**CHAPTER 1**

**INTRODUCTION**

The introduction briefs about evolution of fabric defect detection in industries and recent approaches.

**1.1 Project Overview**

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.

**1.2 Purpose**

When you track your spending, you know where your money goes and you can ensure that your money is used wisely. Tracking your expenditures also allows you to understand why you're in debt and how you got there. This will then help you design a befitting strategy of getting out of debt.

Budgeting ensures you're not spending more than you're making, allowing you to plan for short- and long-term expenses. It's an easy, helpful way for people with all types of income and expenses to keep their finances in order.

**CHAPTER 2**

**LITERATURE SURVEY**

Literature review was carried out to gain knowledge and improve the skills needed to complete this project. This chapter shows the different techniques that have been implemented.

**2.1 Existing Problem**

An expense tracker is a software or application that helps to keep an accurate record of your money inflow and outflow. Many people in India live on a fixed income, and they find that towards the end of the month they don't have sufficient money to meet their needs.

**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] Developer.android.com

[6] http://www.appbrain.com/app/expensemanager/ com.expensemanager

[7] https://www.xpenditure.com/en?

[8] http://expense-manager.com/how-expense software/

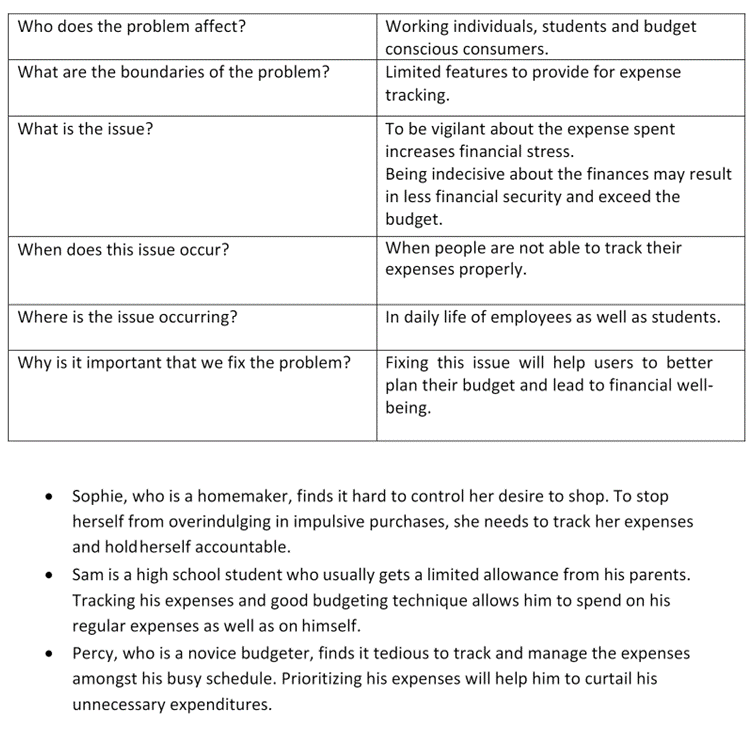
[9] Donn Felker, “Android Application Development for Dummies”, published by For Dummies, 2010.

[10] Ed Burnette, “Hello, Android: Introducing Google's Mobile Development Platform”, published by Pragmatic Bookshelf, 2009.

Lee, “Beginning Android Application Development”, Published by WroxPress, 2011.

**2.3 Problem Statement Definition**

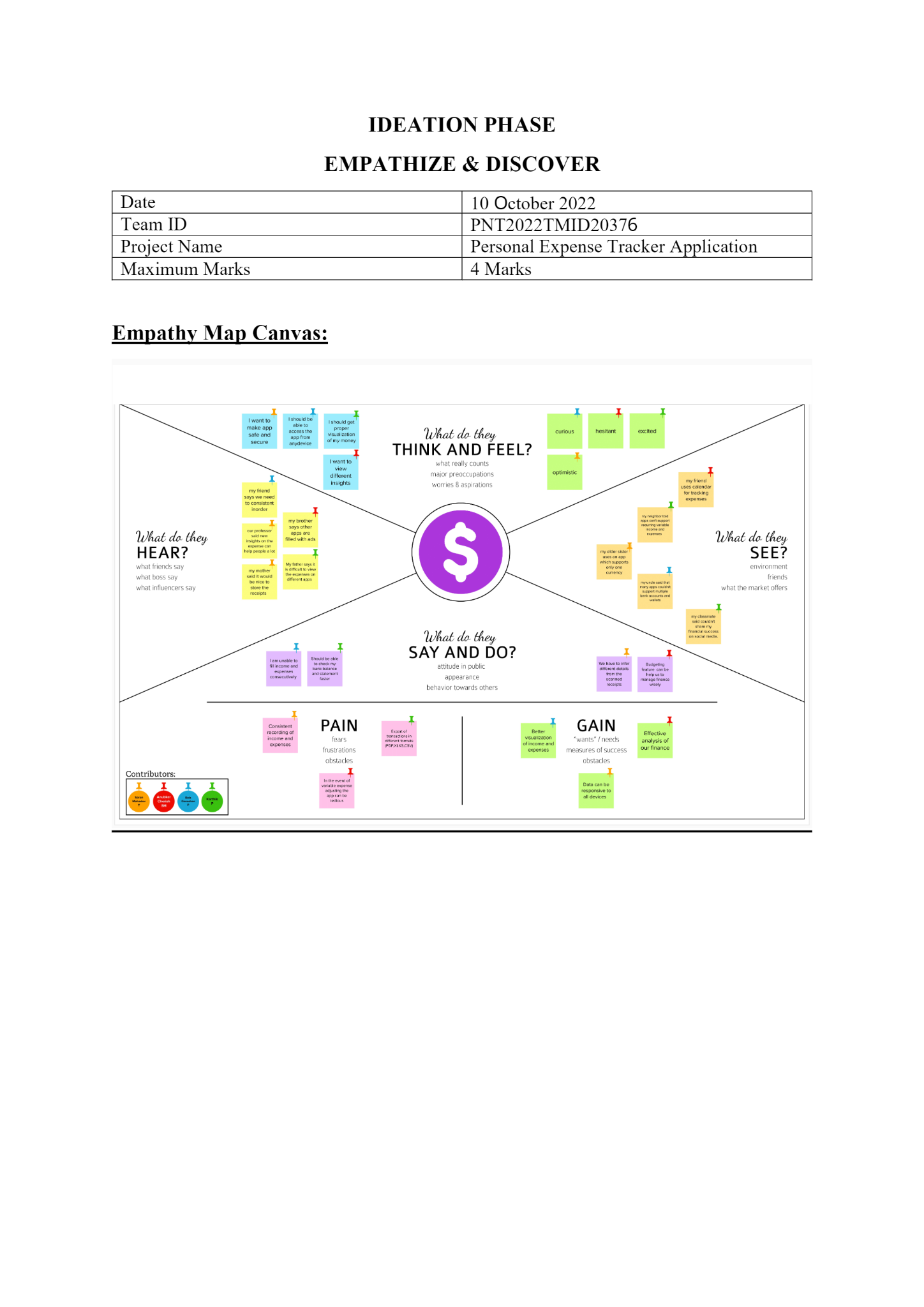
It is tough to keep track of all the financial decisions and activities that a person makes. Traditional expense tracking methods are inconvenient and unreliable. In order to get a quick overview about your total incomes and expenses and control spending , its convenient to digitize the process by having a personal expense tracker.



**CHAPTER 3**

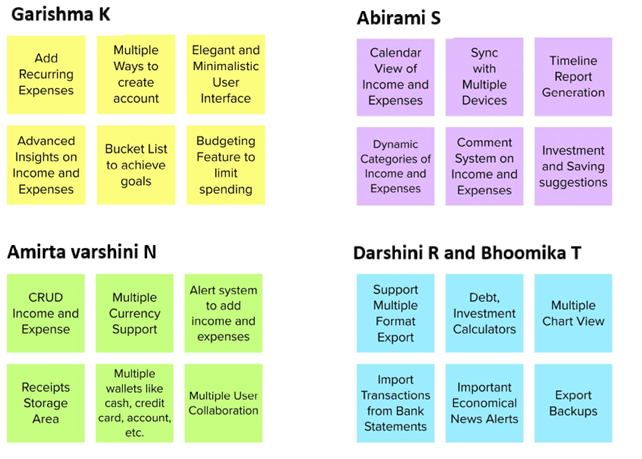
**IDEATION & PROPOSED SOLUTION**

**3.1 Empathy Map Canvas**

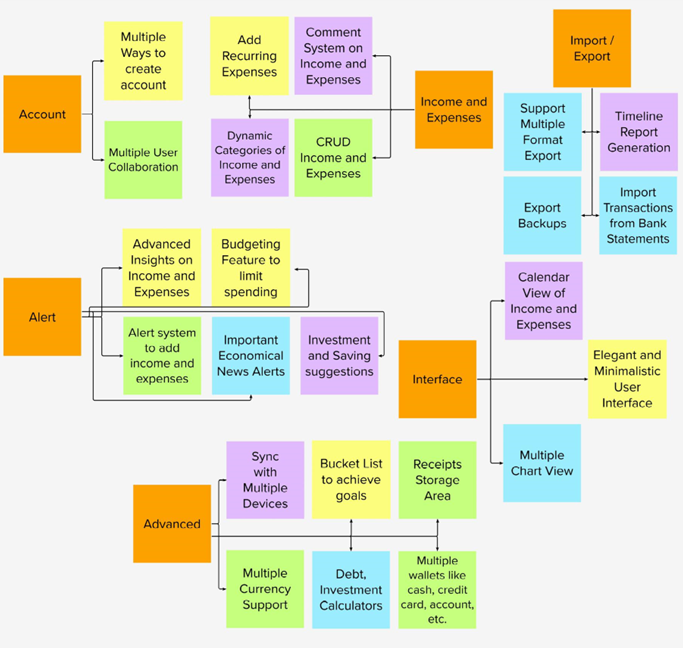
****

**3.2 Ideation & Brainstorming**

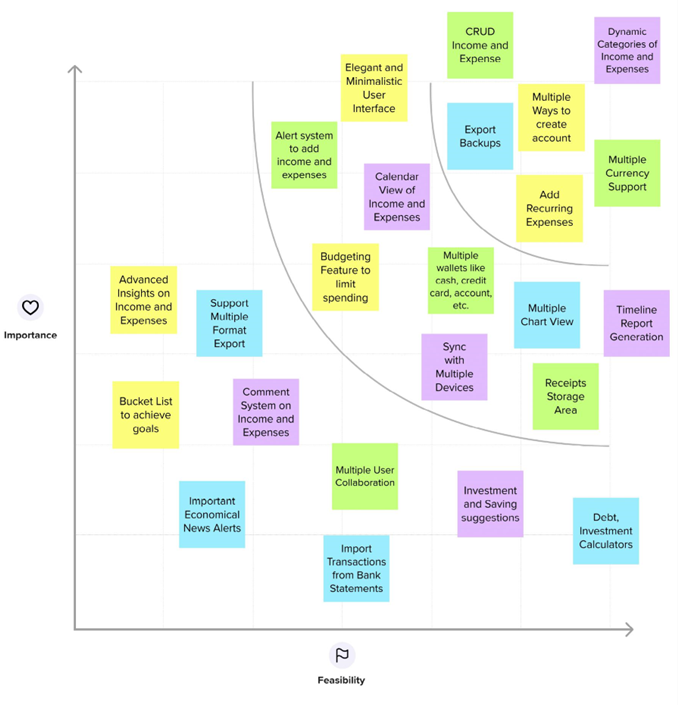
**3.2.a Brainstorm**

****

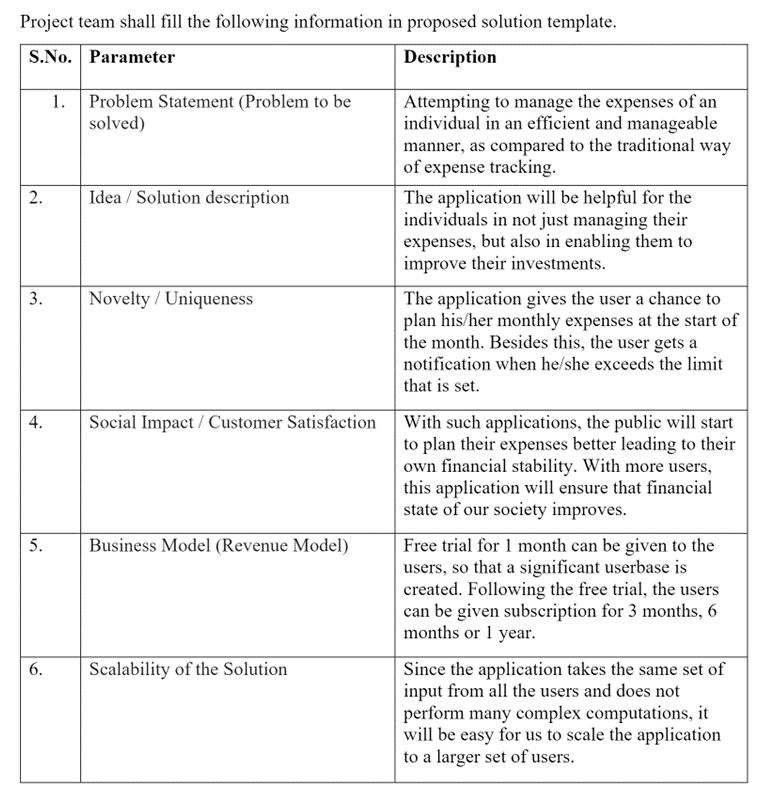
**3.2.b Group Ideas**

****

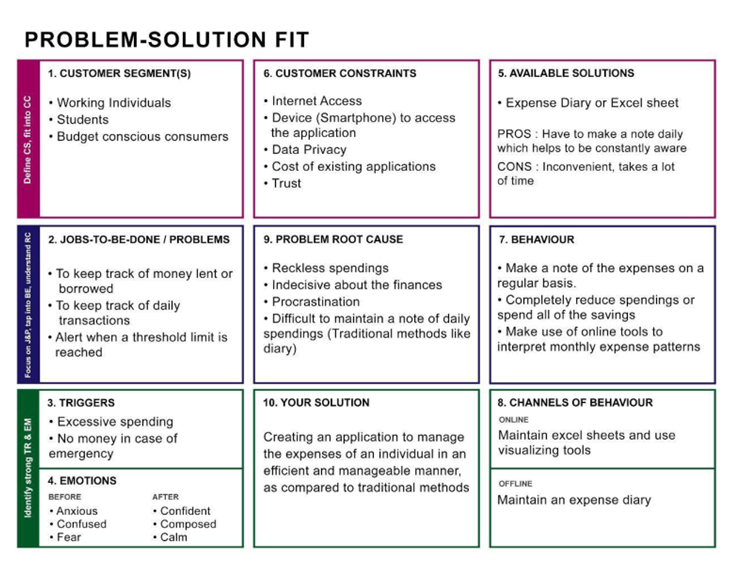
**3.2.c Prioritize**

****

**3.3 Proposed Solution**

****

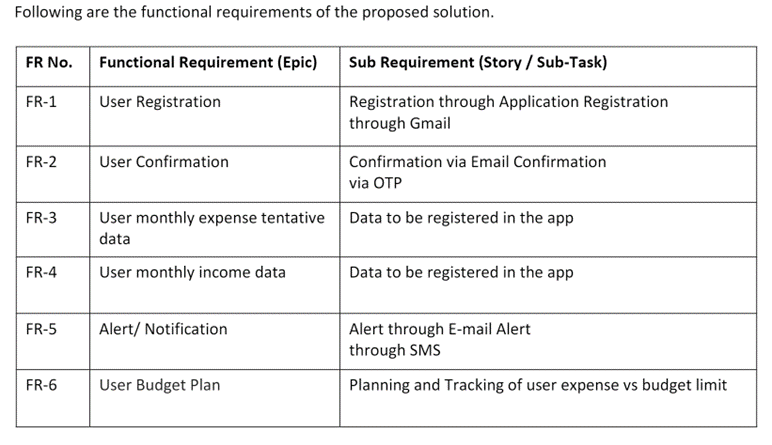
**3.4 Problem Solution fit**

****

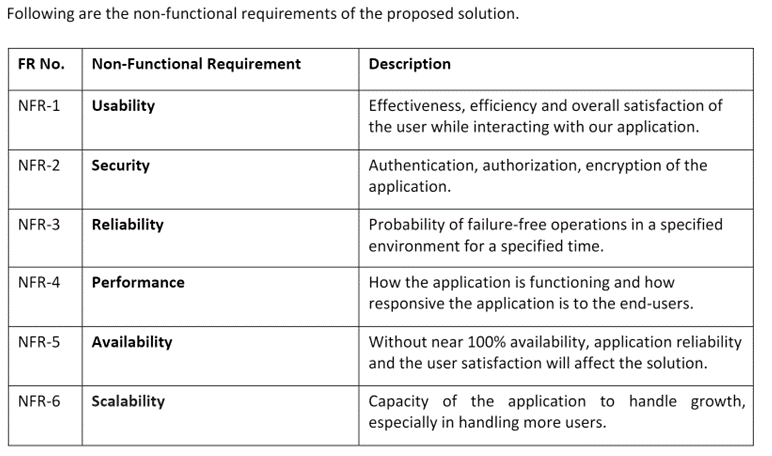
**CHAPTER 4**

**REQUIREMENT ANALYSIS**

**4.1 Functional Requirements**

****

**4.2 Non Functional Requirements**

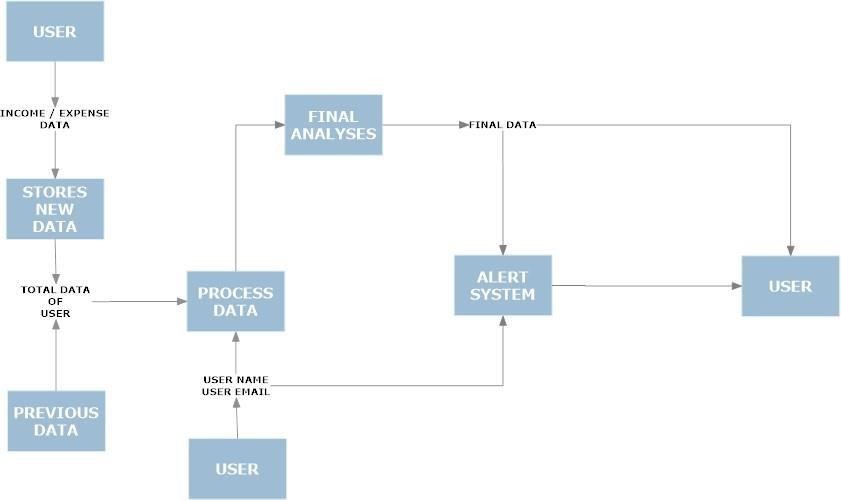
****

**CHAPTER 5**

**PRODUCT DESIGN**

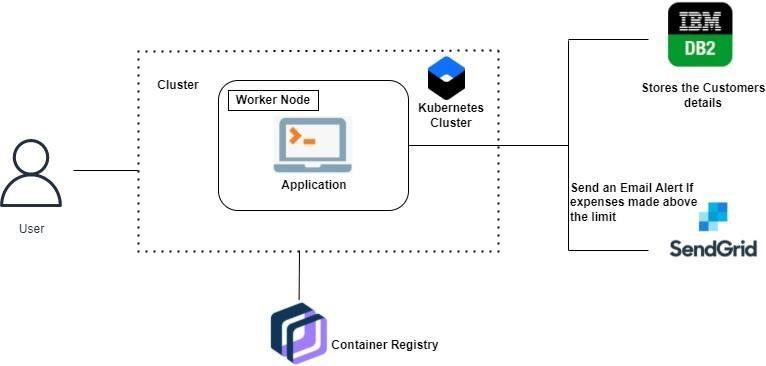
**5.1 Data Flow Diagrams**

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



**5.2Technical Architecture**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



**5.3 User Stories**

Usethe below template to list all the user stories of the product.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer  (Mobileuser & 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 |  |
|  |  | 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 |  |
|  |  | USN- 3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low |  |
|  | Login | USN -4 | As a user, I can log into the application by entering email & password | I can access the application | High |  |
|  | Dashboard | USN -5 | As a user I can enter my income and expenditure details. | I can view my daily expenses | High |  |
| Customer Care Executive |  | USN 6 | As a customer care executive I can solve the log in issues and other issues of the application. | I can provide support or solution at any time 24\*7 | Medium |  |
| Administrator | Application | USN -7 | As a administrator I can upgrade or update the application. | I can fix the bug which arises for the customers and users of the application | Medium |  |

**CHAPTER 6**

**PROJECT PLANNING & SCHEDULING**

**6.1Sprint Planning & Estimation**

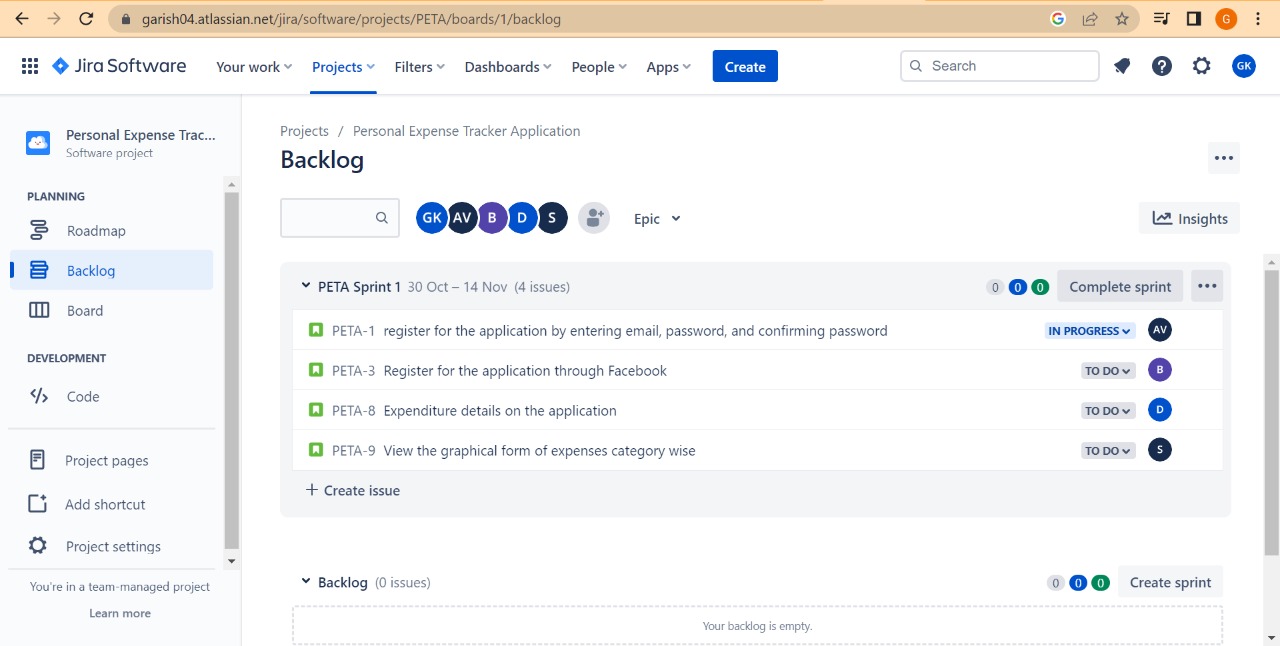
Use the below template to create product backlog and sprint schedule.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional requirement (Epic)** | **User story number** | **User story/task** | **Story points** | **priority** | **Team members** |
| Sprint-1 | Registration | USN-1 | Register for the application by entering email,password, and confirming password. | 2 | High | Amirta Varshini |
| Sprint-1 |  | USN-2 | As a user will receive confirmation email once have registered for the application | 1 | High | Amirta  Varshini |
| Sprint-2 |  | USN-3 | Register for the application through Facebook | 2 | Low | Bhoomika |
| Sprint-1 |  | USN-4 | User register for the application through Gmail | 2 | Medium | Garishma |
| Sprint-1 | Login | USN-5 | User log into the application by entering email & password | 1 | High | Garishma |
| Sprint-3 | Dashboard | USN-6 | Expenditure details on the application | 3 | High | Darshini |
| Sprint-3 | Limits | USN-6 | User can set monthly expense limit so that receive a mail on exceeding | 4 | High | Darshini |
| Sprint-4 | Reports | USN-6 | View the graphical form of expenses category wise | 5 | Medium | Abirami |

**6.2 Sprint Delivery Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points**  **Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| Sprint-1 | 6 | 6 Days | 15 Oct 2022 | 30 Oct 2022 | 6 | 30 Oct 2022 |
| Sprint-2 | 2 | 6 Days | 30 Oct 2022 | 05 Nov 2022 | 2 | 05 Nov 2022 |
| Sprint-3 | 7 | 6 Days | 05 Nov 2022 | 12 Nov2022 | 7 | 12 Nov 2022 |
| Sprint-4 | 5 | 6 Days | 12 Nov2022 | 19 Nov2022 | 5 | 1. ov 2022 |

**6.3 Reports from JIRA**



**CHAPTER 7**

**7.1 Feature 1**

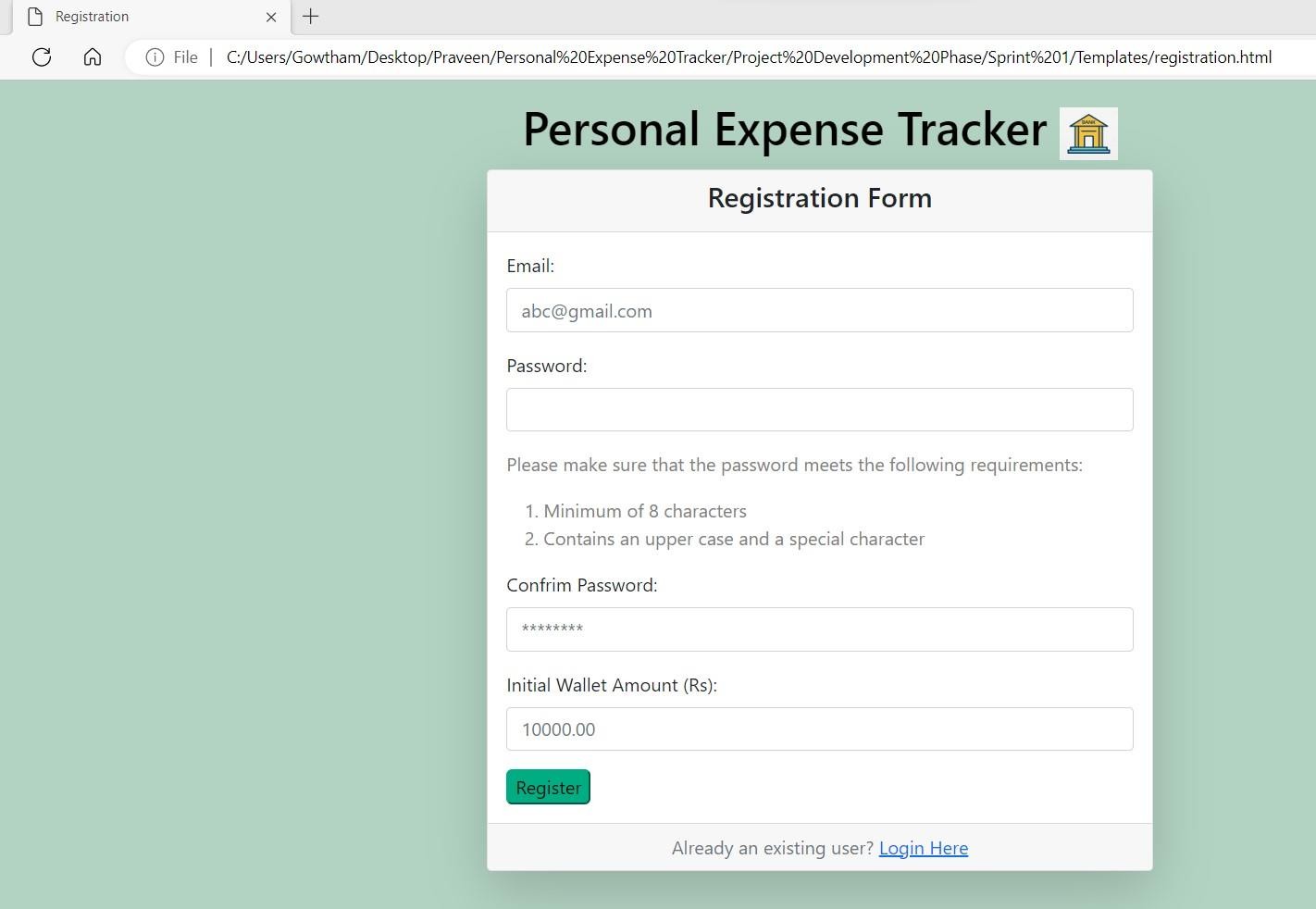
1. Expense and revenue tracking.
2. Managing transaction receipts and records.
3. Paying taxes in time.
4. Processing payment and invoices.
5. Create in-depth reports.

**CHAPTER 8**

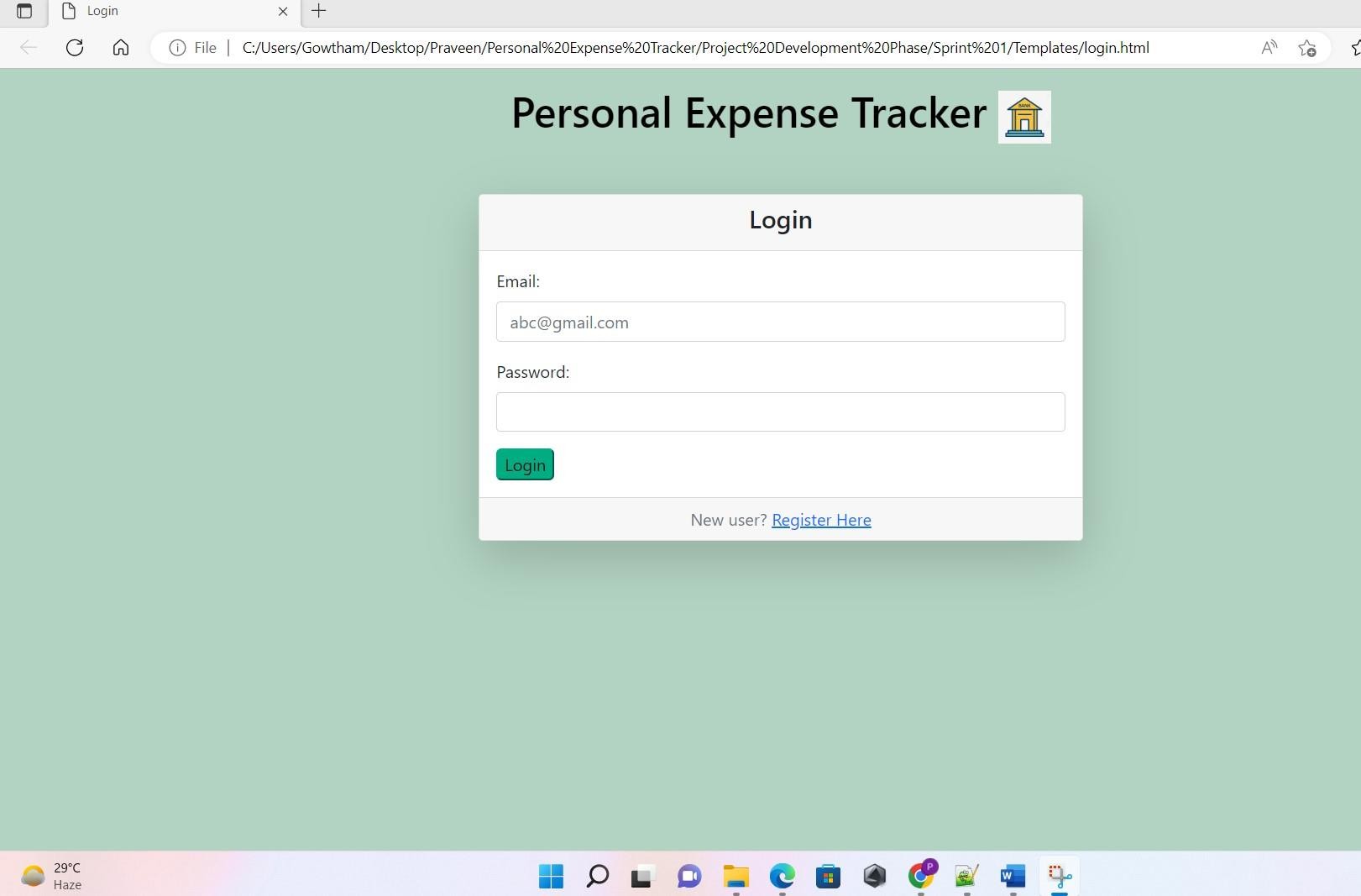
**TESTING**

**8.1 Test Cases**

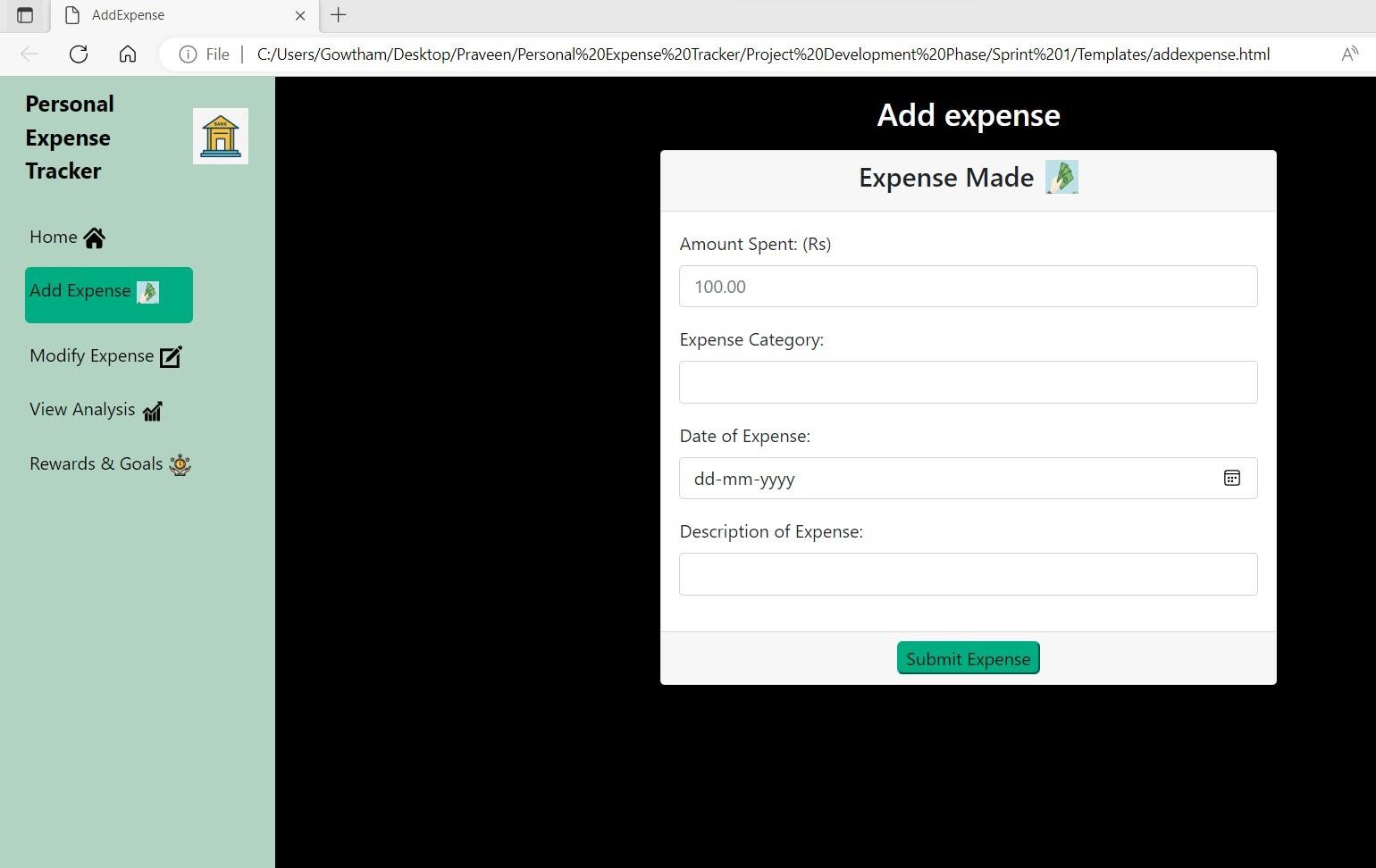
Registration Page



**LOGIN**



**ADD EXPENSE**



**CHAPTER 9**

**RESULTS**

**9.1Performance Metrics**

An application can be a very powerful tool for businesses if once the app becomes a success. However, the success of an app is measured through numbers, metrics, and analytics. Developing an app takes quite a lot, so once you’ve dedicated much time, money, and effort to the process, it’s mandatory to measure mobile app performance.



**CHAPTER 10**

**ADVANTAGES & DISADVANTAGES**

**10.1 Advantages**

* Improved customer service
* Cloud-based solution
* Order Fulfillment
* Harness Customer Loyalty and Retention

**10.2 Disadvantages**

* System Clash
* Reduced Physical Audits
* No solution to improve or eliminate bottlenecks in the service cycle

**CHAPTER 11**

**CONCLUSION**

Taking proper care of our record is crucial in every business, no matter how big or little, we must understand. We must educate ourselves about the idea of effective inventory management and its applications because we can see that managers do not fully grasp it. A company's inventory management system is one of the reasons for its failure. Many customs to combat failure are present, and we can start from this point. Modern technologies can support us in managing and keeping an eye on our inventory. We may learn, put new ideas into practice, and assess our company**.**

**CHAPTER 12**

**FUTURE SCOPE**

1)It will have various options to keep record (for example Food, Travelling Fuel, Salary etc.).

2) Automatically it will keep on sending notifications for our daily expenditure.

3) In today’s busy and expensive life, we are in a great rush to make moneys, but at the end of the month we broke off. As we are unknowingly spending money on title and unwanted things. So, we have come over with the plan to follow our profit.

4) Here user can define their own categories for expense type like food, clothing, rent and bills where they have to enter the money that has been spend and likewise can add some data in extra data to indicate the expense.

**CHAPTER 13**

**APPENDIX**

**SOURCE CODE:**

Registration Page

<!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](https://cdn.jsdelivr.net/npm/bootstrap%405.0.2/dist/css/bootstrap.min.css)" rel="stylesheet" integrity="sha384- EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>Registration</title>

</head>

<script src="[https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js](https://cdn.jsdelivr.net/npm/bootstrap%405.0.2/dist/js/bootstrap.bundle.min.js)" integrity="sha384- MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<body style="background-color:#B2D3C2">

<div class="container mt-3">

<h1 style="color: black; text-align: center;">

Personal Expense Tracker <img src="bank.png" style="width:50px;height: 45px;">

</h1>

<div class="container mt-2" style="width: 600px;">

<div class="card shadow-lg bg-white rounded">

<div class="card-header" style="text-align: center;">

<h4>Registration Form</h4>

</div>

<div class="card-body">

<form action="/" method="POST">

<div class="mb-3">

<label for="email" class="form-label">Email: </label>

<input type="email" class="form-control" name="email" id="email" placeholder="[abc@gmail.com](mailto:abc@gmail.com)">

</div>

<div class="mb-3">

<label for="passowrd" class="form-label">Password: </label>

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

<p style="color: gray;" class="mt-3">Please make sure that the password meets the following requirements:</p>

<ol style="color: gray;"><li>Minimum of 8 characters</li><li>Contains an upper case and a special character</li></ol>

</div>

<div class="mb-3">

<label for="confirmpassword" class="form-label">Confrim Password:

</label>

<input type="password" class="form-control" name="confirmpassword"

id="confirmpassword" placeholder="\*\*\*\*\*\*\*\*">

</div>

<div class="mb-3">

<label for="wallet" class="form-label">Initial Wallet Amount (Rs):

</label>

<input type="number" class="form-control" name="wallet" id="wallet"

placeholder="10000.00">

</div>

<button type="submit" style="background-color:#00AD83; border- color:#00AD83; border-radius:5px;">Register</button>

</form>

</div>

<div class="card-footer text-muted" style="text-align:center">

Already an existing user? <span><a href="login">Login Here</a></span>

</div>

</div>

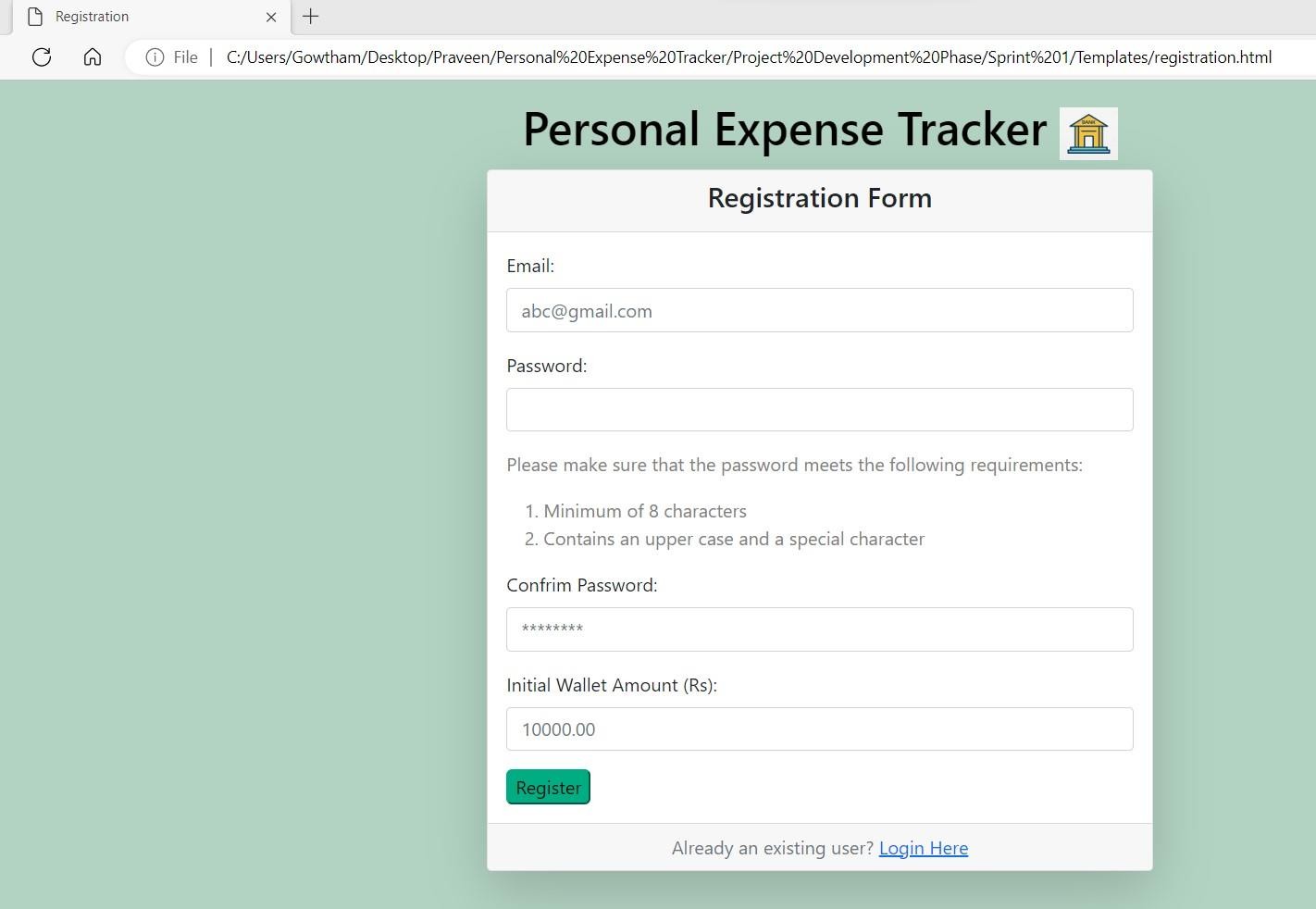
</div>

</div>

</body>

</html>

**OUTPUT:**



# LOGIN

<!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"](https://cdn.jsdelivr.net/npm/bootstrap%405.0.2/dist/css/bootstrap.min.css) rel="stylesheet" integrity="sha384- EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>Login</title>

</head>

<script src="[https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"](https://cdn.jsdelivr.net/npm/bootstrap%405.0.2/dist/js/bootstrap.bundle.min.js) integrity="sha384- MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<body style="background-color:#B2D3C2">

<div class="container mt-3">

<h1 style="color: black; text-align: center;">

Personal Expense Tracker <img src="bank.png" style="width:50px;height: 50px;">

</h1>

<div class="container mt-5" style="width: 600px;">

<div class="card shadow-lg bg-white rounded">

<div class="card-header" style="text-align: center;">

<h4>Login</h4>

</div>

<div class="card-body">

<form action="/login" method="POST">

<div class="mb-3">

<label for="email" class="form-label">Email: </label>

<input type="email" class="form-control" name="email" id="email" placeholder="[abc@gmail.com"](mailto:abc@gmail.com)>

</div>

<div class="mb-3">

<label for="passowrd" class="form-label">Password: </label>

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

</div>

<button type="submit" style="background-color:#00AD83; border- color:#00AD83; border-radius:5px;">Login</button>

</form>

</div>

<div class="card-footer text-muted" style="text-align:center">

New user? <span><a href="registration.html">Register Here</a></span>

</div>

</div>

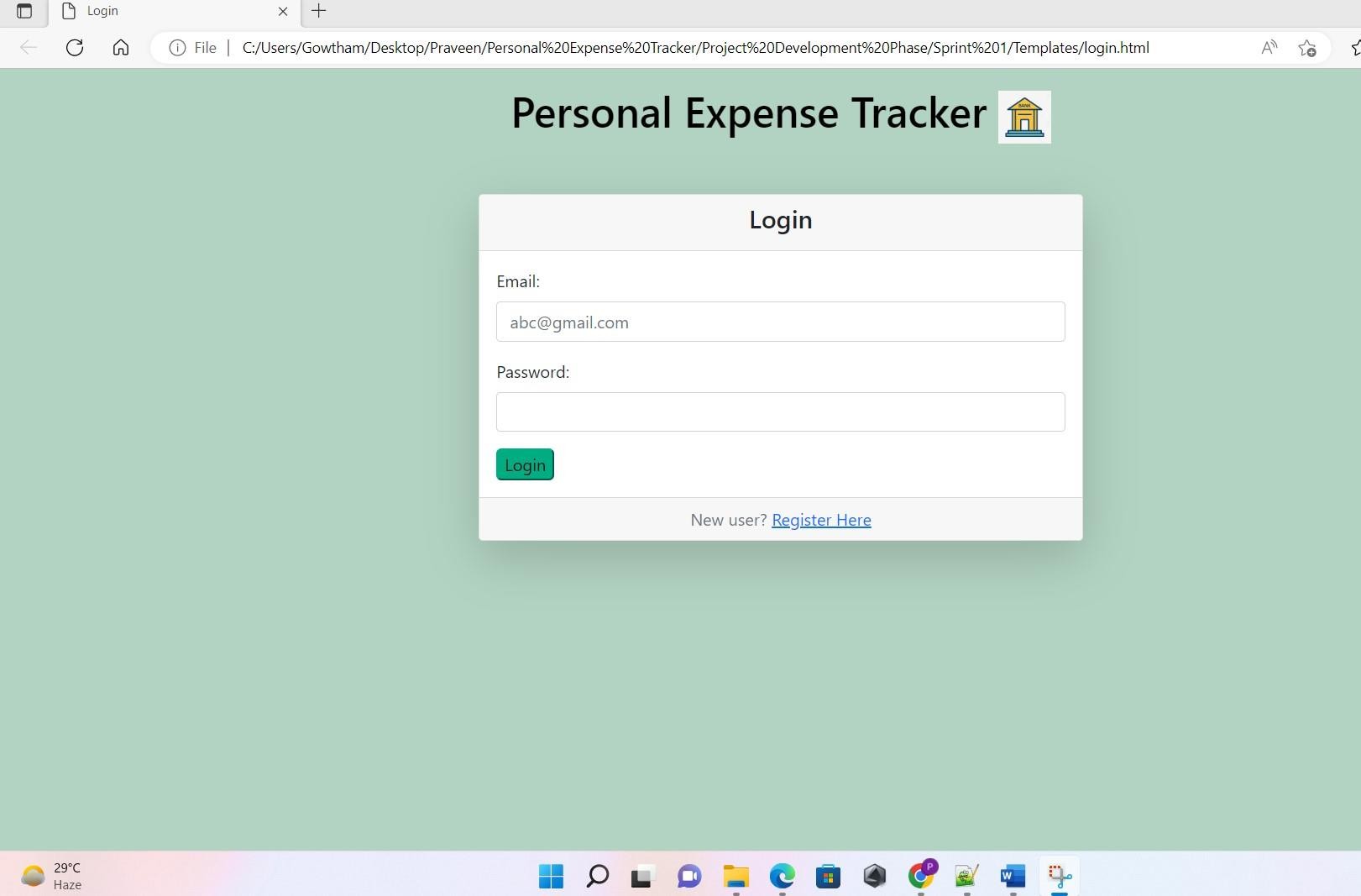
</div>

</div>

</body>

</html>

# OUTPUT:



**Add Expense**

!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](https://cdn.jsdelivr.net/npm/bootstrap%405.0.2/dist/css/bootstrap.min.css)" rel="stylesheet" integrity="sha384- EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>AddExpense</title>

</head>

<script src="[https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js](https://cdn.jsdelivr.net/npm/bootstrap%405.0.2/dist/js/bootstrap.bundle.min.js)" integrity="sha384- MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<div class="container-fluid" >

<div class="row flex-nowrap">

<div class="col-auto col-md-3 col-xl-2 px-sm-2 px-0" style="background-color: #B2D3C2">

<div class="d-flex flex-column align-items-center align-items-sm-start px-3 pt-2 min-vh-100" style="color:black">

<p class="d-flex align-items-center pb-3 mb-md-0 me-md-auto text-white text- decoration-none">

<span class="fs-5 d-none d-sm-inline" style="color:black; font-weight: bold;">Personal Expense Tracker</span>

<img src="bank.png" style="width:50px;height: 50px;">

</p>

<ul class="nav nav-pills flex-column mb-sm-auto mb-0 align-items-center align- items-sm-start" id="menu">

<li class="nav-item mt-2">

<a href="dashboard.html" class="nav-link align-middle px-0" style="color:black;">

<span class="ms-1 d-none d-sm-inline">Home</span><img src="house- outline.svg" style="width:20px;height:20px;margin-left: 5px;">

</a>

</li>

<li class="nav-item mt-2" style="background-color:#00AD83; height: 50px; width: 150px; border-radius: 5px;" >

<a href="addexpense.html" class="nav-link px-0 align-middle" style="color:black;">

<span class="ms-1 d-none d-sm-inline">Add Expense</span><img src="pay.png" style="width:20px;height:20px;margin-left: 5px;">

</a>

</li>

<li class="nav-item mt-2">

<a href="modifyexpense.html" class="nav-link px-0 align-middle" style="color:black;">

<span class="ms-1 d-none d-sm-inline">Modify Expense</span><img src="edit\_icon.svg" style="width:20px;height:20px;margin-left: 5px;">

</a>

</li>

<li class="nav-item mt-2">

<a href="analysis.html" class="nav-link px-0 align-middle" style="color:black;">

<span class="ms-1 d-none d-sm-inline">View Analysis</span><img src="graph.png" style="width:20px;height:20px;margin-left: 5px;">

</a>

</li>

<li class="nav-item mt-2">

<a href="rewards.html" class="nav-link px-0 align-middle" style="color:black;">

<span class="ms-1 d-none d-sm-inline">Rewards & Goals</span><img src="reward.png" style="width:20px;height:20px;margin-left: 5px;">

</a>

</li>

</ul>

</div>

</div>

<div class="col py-3" style="background-color:black">

<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">

<div class="card-header" style="text-align: center;">

<span style="display:inline-flex"><h4>Expense Made</h4><img src="pay.png" style=" margin-left:10px; width:30px; height:30px"></span>

</div>

<div class="card-body">

<form>

<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">

</div>

<div class="mb-3">

<label for="expensecategory" class="form-label">Expense Category:

</label>

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

id="expensecategory"></input>

</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"></input>

</div>

</label>

<div class="mb-3">

<label for="description" class="form-label">Description of Expense:

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

id="description"></input>

</div>

</form>

</div>

<div class="card-footer text-muted" style="text-align:center">

<button type="submit" style="background-color:#00AD83; border- color:#00AD83; border-radius:5px;">Submit Expense</button>

</div>

</div>

</div>

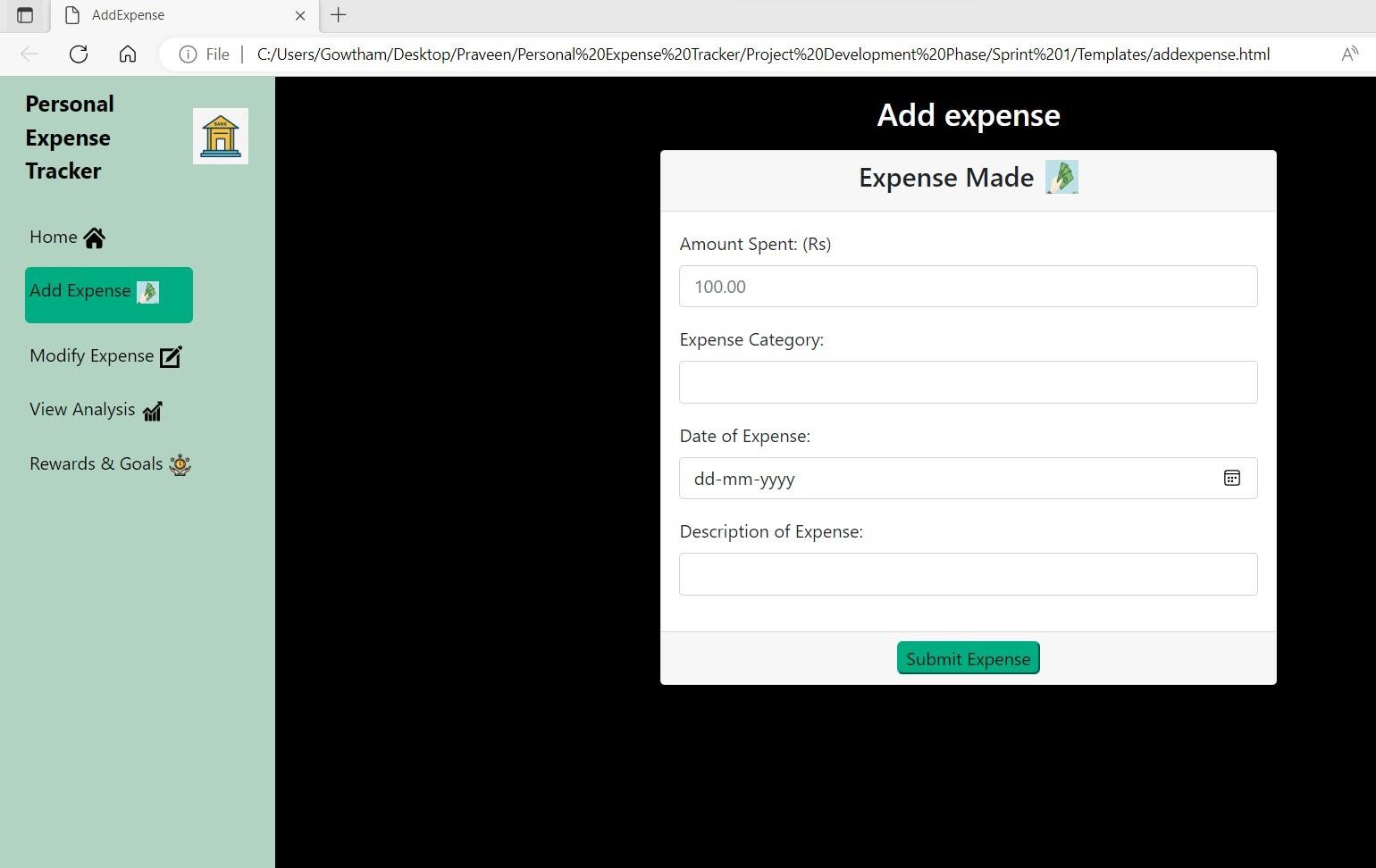
</div>

</div>

</div>

</html>

**Output:**



The python code to Connect with DB

import ibm\_db dictionary={}

def printTableData(conn):

sql = "SELECT \* FROM userdetails" out = ibm\_db.exec\_immediate(conn, sql) document = ibm\_db.fetch\_assoc(out) while document != False:

dictionary.update({document['USERNAME']:

documen t['PASSWORD']}) document = ibm\_db.fetch\_assoc(out) def

insertTableData(conn,rollno,username,email,pass word): sql="INSERT INTO userdetails(rollno,username,email,password) VALUES ({},'{}','{}','{}')".format(rollno,username,ema il,password) out = ibm\_db.exec\_immediate(conn,sql)

print('Number of affected rows : ',ibm\_db.num\_rows(out),"\n") def updateTableData(conn,rollno,username,email,pass word):

sql = "UPDATE userdetails SET (username,email,password)=('{}','{}','{}') WHERE rollno={}".format(username,email,password,rolln o)

out = ibm\_db.exec\_immediate(conn, sql)

print('Number of affected rows : ', ibm\_db.num\_rows(out), "\n") def deleteTableData(conn,rollno):

sql = "DELETE FROM userdetails WHERE rollno={}".format(rollno)

out = ibm\_db.exec\_immediate(conn, sql) print('Number of affected rows : ', ibm\_db.num\_rows(out), "\n")

try:

conn=ibm\_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0- a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;Security= SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=ctb99199;PWD=GybYxLw1rH z86oSh;","","")

print("Db connected") except:

print("Error")

from flask import Flask,render\_template,request,url\_for,session app=Flask( name )

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

if request.method=="POST":

printTableData(conn) username=request.form['username'] password=request.form['password']

try: if dictionary[username] == password and username in dictionary: