PROJECT REPORT
PERSONAL EXPENSE TRACKER APPLICATION

PNT2022TMID43646:

Aswin S

Allen Joshy

Navaneeth Krishnan RV

Sarath A

1. INTRODUCTION

a. Project Overview

The Daily Expense Tracker System separates inputs into daily expense allocable amounts. If you go over that day's allocable amount, it deducts it from your income and gives you a new daily expense allocable amount; if that day's allocable amount is less, it adds it to your savings. At the end of the month, a report genera ng system for daily expenses will produce the income-expenditure curve. You can use it to add any savings you have set aside for special occasions like birthdays and anniversaries.   b. Purpose

The "Expense Tracker" programme was created to help manage daily spending in a more effec ve and prac cal manner. u lising this programme. We can lessen the manual tracking of expenditures and daily expense calcula ons.

2. LITERATURE SURVEY

Existing problem

Using a manual accoun ng system may have a number of drawbacks. Any business' accoun ng process can be challenging. A computerised accoun ng system may not require you to comprehend the accoun ng process in the same way that a manual accoun ng system does. Depending on who is conduc ng the bookkeeping, this could be a benefit or a drawback. To ensure

that accoun ng is done correctly, a skilled professional is frequently required. By hand, it could take a while to sort through your financial records' intricacy. Report genera on takes b. References

[1]. Pales nian Ministry of Educa on and Higher Educa on. Pales nian Higher Educa on Sta s cs.

 [2]. Accredita on and Quality Assurance Commi ee (AQAC) in Pales ne. General Report of Informa on Technology and Engineering Higher Educa on in Pales ne. Accredita on and Quality Assurance Commission (AQAC). Ramallah, Pales ne: Pales nian Ministry of Educa on and Higher Educa on; 2007 Apr.

[3]. Engineering Associa on of Pales ne. Current Engineering Sta s cs Book. Ramallah; 2005.

 [4]. Prados J, Peterson G, La uca L. Quality Assurance of Engineering Educa on Through Accredita on: The Impact of Engineering Criteria 2000 and Its Global Influence. Journal of Engineering Educa on. 2005 Jan; 94(1):165–
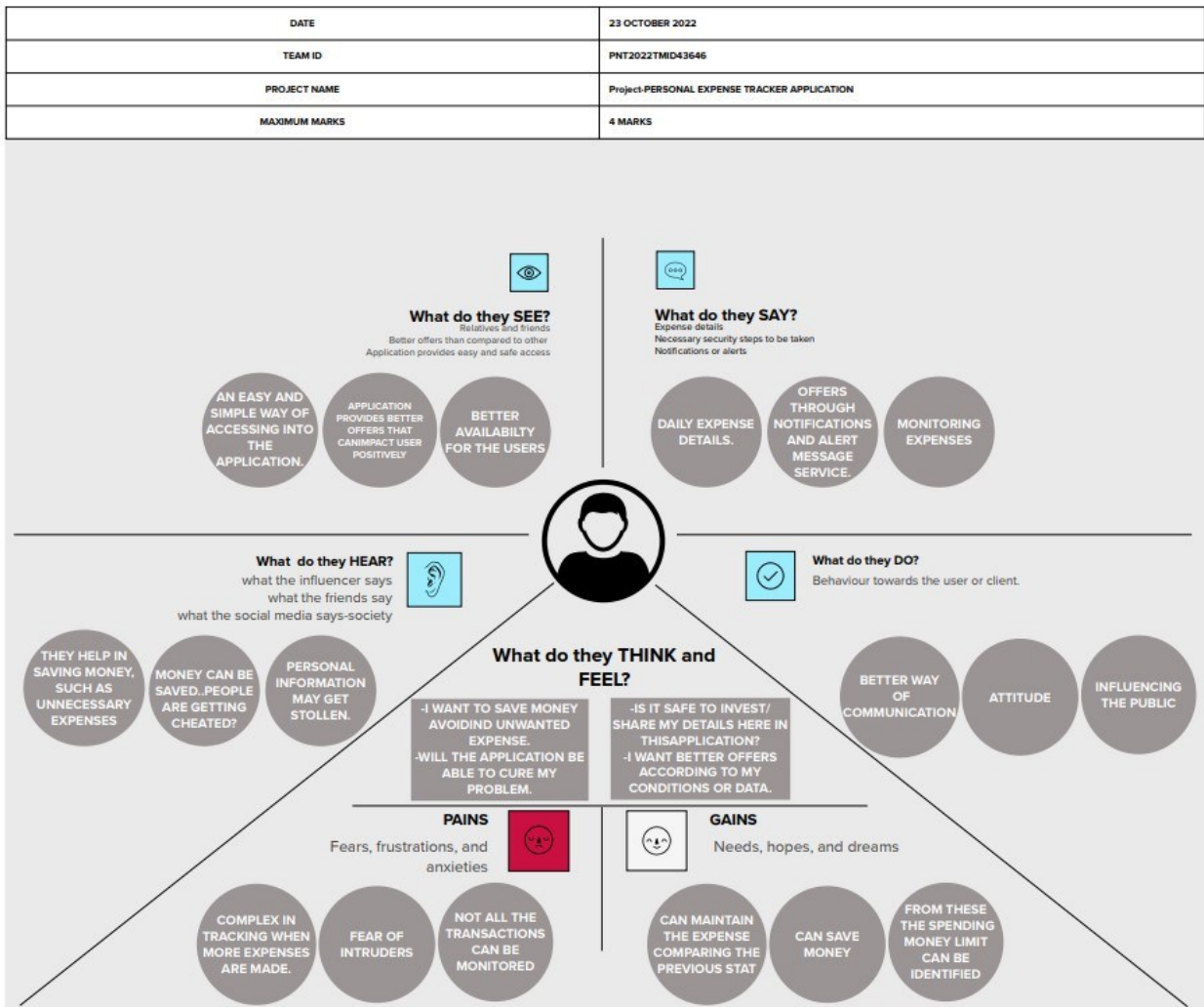
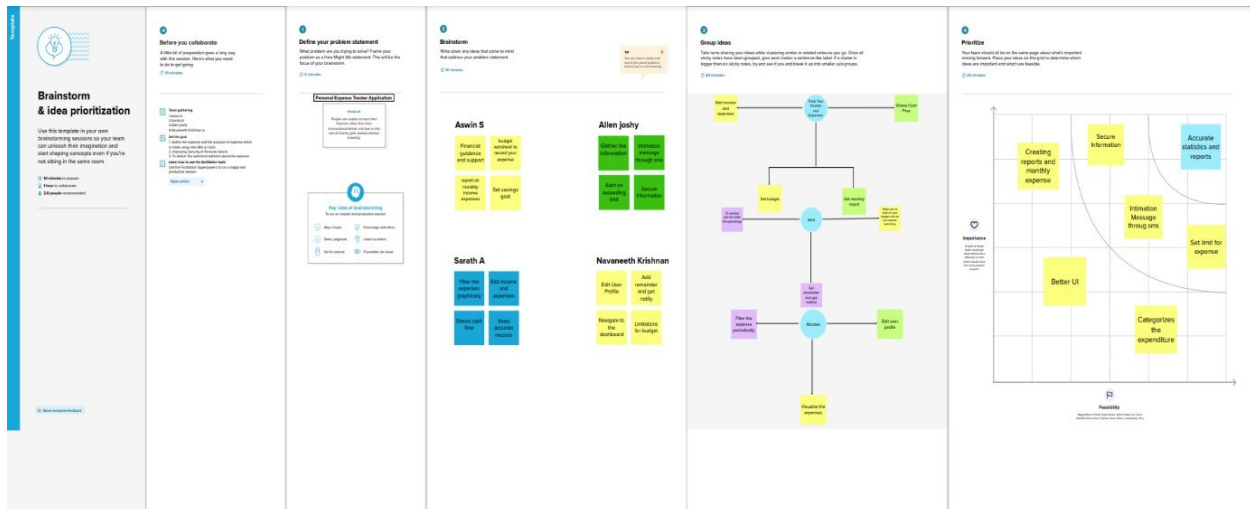## PERSONAL EXPENSE TRACKER APPLICATION

Literature Survey

S.NO	PAPER	AUTHOR	YEAR	METHOD AND ALGORITHM	ACCURACY
1	Spending Tracker: A Smart Approach to Track Daily Expense	Uday Pratap Singh, et al	2021	The waterfall model is used. This application will help its users to manage the cost of their daily expenditure. It will guide them and aware them of their daily expenses.	91.00%
2	Expense Manager Application	A Velmurugan, et al	2020	This paper's main aim to eliminate the use of sticky notes, spreadsheets and handling of large chunks of data is successful, the new experience is hassle-free and very handy. It uses the Core Data Model.	94.02%
3	Expense Tracker : A Smart Approach to Track Everyday Expense	Hrithik Gupta, et al	2020	This application will help its users to manage the cost of their daily expenditure. It will guide them and aware them about their daily expenses. Waterfall model is used for the project because all the requirements are clear as this project is not dealing with the clients and hence beforehand planning can be made about how to carry out each phase of development.	89.92%
4	Cloud based Expense Tracker	Asthha Wahal, et al	2018	The waterfall model is used. This application will help its users to overcome the wastage of money. It will guide them and aware them about their daily expenses	93.4%
5	Expense tracker mobile application.	Angad Manchanda	2012	Modern life offers a plethora of options of service and goods for the customers. As a result people's expenses have gone up dramatically. compareing to a decade ago and cost of living has been increasing day by day. thus it is essential to keep a check on the expenses in order to live a good life with a proper budget set up.	TECHNOLOGY: HTML, CSS & JAVASCRIPT CODE

3. IDEATION & PROPOSED SOLUTION

a. Empathy Map Canvas

DATE	23 OCTOBER 2022
TEAM ID	PNT2022TMID43646
PROJECT NAME	Project-PERSONAL EXPENSE TRACKER APPLICATION
MAXIMUM MARKS	4 MARKS

What do they SEE?
Relatives and friends
Better offers than compared to other
Application provides easy and safe access

- AN EASY AND SIMPLE WAY OF ACCESSING INTO THE APPLICATION.
- APPLICATION PROVIDES BETTER OFFERS THAT CANIMPACT USER POSITIVELY
- BETTER AVAILABILTY FOR THE USERS

What do they SAY?
Expense details
Necessary security steps to be taken
Notifications or alerts

- DAILY EXPENSE DETAILS.
- OFFERS THROUGH NOTIFICATIONS AND ALERT MESSAGE SERVICE.
- MONITORING EXPENSES

What do they HEAR?
what the influencer says
what the friends say
what the social media says-society

- THEY HELP IN SAVING MONEY, SUCH AS UNNECESSARY EXPENSES
- MONEY CAN BE SAVED..PEOPLE ARE GETTING CHEATED?
- PERSONAL INFORMATION MAY GET STOLLEN.

What do they DO?
Behaviour towards the user or client.

- BETTER WAY OF COMMUNICATION
- ATTITUDE
- INFLUENCING THE PUBLIC

What do they THINK and FEEL?

- -I WANT TO SAVE MONEY AVOIDIND UNWANTED EXPENSE.
  -WILL THE APPLICATION BE ABLE TO CURE MY PROBLEM.
- -IS IT SAFE TO INVEST/ SHARE MY DETAILS HERE IN THISAPPLICATION?
  -I WANT BETTER OFFERS ACCORDING TO MY CONDITIONS OR DATA.

PAINS
Fears, frustrations, and anxieties

- COMPLEX IN TRACKING WHEN MORE EXPENSES ARE MADE.
- FEAR OF INTRUDERS
- NOT ALL THE TRANSACTIONS CAN BE MONITORED

GAINS
Needs, hopes, and dreams

- CAN MAINTAIN THE EXPENSE COMPARING THE PREVIOUS STAT
- CAN SAVE MONEY
- FROM THESE THE SPENDING MONEY LIMIT CAN BE IDENTIFIED

b.   Brainstorm and Ideation

c. Define Problem Statement

Ideation Phase
Define the Problem Statements

Date	25 october 2022
Team ID	PNT2022TMID43646
Project Name	Personal Expense Tracker Application
Maximum Marks	2 Marks

Customer Problem Statement:

d. Proposed solution

PROJECT: PERSONAL EXPENSE TRACKER APPLICATION	PROJECT DESIGN PHASE-1	Team ID: PNT2022TMID43646

Define CS, fit into CC	1. CUSTOMER SEGMENT • Customers those who spend money unwontedly and to track their expenses. • Customer those who can't remember their expense. • Those who expecting to track their expense via statistics.	6. CUSTOMER CONSTRAINTS • Customer should use UPI or Net-Banking to track the expense. • If the money is spend through cash customer must add the expense in the application.	5. AVAILABLE SOLUTIONS • **SPENDEE** Application available both android and the ios.	Explore AS, differentiate
Focus on J&P, tap into BE, understand RC	2. JOBS-TO-BE-DONE/PROBLEMS • The main Intention of the application is to track the expense and provide statistics of expenses • It provides statistics based on categories of expenses. • To include money spend through cash, bank cheque's etc.	9. PROBLEM ROUTE CAUSE • The Main problem is gathering the data from the UPI apps or Nat-Banking application. • This will act as the main problem of the application. • Laziness of the customer to add the expense done through cash in the application.	7. BEHAVIOUR • Customer should responsibly add the expenses done through off-line mode. • To assure the data safety to the user.	Focus on J&P, tap into BE, understand RC
Identify strong TR & EM	3. TRIGGERS • Customer may think, they spend more money and no saving. 4. EMOTIONS: BEFORE / AFTER • **BEFORE:** No Savings. • **AFTER:** Few saving due to expense tracking application	10. YOUR SOLUTION • Design a cloud based web Application of the expense tracker. • Provide statistic of the expense done by the user through the graphs or charts. • Providing email alerts if the total expense exceed the limit.	8. CHANNELS OF BEHAVIOUR • In Online mode user don't have more work user need to set the maximum expense limit. • In Off-line mode user should responsibly add the expenses done through cash	Identify strong TR & EM

4. REQUIREMENT ANALYSIS

a. Func onal requirement and Non func onal requirements

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	31 October 2022
Team ID	PNT2022TMID43646
Project Name	Project - Personal Expense Tracker Application
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

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	User Data	Data gathered in the application server is saved in the high security cloud server.
FR-4	Alert Notification	Alert messages through the Email or SMS.
FR-5	User Monthly Budget Plan	Setting Monthly budget to manage their expenses.
FR-6	Cloud Data Storage	To save the user valuable data high security cloud storage are used (AWS, IBM, GOOGLE, etc)

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 experience of the user interacting with our application should be maximised.
NFR-2	Security	Authentication, authorization, encryption of the data must be done in the application.
NFR-3	Reliability	Probability of error in the operations in a specified environment for a specified time should be minimised.
NFR-4	Performance	How the application is functioning accurately and effectively the application is to the end-users.
NFR-5	Availability	Using Cloud Storage and database, application reliability and the user satisfaction will affect the solution

b. Customer Journey Map

5. PROJECT DESIGN

a. Data Flow Diagram

Date	10 November 2022
Team ID	PNT2022TMID43646
Project Name	Project-Personal Expense Tracker Application
Maximum Marks	4 marks

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.
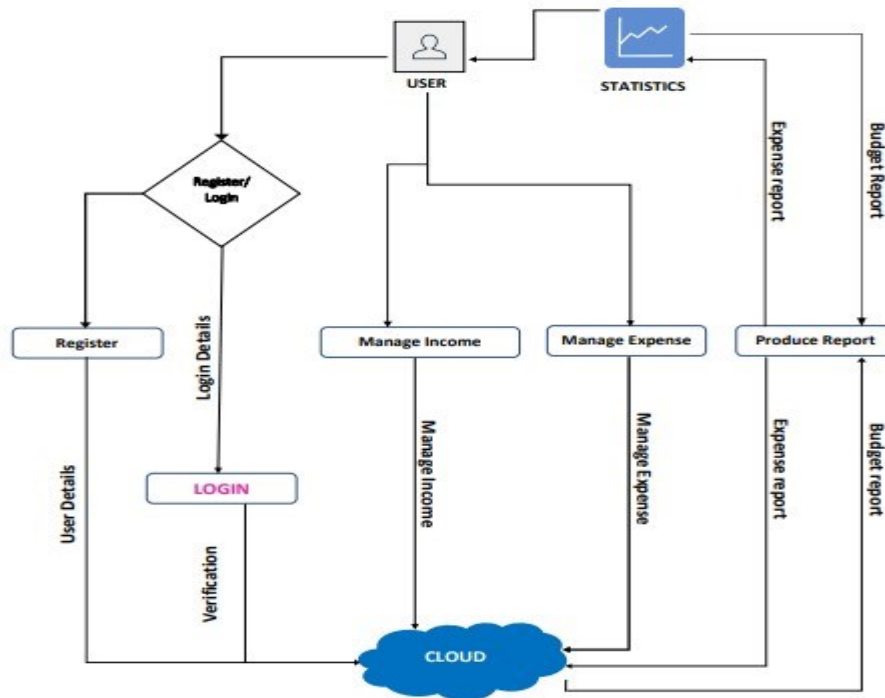
6. SOLUTION AND TECHNICAL ARCHITECTURE.

Date	31 October 2022
Team ID	PNT2022TMID43646
Project Name	Project - Personal Expense Tracker Application
Maximum Marks	4 Marks

Technical Architecture:
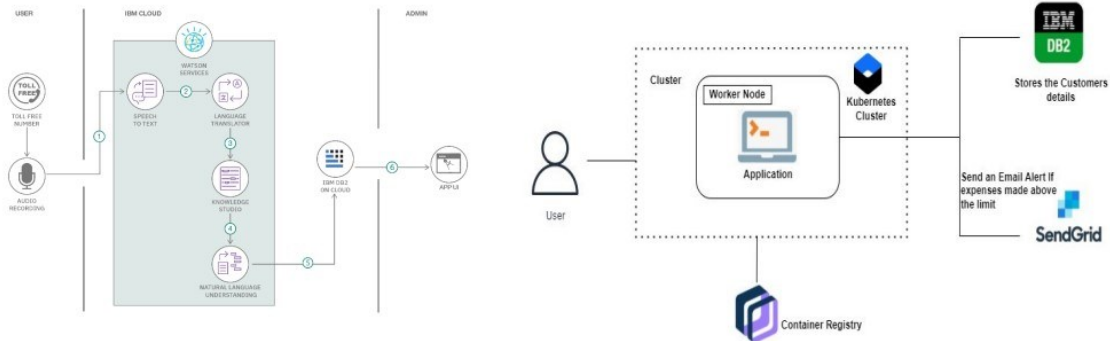
7. USER STORIES

User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Mobile user	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account /dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-2	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard		As a user, I can access my detail, manage the expense, add budget, expense report from the app etc..		High	Sprint-1
Web user	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account /dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-3	As a user, I can access my detail, manage the expense, add budget, expense report from the app etc..		High	Sprint-1
Customer Care Executive	Email or Customer Care no		As a user, I can contact the service administration for the support.	I can solve the Issue.	High	Sprint-3
Administrator	Email or Customer Care no		As a user, I can contact the service administration for the support.	I can solve the Issue.	High	Sprint-1

9.PROJECT PLANNING PHASE.

a. Milestone and Tasks

Project Planning Phase
Project-Personal Expense Tracker Application
Milestone and Activity List

Date	26 October 2022
Team ID	PNT2022TMID43646
Project Name	Personal Expense tracker Application

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	23 OCTOBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	24 OCTOBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	25 OCTOBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	31 OCTOBER 2022

Problem Statement	Prepare problem - statement document.	31 OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	25 OCTOBER 2022

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	10 NOVEMBER 2022
Solution Requirement	Prepare the solution requirement document.	31 OCTOBER 2022
Technology Stack Architecture	Prepare the technology stack architecture diagram.	31 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	10 NOVEMBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	10 NOVEMBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	12 NOVEMBER 2022

9. CODING & SOLUTIONING

a. Feature 1

We have added the data visualiza on methods for expenditure. The pie chart have been used to represent the monthly expenses. The pie chart is a pictorialrepresenta on of data that makes it possible to visualize the rela onshipsbetween the parts and the whole of a variable. For example, it is possible tounderstand the industry count or percentage of a variable level from the division by areas or sectors. The recommended use for pie charts is twodimensional, as three-dimensional use can be confusing

code

```python
1   import re
2   import ibm_db
3   from flask import Flask, render_template, request, session
4
5   global table
6   global userid
7
8
9   def insertTableData(conn, username, email, password, age, profession, table):
10      table = 'no'
11      sql = "INSERT INTO usersdetails(username,email,password,age,profession,table) VALUES ('{}','{}','{}','{}','{}','{}')".format(
12          username, email,
13          password, age, profession, table)
14      out = ibm_db.exec_immediate(conn, sql)
15      print('Number of affected rows : ', ibm_db.num_rows(out), "\n")
16
17
18  def conditionCheak():
19      username = session.get('username', None)
20      sql = "SELECT table FROM usersdetails WHERE username=?"
21      stmt = ibm_db.prepare(conn, sql)
22      ibm_db.bind_param(stmt, 1, username)
23      ibm_db.execute(stmt)
24      out = ibm_db.fetch_assoc(stmt)
25      print("out cheak condition ->", out)
26      value = out['TABLE']
27      session['tablequery'] = value
28
29
```

```python
86   @app.route("/")
87   @app.route("/login", methods=['POST', 'GET'])
88   def login():
89       msg = ''
90
91       if request.method == 'POST':
92           username = request.form['username']
93           password = request.form['password']
94           sql = "SELECT * FROM usersdetails  WHERE username =? AND password=?"
95           stmt = ibm_db.prepare(conn, sql)
96           ibm_db.bind_param(stmt, 1, username)
97           ibm_db.bind_param(stmt, 2, password)
98           ibm_db.execute(stmt)
99           account = ibm_db.fetch_assoc(stmt)
100          print(account)
101
102          table = account['TABLE']
103          userid = account['USERNAME']
104
105          session['username'] = userid
106
107          sql1 = "SELECT username, email, age, profession FROM usersdetails WHERE username=?"
108          stmt_db = ibm_db.prepare(conn, sql1)
109          ibm_db.bind_param(stmt_db, 1, userid)
110          ibm_db.execute(stmt_db)
111          accounts = ibm_db.fetch_assoc(stmt_db)
112
113          if account:
114              session['id'] = account['USERNAME']
```

```
58            stmt = ibm_db.prepare(conn, sql1)
59            ibm_db.bind_param(stmt, 1, email)
60            ibm_db.bind_param(stmt, 2, profession)
61            ibm_db.bind_param(stmt, 3, username)
62            ibm_db.execute(stmt)
63
64
65    def displayDetails(userid):
66        sql1 = "SELECT username, email, age, profession FROM usersdetails WHERE username=?"
67        stmt_db = ibm_db.prepare(conn, sql1)
68        ibm_db.bind_param(stmt_db, 1, userid)
69        ibm_db.execute(stmt_db)
70        accounts = ibm_db.fetch_assoc(stmt_db)
71        return accounts
72
73
74    try:
75        conn = ibm_db.connect(
76            "DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSLServerCertificate=DigiCertGloba
77            "", "")
78        print("Db connected")
79    except:
80        print("Error")
81
82    app = Flask(__name__)
83    app.secret_key = 'aa'
84
85
```

```
29
30    def createNewTableForUser(userid):
31        sqlc = "CREATE TABLE userid=? (date DATE , expensename VARCHAR(22),expenseamount INTEGER,paymode VARCHAR(24),category VARCHAR(22))"
32        stmt = ibm_db.prepare(conn, sqlc)
33        ibm_db.bind_param(stmt,0, userid)
34        createtable = ibm_db.execute(stmt)
35        session['createtable']=createtable
36
37
38    def updateTabletable(table):
39        username = session.get('username', None)
40        sql1 = "UPDATE usersdetails SET TABLE=? WHERE USERNAME=?"
41        stmt = ibm_db.prepare(conn, sql1)
42        ibm_db.bind_param(stmt, 1, table)
43        ibm_db.bind_param(stmt, 2, username)
44        ibm_db.execute(stmt)
45
46
47    def updateTableData(username, password, email, profession):
48        sql = "SELECT * FROM usersdetails  WHERE username =? AND password=?"
49        stmt = ibm_db.prepare(conn, sql)
50        ibm_db.bind_param(stmt, 1, username)
51        ibm_db.bind_param(stmt, 2, password)
52        ibm_db.execute(stmt)
53        account = ibm_db.fetch_assoc(stmt)
54        print(account)
55        if account:
56            username = account['USERNAME']
57            sql1 = "UPDATE usersdetails SET EMAIL=?,PROFESSION=? WHERE USERNAME=?"
58            stmt = ibm_db.prepare(conn, sql1)
```

```python
145                return render_template('login.html')
146        return render_template('registration.html',msg=msg)
147
148
149    @app.route("/add", methods=['POST', 'GET'])
150    def add():
151
152        if request.method == 'POST':
153            username = request.form['username']
154            password = request.form['password']
155            date = request.form['date']
156            expensename = request.form['expensename']
157            expenseamount = request.form['expenseamount']
158            paymode = request.form['paymode']
159            category = request.form['category']
160
161            sql = "SELECT * FROM usersdetails  WHERE username =? AND password=?"
162            stmt = ibm_db.prepare(conn, sql)
163            ibm_db.bind_param(stmt, 1, username)
164            ibm_db.bind_param(stmt, 2, password)
165            ibm_db.execute(stmt)
166            account = ibm_db.fetch_assoc(stmt)
167            print(account)
168            if account:
169                userids = account['USERNAME']
170                sqli = "INSERT INTO expenses(username,date,expensename,expenseamount,paymode,category) VALUES ('{}','{}','{}','{}','{}','{}')".format(
171                    username,
172                    date, expensename, expenseamount, paymode, category)
173                out = ibm_db.exec_immediate(conn, sqli)
```

```python
115                session['username'] = account['USERNAME']
116                msg = 'Logged in successfully !'
117                return render_template('dashboard.html', accounts=accounts)
118            else:
119                msg = 'Incorrect username / password !'
120        return render_template('login.html', msg=msg)
121
122
123    @app.route("/register", methods=['POST', 'GET'])
124    def register():
125        if request.method == "POST":
126            email = request.form['email']
127            password = request.form['password']
128            age = request.form['age']
129            profession = request.form['profession']
130            username = request.form['username']
131            sql = "SELECT * FROM usersdetails WHERE username =?"
132            stmt = ibm_db.prepare(conn, sql)
133            ibm_db.bind_param(stmt, 1, username)
134            ibm_db.execute(stmt)
135            account = ibm_db.fetch_assoc(stmt)
136            print(account)
137            if account:
138                msg = 'Account already exists !'
139            elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
140                msg = 'Invalid email address !'
141            elif not re.match(r'[A-Za-z0-9]+', username):
142                msg = 'name must contain only characters and numbers !'
143            else:
144                insertTableData(conn, username, email, password, age, profession, table)
```

```python
204        if request.method == 'GET':
205            user = session.get('username', None)
206            print(user)
207            expensedetails = []
208            sql = "SELECT CHAR(DATE(date),USA) as date, expensename, expenseamount, paymode, category FROM expenses WHERE username=?"
209            stmt = ibm_db.prepare(conn, sql)
210            ibm_db.bind_param(stmt, 1, user)
211            ibm_db.execute(stmt)
212            details = ibm_db.fetch_assoc(stmt)
213            while details != False:
214                expensedetails.append(details)
215                details = ibm_db.fetch_assoc(stmt)
216
217            print(expensedetails)
218
219            sql2 = "SELECT SUM(expenseamount) AS TOTALVAL FROM expenses WHERE username = ?"
220            stmt2 = ibm_db.prepare(conn, sql2)
221            ibm_db.bind_param(stmt2, 1, user)
222            ibm_db.execute(stmt2)
223            totalexpense = ibm_db.fetch_assoc(stmt2)
224            print(totalexpense)
225            return render_template('dispexpense.html', expensedetails=expensedetails, totalexpense=totalexpense['TOTALVAL'])
226
227
228    @app.route('/logout')
229    def logout():
230        session.pop('loggedin', None)
231        session.pop('id', None)
232        session.pop('username', None)
233        return render_template('login.html')
```

```python
174                accounts = displayDetails(userids)
175                return render_template('dashboard.html', accounts=accounts)
176            return render_template('add.html')
177
178            if request.method == 'GET':
179                return render_template('dashboard.html', accounts=accounts)
180
181
182    @app.route("/changedetails", methods=['POST', 'GET'])
183    def changedetails():
184        if request.method == "POST":
185            username = request.form['username']
186            password = request.form['password']
187            email = request.form['email']
188            profession = request.form['profession']
189            updateTableData(username, password, email, profession)
190            return render_template('login.html')
191
192        return render_template('changedetails.html')
193
194
195    @app.route("/dashboard",methods=['POST','GET'])
196    def dashboard():
197        username = session.get('username',None)
198        accounts = displayDetails(username)
199        return render_template('dashboard.html',accounts=accounts)
200
201
202    @app.route("/dispexpense", methods=['POST', 'GET'])
203    def dispexpense():
```

```python
223            totalexpense = ibm_db.fetch_assoc(stmt2)
224            print(totalexpense)
225            return render_template('dispexpense.html', expensedetails=expensedetails, totalexpense=totalexpense['TOTALVAL'])
226
227
228    @app.route('/logout')
229    def logout():
230        session.pop('loggedin', None)
231        session.pop('id', None)
232        session.pop('username', None)
233        return render_template('login.html')
234
235
236    if __name__ == '__main__':
237        app.run(host='0.0.0.0', debug=True)
```

b. Feature 2

Email no fica ons will be sent to the users once they cross the expenditure limit through send grid mail system. Most no fica ons are transac onal, meaning a recipient's ac on or account ac vity triggers them. But some no fica ons are marke ng related,

encouraging the recipient to take a specific ac on. Ecommerce product no fica ons inform recipients about new products or discounts. Plus, unlike general marke ng emails, these are highly personalized and focus on a single product. For example, if a customer views an item on your website and that item goes on sale, you can send the customer a no fica on to let them know this is the best me to buy. Users can also opt into receiving no fica ons when an outofstock item is back in stock. No fica on emails tend to perform well because the content is highly relevant to the recipient. But the only way for the recipient to know this is if you state the content clearly in the subject line.

10. TESTING   a.

Test Cases

					Date	19 November 2022							
					Team ID	PNT2022TMID26057							
					Project Name	Personal Expense Tracker Application							
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comment	TC for Atom / BUG ID	Executed By
LoginPage_TC_OO1	Functional	Login Page	Verify user is able to see t e Login/Signup popup when user clicked on My account button	None	1. Go to website 2.Login page appears	Username: Prithivi password: qwerty	Login/Signup popup should display	Workingas expected	Pass	-		Prithivi Krishna
LoginPage_TC_OO2	UI	Login Page	Verify the UI elements in Login/Signup	Home	1. Go to website 2. Enter details and click login	Username: Prithivi password: qwerty	Application should show below UIelements: a. username text box b. password text box c. Login button d. New customer? register	Workingas expected	Pass	-		Vikranam
LoginPage_TC_OO3	Functional	Login page	Verify user is able to log into application with Valid credentials	Username & password	1. Go to website 2. Enter details and click login	Username: Prithivi password: qwerty	User should navigate to user accounthomepage	Workingas expected	Pass	-		Yeshwanth
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with Invalid credentials.	Username & password	1. Go to website 2. Enter details and click login	Username: Prithivi password: 123456	Application should show 'Incorrect username or password ' validation message.	Workingas expected	Pass	-		Chandra Mouli
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: Prithivi password: qwerty	Application should show 'Incorrect username or password ' validation message.	Workingas expected	Pass	-		Prithivi & Yeshwanth
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: Prithivi password: qwerty	Application should show 'Incorrect username or password ' validation message.	Workingas expected	Pass	-		Chandra Mouli
Add Expense Page	Functional	Add Expense page	Verify whether user is able to add expense or not	Have some expense to add	1. Add date, expense name and other details.. 2. Check if the expense gets added	add expense = 550	Application adds expenses	Workingas expected	Pass	-		Vikraman

b. User Acceptance Testing

Acceptance Testing
UAT Execution & Report Submission

Date	20 november 2022
Team ID	PNT2022TMID43646
Project Name	Project – Personal Expense Tracker Application

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Personal Expense Tracker Application project at the time of the release to User AcceptanceTesting (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	6	0	2	4	12
Duplicate	1	0	0	0	1
External	2	0	0	1	3
Fixed	11	2	2	10	25
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	5	2	1	8
Totals	20	7	6	16	49

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
Print Engine	7	0	0	7
Client Application	15	0	0	15
Security	2	0	0	2

Outsource Shipping	0	0	0	6
Exception Reporting	3	0	0	3
Final Report Output	5	0	0	5
Version Control	1	0	0	1

11. RESULTS

a. Performance Metrics

Tracking income and expenses: Monitoring the income and tracking all expenditures (through bank accounts, mobile wallets, and credit & debit cards).

12. ADVANTAGES & DISADVANTAGES

*Advantages

It is easy to set up and use. It keeps track of everything for you in real me using an automa c app. It has a wealth of informa on, so any data you believe is crucial to track is there and at your disposal; you only need to look to see it.

Everything has a simple user interface as well. There is a tab or an op on available for you whether you want to create a budget, keep track of a specific sort of spending, or review your financial history .It happens instantly. Your data will be tracked for you by the applica on. In contrast to what you might manually perform, it doesn't do it once a week or once a month.

*Disadvantage

Your data is likely being exploited and sold, and it is less secure. If the product is a free service, then you are the product. Like other financial apps, Mint.com offers its services for nothing. No ma er what their privacy policy may or may not say, just expect that someone, somewhere is going to record and analyse your spending pa erns because they need to pay their expenses. Now, you shouldn't have to worry about iden ty the or credit card fraud since these firms are big enough and safe enough to prevent those things from happening.

13. CONCLUSION

You can save money by keeping track of your daily costs, but it can also help you set future financial goals. If possible, look for areas where you may cut costs and nego ate be er terms if you know exactly where your money is going. The Expense Tracker project will help us keep track of our everyday spending and make a record of it. Compared to other income and expense trackers, the project we designed is more effec ve. The project successfully avoids manual calcula on by a emp ng to avoid determining the salary and expense each month. It's an intui ve applica on.

14. FUTURE SCOPE

1. It will have a variety of record-keeping choices (such as food, travel expenses, salary, etc.).
2. It will continue to give updates about our daily spending automatically. 3. Despite being in a haste to make money in today's hec c and expensive world, we eventually gave up. As we naively waste money on unnecessary items and tles. We so came over with the inten on of following our profit.
4) The user can specify their own expense categories here, such as those for food, clothing, rent, and bills, where they must input the money that has been spent and may also add addi onal informa on to denote the expense.

15.APPENDIX

https://drive.google.com/file/d/1wPfPBbJmMHsfwLPZvpknDgAOnebLy5-b/view?usp=drivesdk