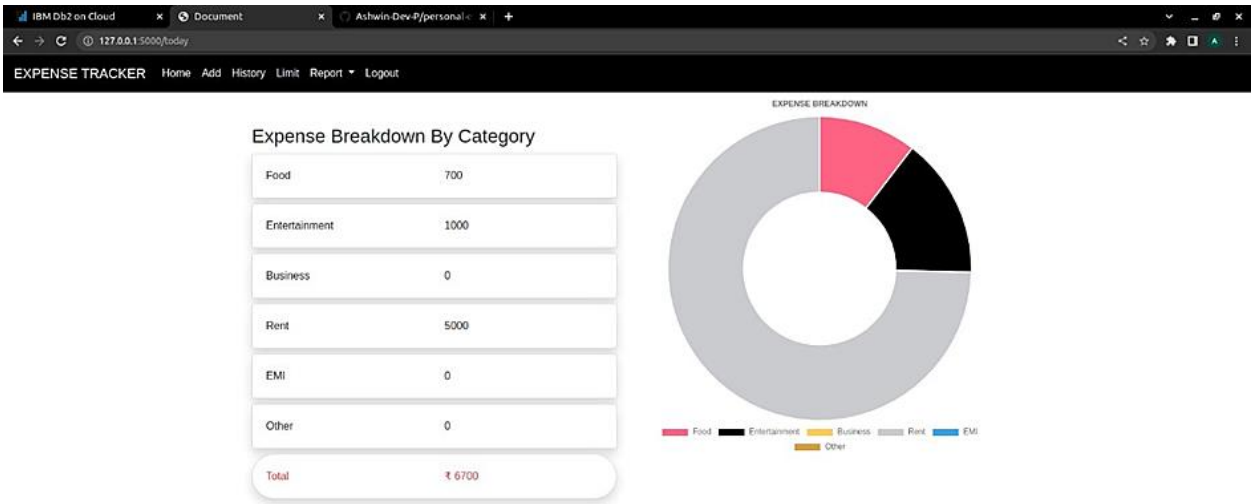


TEAM ID - PNT2022TMID00673





TEAM ID - PNT2022TMID00673



8.Testing

8.1 Test cases

				Date	12-Nov-22								
				Team ID	PNT2022TMID00713								
				Project Name	Personal Expense Tracker Applicati								
				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	None	1Go to Homepage	Username: Rahul password: 123	Login/Signup popup should display	Working as expected	Pass				Rahul S
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup	Home	1.Go to website 2.Enter details and click login	Username: Rahul password:	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link	Working as expected	Pass				Tarun V
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	Username and password	1.Go to website 2.Enter details and click login	Username: Rahul password: 123	User should navigate to user account homepage	Working as expected	Pass				Sai Kumar M
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	Username and password	1.Go to website 2.Enter details and click login	Username: Rahul password: 1234	Application should show 'Incorrect email or password ' validation message	Working as expected	Pass				Rakesh A
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	Login First	1.Go to website 2.Enter details and click login	Username: Rahul password: 1234	Application should show 'Incorrect email or password ' validation	Working as expected	Pass				Rahul S
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials	Login First	1.Go to website 2.Enter details and click login	Username: Rahul password: 1234	Application should show 'Incorrect email or password ' validation	Working as expected	Pass				Tarun V
dExpensePage_TC_O	Functional	d Expense Pa	Verify whether user is able to add exp	Add some expenses	Add date,expense name and other deta	Food:2000	Application adds expenses	rking as expect	Pass				Sai Kumar M
dExpensePage_TC_O	Functional	d Expense Pa	Verify whether user is able to add exp	Add some expenses	Add date,expense name and other deta	Cloths:500	Application adds expenses	rking as expect	Pass				Rakesh A
teExpensePage_TC_	Functional	ate Expense	Verify whether user is able to verify e	Update expenses	1.Update expenses	Update Food:2400	Application updates expenses	rking as expect	Pass				Rahul S
AddLimit_TC_OO7	Functional	Add limit	Verify whether user is able to add limit or not	Add Limit	1.Add Limit	Limit:50000	Application add limits for expenses	Working as expected	Pass				Tarun V
GenerateReport_TC OO8	Functional	Generate report	Verify whether the report is generating or not	Report Generation	1.Generate Report	Report	Application generates reports	Working as expected	Pass				Sai Kumar M

8.2 User Acceptance Testing:

TEAM ID - PNT2022TMID00673

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	8	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	11	20
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	0	1	8
Totals	22	14	11	22	51

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Interface	7	0	0	7
Login	43	0	0	43
Logout	2	0	0	2
Limit	3	0	0	3
Signup	8	0	0	8
Final Report Output	4	0	0	4

9.Results

The new system has overcome most of the limitations of the existing system and works according to the design specification given. The project what we have developed is work more efficient than the other income and expense tracker. The project successfully avoids the manual calculation for avoiding calculating the income and expense per month. The modules are developed with efficient and also in an attractive manner. The developed systems dispense the problem and meet the needs of by providing reliable and comprehensive information. All the requirements projected by the user have been met by the system. The newly developed system consumes less processing time and all the details are updated and processed immediately.

10.Advantages & Disadvantages

ADVANTAGES:

1.You have no controlover your money If you don't check your spending andcreate a budget, you will have no control whatsoever on your money. Instead, money will control you, and you will either have perpetual lack of funds or you will end up steeped in debt. A money manager app helps you decide between short-term and long-term spending.

2.You have no financial goals If you are spending money frivolously, you will not have money to set financial goals. However, when you have a daily expensemanager, you will be able to work with limited resources and use your money ina wise manner so that you can createfinancial goals and ensure you meetthem.

3.You are unaware what is happening with your money If you are clueless about how much is your inflow and how much you are spending, you will not know at the end of the month what happenedto your money. An expensetrackerhelps you figure out what is happening to your money, and whether you can afford something you want.

4.You spend and save ina haphazard manner

If you don't have great financial management skills, you will not know how to categorize your expenses. However, tracking your expenses and budgeting themwill help you become aware of how much you have to allocate to each expense category, and ifyou are short, you will beable to make adjustments with ease.

5.You have no clue about making your money work for you, In this day andage, when expenses are going through the roof, it has become crucial that you learn to make your money work for you so that you can create a nest egg for the future.

6.You don't have funds for emergencies Remember, emergencies come whenyou least expect. Hence, if you don't have money stashed

away for a rainy day, you will end up borrowing from family and friends. This way you could get intodebt that will be difficult to pay back due to your poor money management skills

DISADVANTAGES:

Your information is less secure, and probably being used and sold. If the serviceis free, then the product is you. Mint.com, like other financial apps, is a free service. They have to pay their bills somehow,so regardless of what theirprivacypolicy may or may not say, just assumethat your spendinghistory and trendsaregoing to be recordedand analyzed, by someone, somewhere. Now, you shouldn'thave to worry about credit card fraud or identity theft, these companies are largeenough and secure enough that you'll never have to worry about something like that. Just recognize that your information, most likely anonymous, will be used and potentially even sold. Personally, I have no problem with that, but if you do, then make sure you avoid these types of services**.** Automating everything to do with your finances can make you financially lazy. If your bills are paid automatically and your finances are track automatically, then what is there left for you to do? Not a lot, to be honest.So you might stop caring_about_what you'respending_andwhere your money is going.Eventually you may look at your Mintdata and realize that you've blown your budget over the last two months, but by then it is too late. So if you do choose to use this program, ensure that you are also being diligent in checking in on your finances. Set up a weekly or biweekly checkfor yourself to go throughyour finances and hit on all the important points.

11.Conclusion

After making this application we assure that this application will help its users to manage the cost of their daily expenditure. It will guide them and aware themabout there daily expenses. It will prove to be helpful for the people who are frustrated with their daily budget management, irritated because of amount of expenses and wishes to manage money and to preserve the record of their daily cost which may be useful to change their way of spending money. In short, this application will help its users to overcome the wastage of money.

12. FUTURE SCOPE

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

1. Multiple account support.
2. Include currency converter.

13.Appendix

13.1 Source code:

#app.py

```
import ibm_db

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

from markupsafe import escape


#mail

from flask import Flask

from flask_mail import Mail, Message



app = Flask(
name__)mail=
Mail(app)

# configuration of mail
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME']
="madhan.2k01@gmail.com"
```

TEAM ID - PNT2022TMID00673

```
app.config['MAIL_PASSWORD']
="ljquijksmoxcgsky"

app.config['MAIL_USE_TLS']=False
app.config['MAIL_USE_SSL']=       True
mail = Mail(app)



conn
=ibm_db.connect("DATABASE=bludb;HOSTNAME=
3883e7e4-18f5-4afe-be8c-
fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdo
main.cloud;PORT=31498;SECURITY=SSL;SSLServer
Certificate=DigiCertGlobalRootCA.crt;UID=slx06367;
PWD=ETdgEY31zKz50BKH","")
app.secret_key='a'



defsend_mail(recipient_mail):
msg = Message(

'Expense tracker',
sender ='madhan.2k01@gmail.com',
recipients = [recipient_mail]
)
msg.body = 'Your Expense Limit has Exceeded'
mail.send(msg)

return True
```

TEAM ID - PNT2022TMID00673

```
@app.route('/')
@app.route('/regi
ster')def
register():
return render_template('register.html')
@app.route('/index')
def index():
return render_template('index.html')


@app.route('/he
ader')def
header():
returnrender_template('header.html')


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




@app.route('/login')
def login():
return render_template('login.html')
```

TEAM ID - PNT2022TMID00673

```
@app.route('/addExpens
e')def addExpense():
returnrender_template('AddExpense.html')



@app.route('/addrec',methods = ['POST', 'GET'])def
addrec():
if request.method == 'POST':


name = request.form['name']

email = request.form['email']

phone                    =
request.form['phone']
password= request.form['password']


sql = "SELECT * FROM USERS WHERE NAME =?"
stmt=ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,name)
ibm_db.execute(stmt)

account = ibm_db.fetch_assoc(stmt)


if account:
return      render_template('login.html',msg="You are already a
member, pleaselogin using your details")
```

```
else:
insert_sql = "INSERT INTO USERS (Name,email,phone,password) VALUES (?,?,?,?)"
prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prep_stmt, 1, name)
ibm_db.bind_param(prep_stmt, 2, email )
ibm_db.bind_param(prep_stmt, 3, phone)
ibm_db.bind_param(prep_stmt, 4, password)
ibm_db.execute(prep_stmt)



return render_template('login.html',msg="Registered successfuly..")




@app.route('/signin', methods =['GET', 'POST'])def
signIn():
glob
al
user
id
msg
= "
if request.method == 'POST':
```

```
email = request.form['email'] password=
request.form['password']
sql ="SELECT * FROM USERS WHERE email = ?
AND password = ?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,email)
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['USERID']
userid=account['USERID']
session['username']=account['NAME']

#session["name"] = request.form.get("name")


#session['username'] = account['Name']
msg = 'Welcome'+" "+session['username']+"!!" return
render_template('home.html', msg = msg)else:
msg = 'Incorrect username / password !'
return render_template('login.html', msg = msg)
```

TEAM ID - PNT2022TMID00673

```
@app.route('/add', methods =['GET', 'POST'])def
add():
global id
if request.method=='POST':
date=request.form['date']
name=request.form['expenseName']
amount=request.form['expenseAmount']
category=request.form['expenseCategory']
paymethod=request.form['payMethod']

id=session['id']
print("session
id",id)



insert_sql = "INSERT    INTO     EXPENSES
(USERID,DATE,NAME,AMOUNT,CATEGORY,PA
YMENTMETHOD) VALUES (?,?,?,?,?,?)"
prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prep_stmt, 1, id)
ibm_db.bind_param(prep_stmt, 2, date )
ibm_db.bind_param(prep_stmt, 3, name)
ibm_db.bind_param(prep_stmt, 4, amount)
ibm_db.bind_param(prep_stmt, 5, category)
ibm_db.bind_param(prep_stmt, 6, paymethod)
ibm_db.execute(prep_stmt)

limit="SELECT   AMOUNT   FROM  LIMIT WHERE
```

```
USERID=?"
stmt = ibm_db.prepare(conn, limit)
ibm_db.bind_param(stmt,1,id)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)

print(account)
limit_amount=account['AMOUNT']
print(limit_amount)


total="SELECT SUM(AMOUNT) FROM EXPENSESWHERE
USERID=?"
stmt = ibm_db.prepare(conn, total)
ibm_db.bind_param(stmt,1,id)
ibm_db.execute(stmt)
account= ibm_db.fetch_assoc(stmt)


print("ac
count")

print(
account)


print("account id")
print(id)total_amount=
account['1']
print("total_amount")
print(total_amount)
```

```
if (int(total_amount)>int(limit_amount)):
print("Limit exceeded")
account_stmt="SELECT EMAIL FROM USERS
WHERE USERID=?"
stmt = ibm_db.prepare(conn, account_stmt)
ibm_db.bind_param(stmt,1,id)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)

send_mail(account['EMAIL'])
return render_template("limitwarn.html")
else:
print(total_amount, limit_amount)
return render_template('AddExpense.html')


@app.route('/hist
ory')

def history():
global
id
id=sess
ion['id']
```

```
print(session['usernam
e'])
students = []

total=0
sql = "SELECT  *  FROM   EXPENSES   where
USERID=?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,id)
ibm_db.execute(stmt)
dictionary = ibm_db.fetch_both(stmt)


while dictionary != False:
# print ("The Name is : ", dictionary)
students.append(dictionary)total+=int(dictionary[3])
dictionary = ibm_db.fetch_both(stmt)




if students:
return  render_template("history.html",students=
students,total=total)




@app.route('/
limit')def
limit():
```

```
return render_template("limit.html")


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

def limitnum():
if request.method == "POST":
number= request.form['number']
sql = "INSERT  INTO   LIMIT(USERID,AMOUNT)
VALUES(?,?)"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,session['id'])
ibm_db.bind_param(stmt,2,number)
ibm_db.execute(stmt)
returnrender_template("today.html")


@app.route('/lo
gout')

def logout():
session.pop('loggedin', None)
session.pop('id', None)
session.pop('username', None)
returnrender_template('register.html')

@app.route("/to
day")def
today():
```

```
sql = "SELECT * FROM EXPENSESWHERE userid
=? AND date = DATE(NOW())"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,session['id'])
ibm_db.execute(stmt)
list2=[]
texpense=ibm_db.fetch_tuple(stmt)
print(texpense)
```

```
sql = "SELECT * FROM EXPENSES WHERE
USERID=? AND DATE(date) = DATE(NOW())"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,session['id'])
ibm_db.execute(stmt)
list1=[]
expense = ibm_db.fetch_tuple(stmt)
while(expense):
list1.append(exp
ense)
expense = ibm_db.fetch_tuple(stmt)
```

```
t
o
t
a
l
=
0
t
_
f
o
o
d
=
0
t_entertain
ment=0
t_business=
0 t_rent=0
t_EMI=0
t_other=0



for x in list1:
total
+=int(x[3])
```

```
if x[4] ==
"food":
t_food += int(x[3])


elif x[4] == "entertainment":

t_entertainment  += int(x[3])

elif x[4] == "business":
t_business +=
int(x[3])

elif x[4] == "rent":
t_rent+= int(x[3])


elif x[4] =="emi":
t_EMI += int(x[3])


elif x[4] == "Miscellaneous":
t_other  += int(x[3])


return  render_template("today.html",  texpense  =  list1,expense
= expense,  total = total ,
t_food = t_food,t_entertainment = t_entertainment,t_business =
t_business,  t_rent = t_rent,
t_EMI =  t_EMI, t_other=  t_other )
```

```
@app.route("/month")

def month():

sql="SELECT MONTHNAME(DATE),SUM(AMOUNT) FROM EXPENSES WHERE USERID=? GROUP BY MONTHNAME(DATE)"

stmt = ibm_db.prepare(conn, sql)

ibm_db.bind_param(stmt,1,session['id'])

ibm_db.execute(stmt)

list2=[]

texpense = ibm_db.fetch_tuple(stmt)

while(texpense):

list2.append(texpense)

texpense = ibm_db.fetch_tuple(stmt)

print(list2)
```

```
sql = "SELECT * FROM EXPENSES WHERE USERID=?
AND
MONTH(date)=MONTH(DATE(NOW()))"

stmt = ibm_db.prepare(conn, sql)

ibm_db.bind_param(stmt,1,session['id'])
```

```
ibm_db.execute(
stmt)list1=[]
expense = ibm_db.fetch_tuple(stmt)
while(expense):
list1.append(expense)
expense = ibm_db.fetch_tuple(stmt)
print(list1)
t
o
t
a
l
=
0
t
_
f
o
o
d
=
0
t_entertain
ment=0
t_business=
0 t_rent=0
t_EMI=0
t_other=0
```

```
for x in list1:

total +=
int(x[3])

 if x[4] ==
"food":

 t_food
+=int(x[3])



elif x[4] == "entertainment":
t_entertainment  += int(x[3])

elif x[4] ==
"business":
t_business +=
int(x[3])
elif x[4] == "rent":
 t_rent += int(x[3])

elif x[4] == "emi":
t_EMI  += int(x[3])
```

```
elif x[4] ==
"Miscellaneous":
t_other  += int(x[3])

print(total)


print(t_food)
print(t_entertain
ment)
print(t_business)
print(t_rent)

print
(t_E
MI)
print
(t_ot
her)




return  render_template("month.html",  texpense  =  list2,
expense = expense,  total = total ,
t_food = t_food,t_entertainment = t_entertainment,t_business =
t_business, t_rent = t_rent,
t_EMI =  t_EMI, t_other=  t_other )
```

```
@app.route("/
year")

def year():



sql = "SELECT YEAR(DATE),SUM(AMOUNT) FROM EXPENSES WHERE USERID=? GROUP BY YEAR(DATE)"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,session['id'])
ibm_db.execute(stmt)
list2=[]

texpense = ibm_db.fetch_tuple(stmt)
while(texpense):
list2.append(texpense)
texpense = ibm_db.fetch_tuple(stmt)
print(list2)


sql = "SELECT * FROM EXPENSES WHERE USERID=?
AND
YEAR(date)=YEAR(DATE(NOW()))"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,session['id'])
ibm_db.execute(stmt)
list1=[]
```

```
expense = ibm_db.fetch_tuple(stmt)
while(expense):

 list1.append(expense)
expense = ibm_db.fetch_tuple(stmt)


t
o
t
a
l
=
0
t
_
f
o
o
d
=
0
t_entertain
ment=0
t_business=
0

t
_
r
```

```
e
n
t
=
0
t
_
E
M
I
=
0
t_other=0



for x in list1:
total +=
int(x[3])
if x[4] ==
"food":
t_food
+=int(x[3])

elif x[4] == "entertainment":
t_entertainment  += int(x[3])
```

```
elif x[4] ==
"business":
t_business +=
int(x[3])
elif x[4] == "rent":
t_rent += int(x[3])

elif x[4] == "emi":
 t_EMI  +=
int(x[3])

elif x[4] == "Miscellaneous":
t_other  += int(x[3])



print(total)


print(t_food)
print(t_entertain
ment)
print(t_business)

 print(t_rent)

print(t_EMI)

print(t_other)
```

```
return render_template("year.html", texpense = list2,
expense = expense,  total = total ,
t_food = t_food,t_entertainment = t_entertainment,t_business =
t_business,  t_rent = t_rent,
t_EMI =  t_EMI, t_other=  t_other )


if____name_____=='___main
':
app.run(host='0.0.0.0',debug=True)
```

13.2 Github & Project demo link:

Github: https://github.com/IBM-EPBL/IBM-Project-4158-1658722219
Project demo link:
https://drive.google.com/file/d/1XEq8sI11TtD3VFkxDya7p5jdLs_d6lqJ/view?usp=share_link