**Professional Readiness For Innovative Employability And Entrepreneurship X8001**

**Project Name: Personal Expense Tracker Application**

Team Id: PNT2022TMID27972

Team Leader: P. Gurunathan

Team Member 1: D. Kishor

Team Member 2: C.S. Srinanthan

Team Member 3: S. Sudheekshan Kumar

**CONTENT**

1. [INTRODUCTION 3](#_TOC_250031)
   1. [Project Overview 3](#_TOC_250030)
   2. [Purpose 4](#_TOC_250029)
2. [LITERATURE SURVEY 5](#_TOC_250028)
   1. [Existing problem 6](#_TOC_250027)
   2. [References 11](#_TOC_250026)
   3. Problem Statement Definition 12
3. [IDEATION & PROPOSED SOLUTION 13](#_TOC_250025)
   1. Empathy Map Canvas 13
   2. [Ideation & Brainstorming 14](#_TOC_250024)
   3. [Proposed Solution 16](#_TOC_250023)
   4. [Problem Solution fit 17](#_TOC_250022)
4. [REQUIREMENT ANALYSIS 18](#_TOC_250021)
   1. Functional requirement 18
   2. [Non-Functional requirements 18](#_TOC_250020)
5. [PROJECT DESIGN 19](#_TOC_250019)
   1. [Data Flow Diagrams 19](#_TOC_250018)
   2. [Solution & Technical Architecture 20](#_TOC_250017)
   3. [User Stories 21](#_TOC_250016)
6. [PROJECT PLANNING & SCHEDULING 22](#_TOC_250015)
   1. [Sprint Planning & Estimation 22](#_TOC_250014)
   2. [Sprint Delivery Schedule 22](#_TOC_250013)
   3. [Reports from JIRA 23](#_TOC_250012)
7. [CODING & SOLUTIONING 24](#_TOC_250011)
   1. Feature 1 24
   2. Feature 2 26
   3. [Database Schema 28](#_TOC_250010)
8. [TESTING 29](#_TOC_250009)
   1. [Test Cases 29](#_TOC_250008)
   2. User Acceptance Testing 29
9. [RESULTS 31](#_TOC_250007)
   1. [Performance Metrics 31](#_TOC_250006)
10. [ADVANTAGES & DISADVANTAGES 35](#_TOC_250005)
11. [CONCLUSION 37](#_TOC_250004)
12. [FUTURE SCOPE 37](#_TOC_250003)
13. [APPENDIX 37](#_TOC_250002)

[Source Code 37](#_TOC_250001)

[GitHub & Project Demo Link 64](#_TOC_250000)

**1.INTRODUCTION**

Modern life offers a plethora of options of services and goods for consumers. As a result, people’s expenses have gone up dramatically, e.g., compared to a decade ago, and the cost of living has been increasing day by day. Thus it becomes essential to keep a check on expenses in order to live a good life with a proper budget set up. The Expense Tracker Mobile Application was developed for Mobile Phone users as well as desktop to 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. Optional data such as sub-category and extra notes about the expense can be entered as well. The application allows users to track their expenses daily, weekly, monthly, and yearly in terms of summary, bar graphs, and pie-charts. This mobile application 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. An analysis comparing existing expense tracking software with the one being introduced is provided

* 1. **PROJECT OVERVIEW:**

**Personal Expense Tracker Application:**

Personal Expense Tracker is an application which tracks the user’s expenses manually and track the whole expenses throughout the month and year as well as daily basis. In this application the user first create an account with the credentials include username, Email, password, Confirm Password and with initial amount or monthly income.

After that the user can able to login into their respective page with a valid credentials. In that page, the user can see the dashboard in the dashboard the is some features such as home, Add expenses, Budget plan, Expense log and log out option are present.

**PROJECT WORK FLOW AFTER SUCESSFUL LOGIN**

* The user can see the initial amount, Balance and the Expense amount till the date.
* User can add the expenses in the add expenses page in that user can fill date, category and description of the expense
* The user can add the budget pan to their expenses
* After that the user can logout from their respective page
  1. **PURPOSE**

This project will help user to manage their personal expenses in a modern and efficient manner. An expense tracking app is an exclusive suite of services for people who seek to handle their earnings and plan their expenses and savings efficiently. It will help to maintain and manage their expenses through mobile phones. This 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 need.

Some specific purposes are:

* It helps you stick to your Budget
* Tracking Your Expenses Can Reveal Spending Issues.
* It Helps You Meet Your Financial Objectives.
* Make It Easier with an App or Software.
* Work Together as a Couple.

1. **LITERATURE SURVEY**
   1. **EXISTING PROBLEM**

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

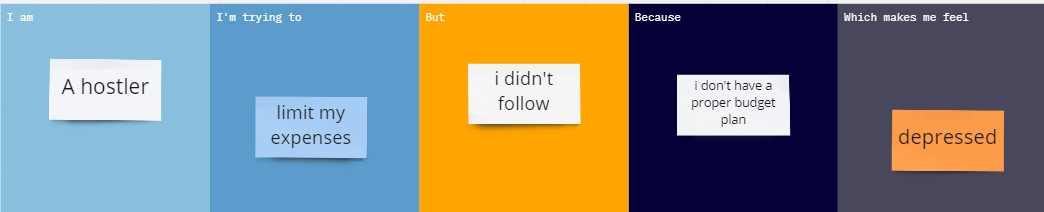
* 1. **REFERENSES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **Author** | **Year** | **Title of the paper** | **Contribution** |
| 1 | [*Ting-Sheng*](https://ieeexplore.ieee.org/author/37087367097)  [*Weng*](https://ieeexplore.ieee.org/author/37087367097)*;* [*Shin-Fa Tseng*](https://ieeexplore.ieee.org/author/37088101674) | 2010 | Design of a Personal Financial Planning Management Information System | This system provides financial planning functions, including accounting, budgeting, financial planning,  and monitoring |
| 2 | [Wangbin,](https://ieeexplore.ieee.org/author/37085572448) [Xieqi,](https://ieeexplore.ieee.org/author/37085566911) [Shihuaxin](https://ieeexplore.ieee.org/author/37085568418), [Caoxinyu,](https://ieeexplore.ieee.org/author/37085570348) [Wangwenjing,](https://ieeexplore.ieee.org/author/37085575239) [Chendi](https://ieeexplore.ieee.org/author/37085582403) | 2015 | The design and implementation of inpatient medical expenses analysis system | They are the module of querying inpatients financial information and the analysis for cost-effectiveness of single disease. These modules help in reducing the hospital cost spent by the patients by  analysing total hospital costs |
| 3 | Sumit Yadav, Richa Malhotra,Jyoti Tripathi | 2016 | Smart Expense Management Model for Smart Homes | This model keeps track of all type of expenses incurred by the house hold owner. The integration of smart home technology with the expense tracker increases the  efficiency |
| 4 | [Shahed Anzarus](https://ieeexplore.ieee.org/author/37086067258) [Sabab](https://ieeexplore.ieee.org/author/37086067258); [Sadman](https://ieeexplore.ieee.org/author/37086607842) [Saumik Islam](https://ieeexplore.ieee.org/author/37086607842); [Md.](https://ieeexplore.ieee.org/author/37086457753) [Jewel Rana](https://ieeexplore.ieee.org/author/37086457753); [Monir](https://ieeexplore.ieee.org/author/37086613694) [Hossain](https://ieeexplore.ieee.org/author/37086613694) | 2018 | A Smart Approach to Track Everyday Expense | It is a mobile application that uses the scanned images of the bills to get the data. It scans the textual information to get data and also monitor the messages received regarding  income |
| 5 | *J.L.Yeo, P.S.*  *JosephNg,*  *K.A. Alezabi, H.C. Eaw,*  *K.Y. Phan* | 2020 | Time Scheduling and Finance Management: University Student Survival Kit | The time scheduling in the application helps the students to perform better in the academics.Money management skills in the application helps the user to log their weekly income and  expenses |
| 6 | DenisE.Yurochkin;Anton  AHoroshiy; Saveliy A.  Karpukhin | 2021 | Development of an Application for Expense Accounting | The obtained data is stored in a database which can infer the expenses made and can beused to make a budget plan. This idea helps in finding out the sectors in which the expenses are made at the most and at the least providing an  understanding of the expenses |

* 1. **PROBLEM STATEMENT DEFINITIONS**

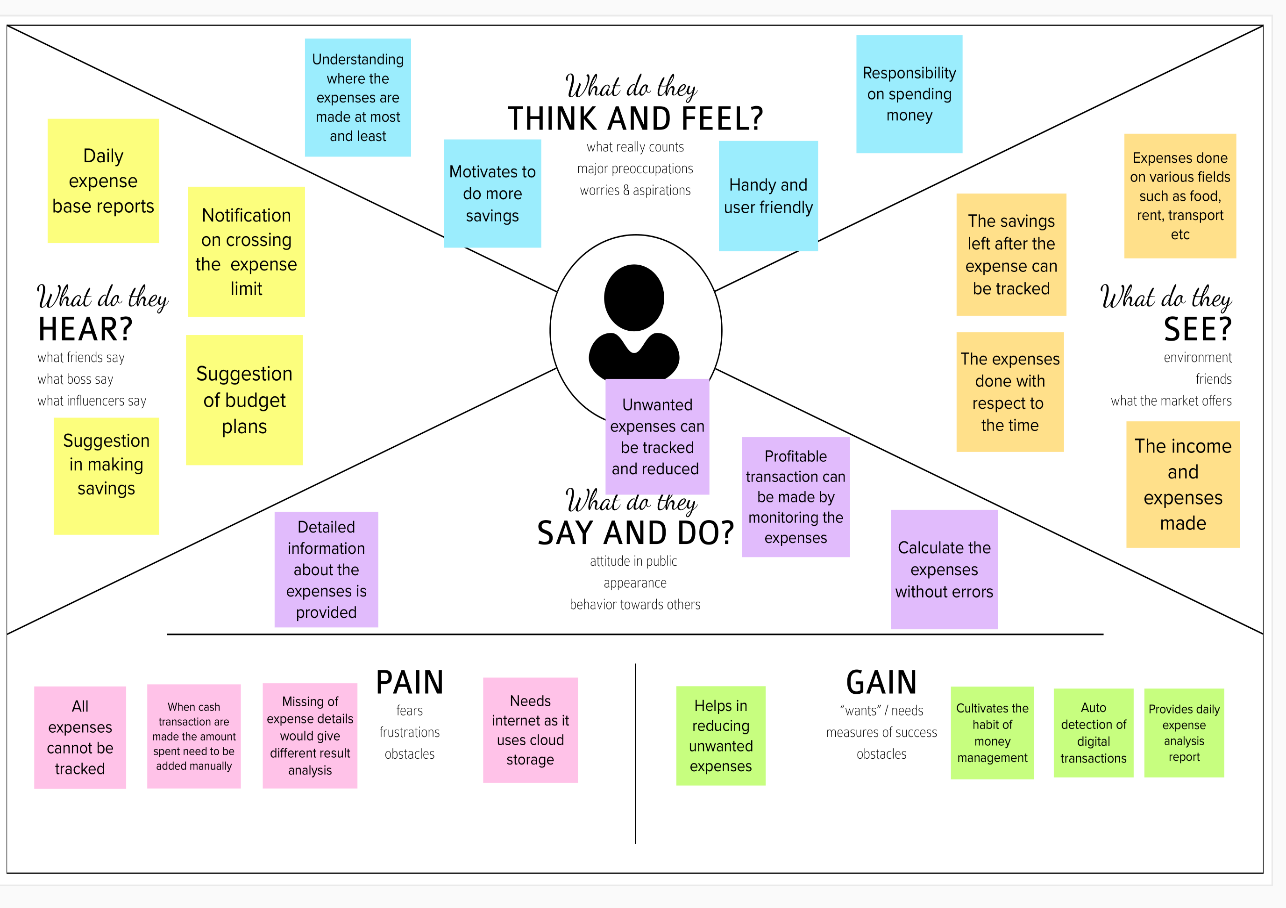
The day-to-day expense has become mostly digitalized in today’s world. When the transactions are made directly from the bank account it becomes difficult to keep track of the expenses. The expenses are made until there is no money left to make transactions. Thus, it becomes necessary to keep track of the daily expenses to reduce unwanted transactions and avoid bankrupt. Analyzing the expenses regularly helps for better plannings. Better budget plans can be carried out based on the expense analysis reports and better savings can be made.

|  |  |
| --- | --- |
| Who does the problem affect? | People making lots of transactions. |
| What is the issue? | The traditional expense trackers were bill books and tally books which needed manual entries of the expenses made. Thus, some entries may get missed out resulting in wrong expense analysis. It also includes manual  calculation which is subjective to |
|  | errors. Thus, it needed to be monitored closely and attentively. |
| When does the issue occurs? | These errors commonly occurs when there is a lot of transactions made. The more the transactions the more will be the expenses which are needed to be accounted. Also, one of the major things to consider is, these are physical data which might get lost, burnt or get  spoiled due to unexpected events |
| Where is the issue occurring? | These issues occurs when the person is unable to track their expenses regularly. |
| Why is it important that we fix the problem? | By solving this issue people can have more savings, better lifestyle, avoid unnecessary expenses and make better  plans before spending money. |





1. **IDEATION AND PROPOSED SOLUTION**
   1. **EMPATHY MAP AND CANVA**



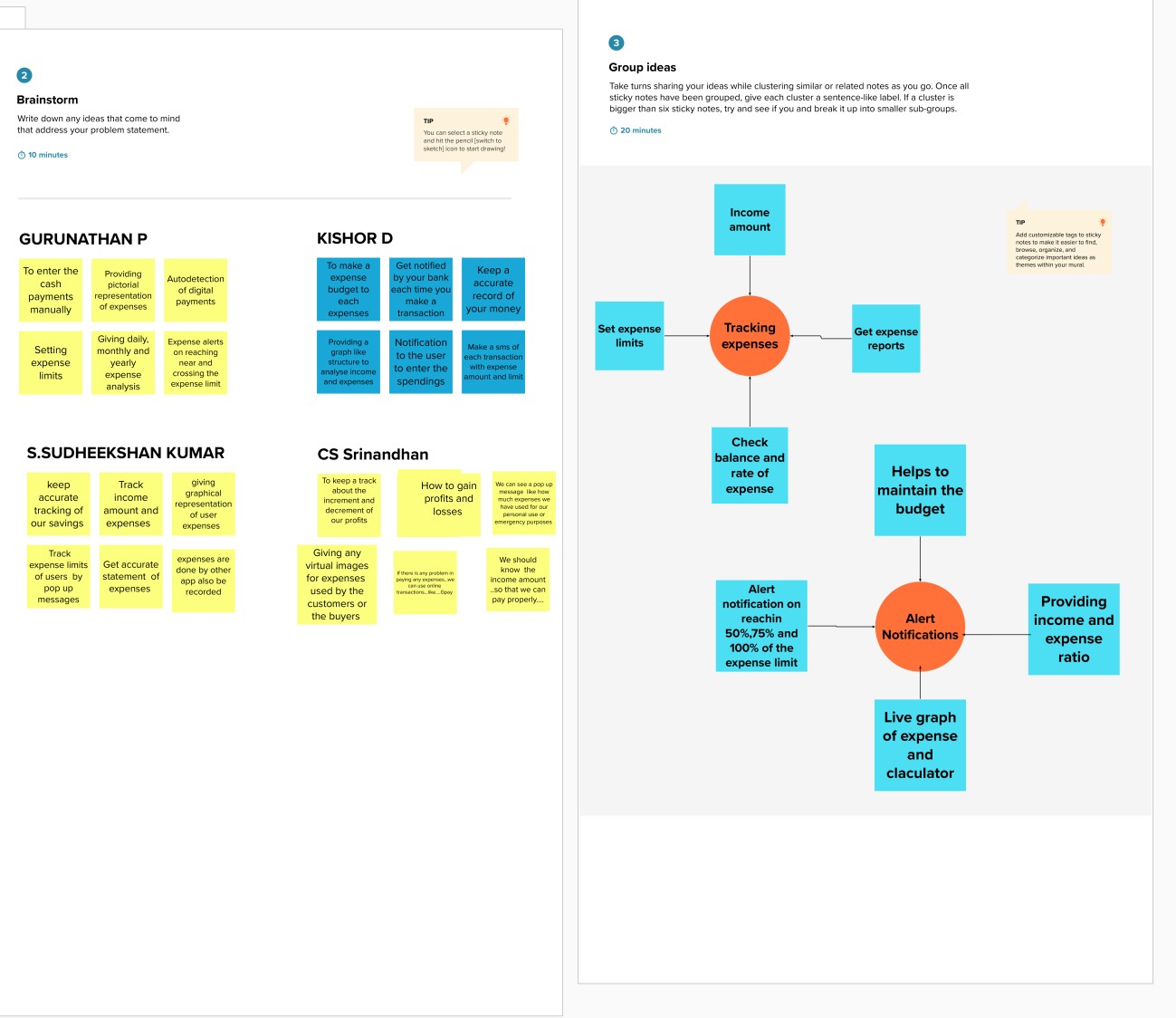
* 1. **IDEATION AND BRAINSTORMING**

Brainstorm & Idea Prioritization:

Step 1: Team Gathering, Collaboration and select the Problem Statement



Step 2: Brainstorm, idea Listing and Grouping

****

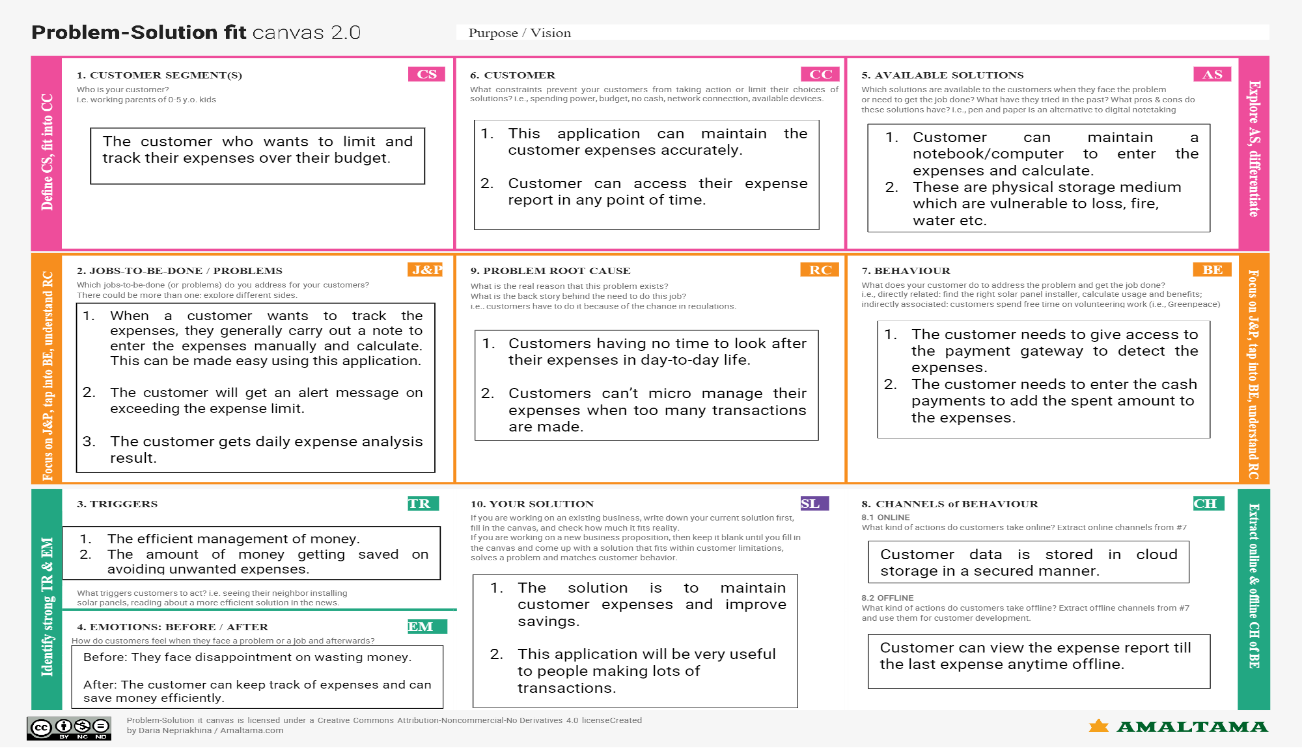
Step 3: Idea Prioritization

****

* 1. **PROPOSED SOLUTION**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | * Reducing unwanted expenses * Ignorance about the savings left * Not being able to make limited expense |
| 2. | Idea / Solution description | * Maintains record of the expenses * Analyses the amount of expenses made on different sectors * Helps in setting up an expense limit and avoids unwanted expense |
| 3. | Novelty / Uniqueness | * The data is stored in cloud for easy retrieval * The data is safe from physical hazards * Provides graphical representation of the expenses which is easy to understand |
| 4. | Social Impact / Customer Satisfaction | * User friendly, simple and easy to use * Handy and accessible anytime * Expenses and savings can be monitored 24/7 |
| 5. | Business Model (Revenue Model) | * Releasing in Playstore * Paid subscriptions for advanced budgeting plans with huge transactions * Advertising the use of the application |
| 6. | Scalability of the Solution | The project will take 4 to 6 months to get completed and can store data depending upon  the cloud storage that has been bought. |

* 1. **PROBLEM SOLUTION FIT**



1. **REQUIREMENT ANALYSIS**
   1. **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 Mobile number |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 | User Authentication | Authenticate via password or OTP. |
| FR-4 | Permission To Notify | Permission via Messages. |
| FR-5 | Handles Documents | Handle via PDF. |
| FR-6 | Tracks Receipts | Track via Mobile Phones. |

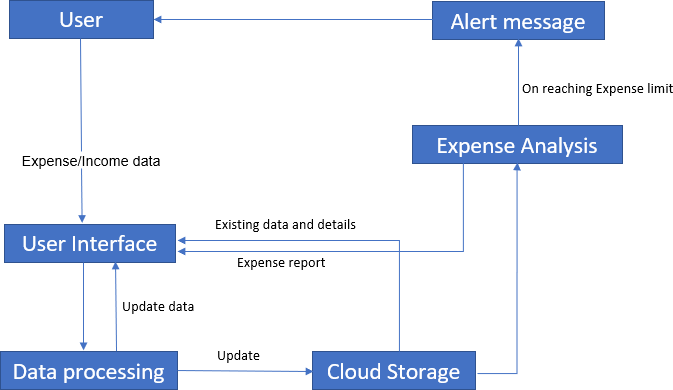
* 1. **NON FUNCTIONAL REQUIREMENTS:**

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

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Track the Expenses. |
| NFR-2 | **Security** | Confidential Password. |
| NFR-3 | **Reliability** | A mobile device to access the application anytime  and anywhere. |
| NFR-4 | **Performance** | Stable network connection. |
| NFR-5 | **Availability** | Good internet connection. |
| NFR-6 | **Scalability** | The expense limits are scalable based on the user  needs. |

1. **PROJECT DESIGN**
   1. **DATAFLOW DIAGRAMS:**

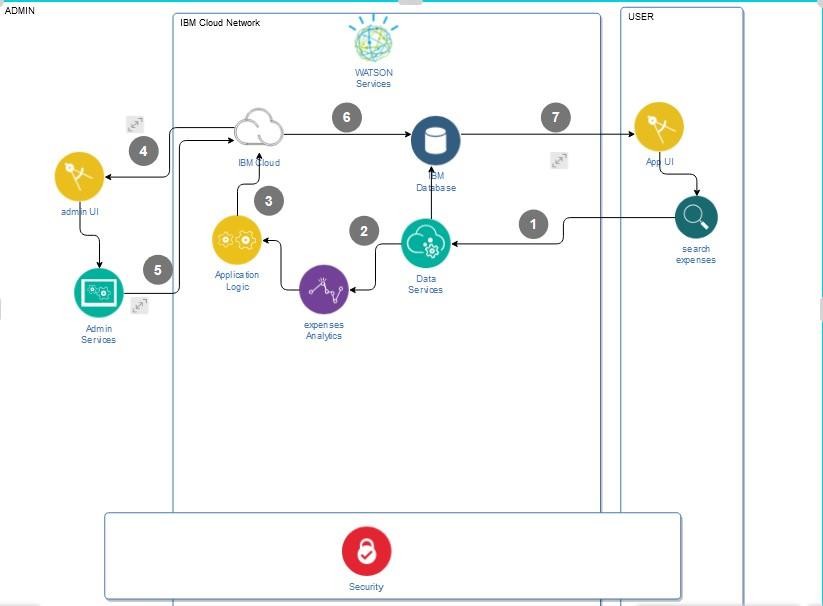
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



* 1. **SOLUTION AND TECHNICAL ARCHITECTURE:**

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

Example: Order processing during pandemics for offline mode

Reference: <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>

Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API’s etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table-1: Components & Technologies:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | How user interacts with application e.g. Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript. |
| 2. | Application Logic-1 | Logic for a process in the application | Python |
| 3. | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
| 4. | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
| 5. | Database | Data Type, Configurations etc. | IBM cloud |
| 6. | Cloud Database | Database Service on Cloud | IBM DB2. |
| 7. | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
| 8. | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
| 9. | External API-2 | Purpose of External API used in the application | Aadhar API, etc. |
| 10. | Machine Learning Model | Purpose of Machine Learning Model | Object Recognition Model, etc. |
| 11. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud Local Server Configuration:  Cloud Server Configuration : | Local, Cloud Foundry, Kubernetes, etc. |

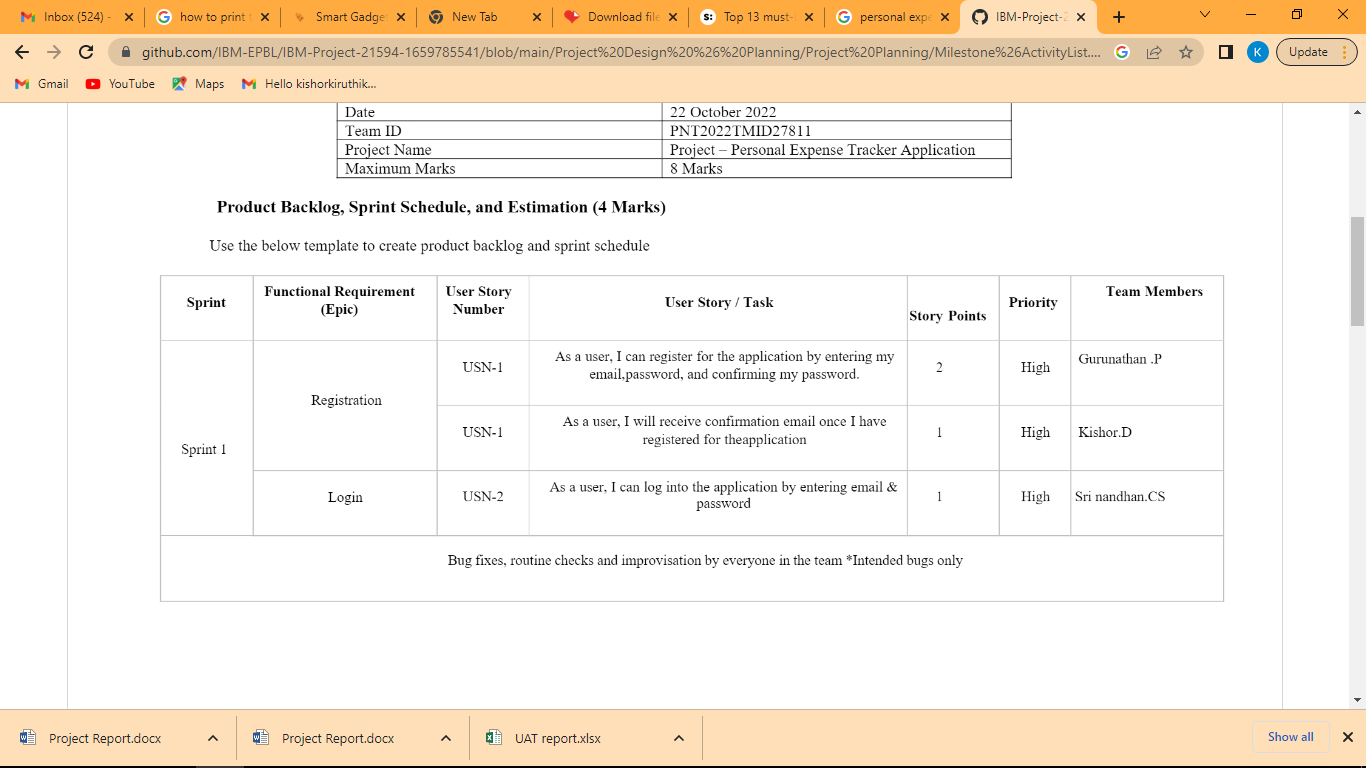
Table-2: Application Characteristics:

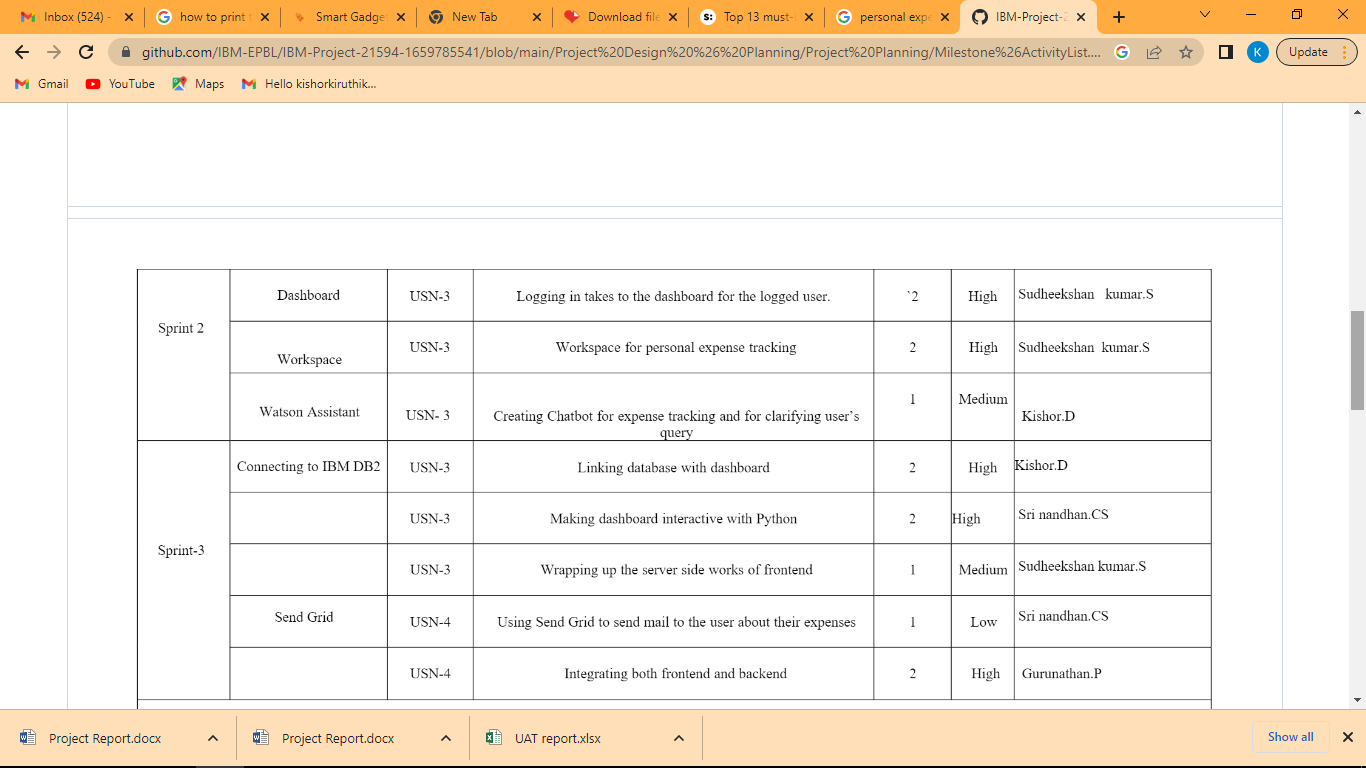
|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | List the open-source frameworks used | Technology of Opensource framework |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Technology used |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Technology used |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of  Cache, use of CDN’s) etc. | Technology used |

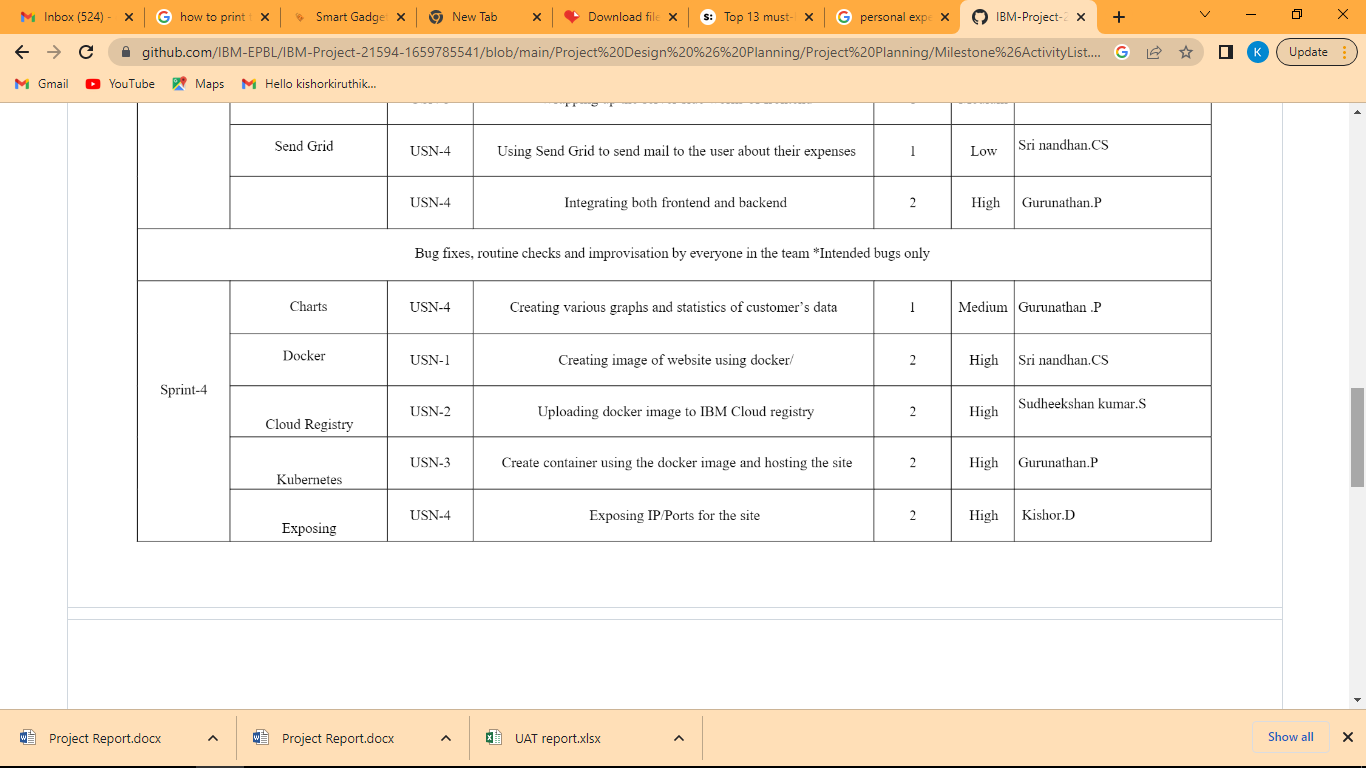
* 1. **USER STORIES:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register for the application by  entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
|  |  | USN-3 | As a user, I can register for the application through 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 | Valid Gmail id and password to sign in | Medium | Sprint-1 |
|  | Login | USN-5 | As a user, I can log into the application by entering email & password | Entering registered mail id and password to login | High | Sprint-1 |
|  | Dashboard | USN-6 | Displays the current expense and savings |  | High | Sprint-1 |
| Customer (Web user) | Dashboard | USN-7 | Displays the current expense and savings |  | High | Sprint-1 |
| Administrator | Add/ Remove account | USN-8 | The administrator can add or remove an account | Using valid credentials to do the task | Low | Sprint-1 |

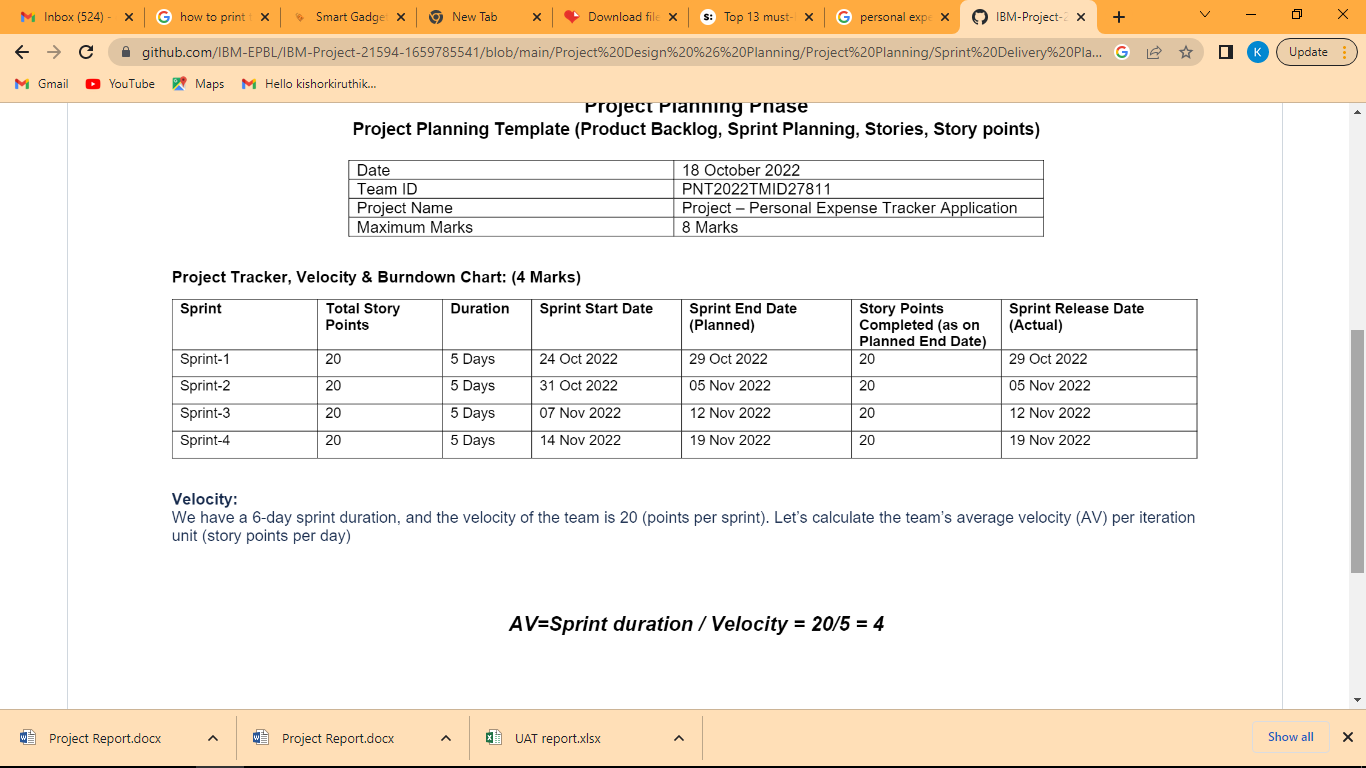
1. **PROJECT PLANNING AND SCHEDULING**
   1. **SPRINT PLANNING AND ESTIMATION:**



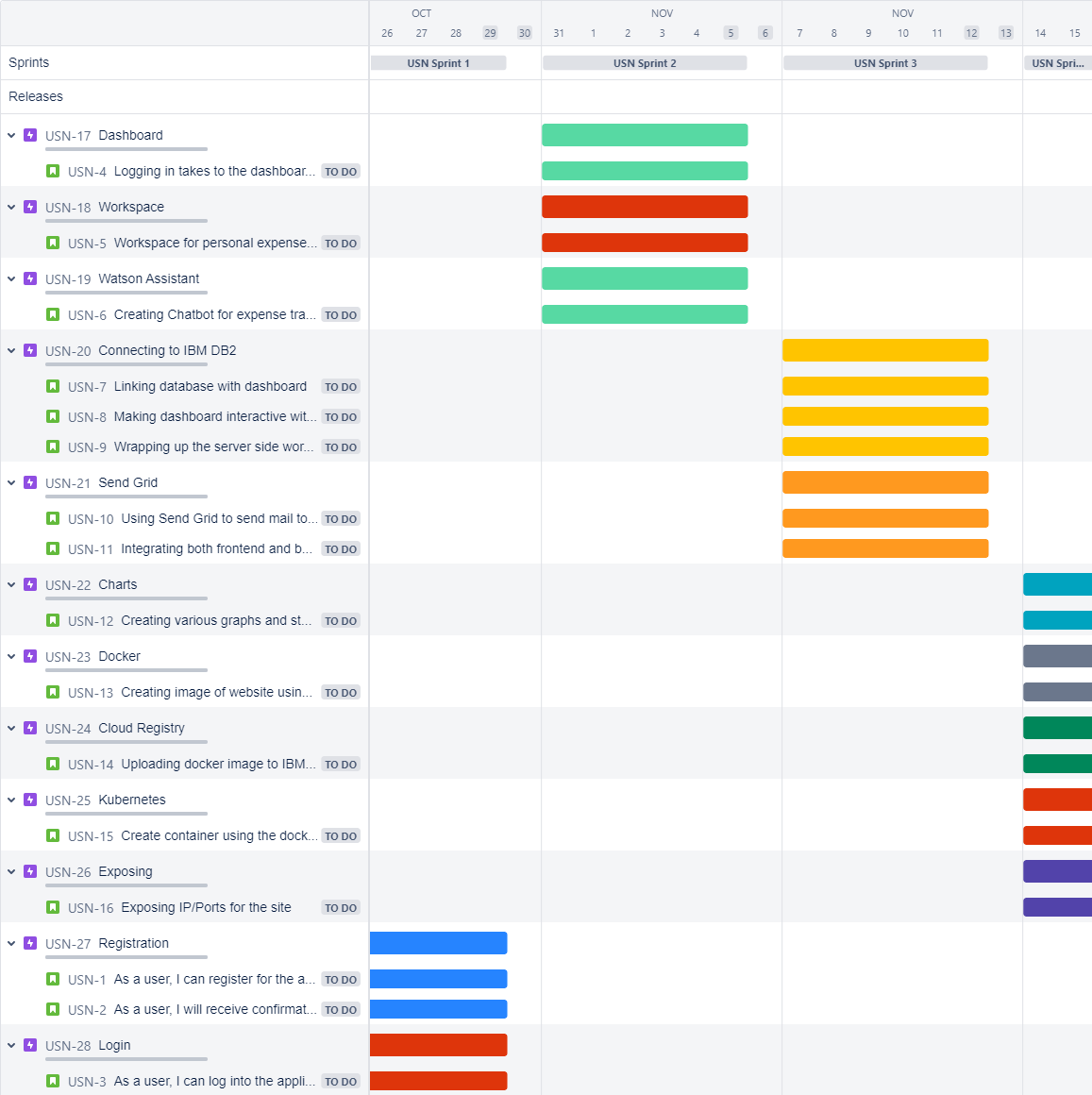




* 1. **SPRINT DELIVERY SCHEDULE**



* 1. **REPORTS FROM JIRA**

****

1. **CODDING AND SOLUTIONING**
   1. **FEATURE 1**

**Cloud Feature**

### Self-service On-Demand

This is one of the most essential and significant characteristics of cloud computing. This means that cloud computing enables clients to regularly monitor the abilities, allotted network storage, and server uptime. Therefore, it is one of the most fundamental features of cloud computing that helps clients control various computing abilities as per their requirements.

### Resources Pooling

This is also a fundamental characteristic of cloud computing. Pooling resources means that a cloud service provider can distribute resources for more than one client and provide them with different services according to their needs. Resource Pooling is a multi-client plan useful for data storing, bandwidth services and data processing services. The provider administers the data stored in real-time without conflicting with the client’s need for data.

### Easy Maintenance

This is one of the best cloud characteristics. Cloud servers are easy to maintain with low to almost zero downtime. Cloud Computing powered resources undergo several updates frequently to optimise their capabilities and potential. The updates are more viable with the devices and perform quicker than the previous versions.

### Economical

This kind of service is economical as it efficiently reduces IT costs and data storage expenditure. Moreover, most cloud computing services are free. Even if there are paid plans, it’s only to expand storage capacity, and these costs are often very nominal. This is a massive advantage of using cloud computing services.

### Rapid Elasticity and Scalability

The best part of using cloud storage is that it can easily handle all the workload and data load concerning storage. Furthermore, as it is fully automated, businesses and organisations can save heavily on manual labour and technical staffing

* 1. **FEATURE 2**

**Docker** **Features**:

### Faster and Easier configuration:

It is one of the key features of Docker that helps you in configuring the system in a faster and easier manner. Due to this feature, codes can be deployed in less time and with fewer efforts. The infrastructure is not linked with the environment of the application as Docker is used with a wide variety of environments.

### Application isolation:

Docker provides containers that are used to run applications in an isolated environment. Since each container is independent, Docker can execute any kind of application.

### Increase in productivity:

It helps in increasing productivity by easing up the technical configuration and rapidly deploying applications. Moreover, it not only provides an isolated environment to execute applications, but it reduces the resources as well.

### Swarm:

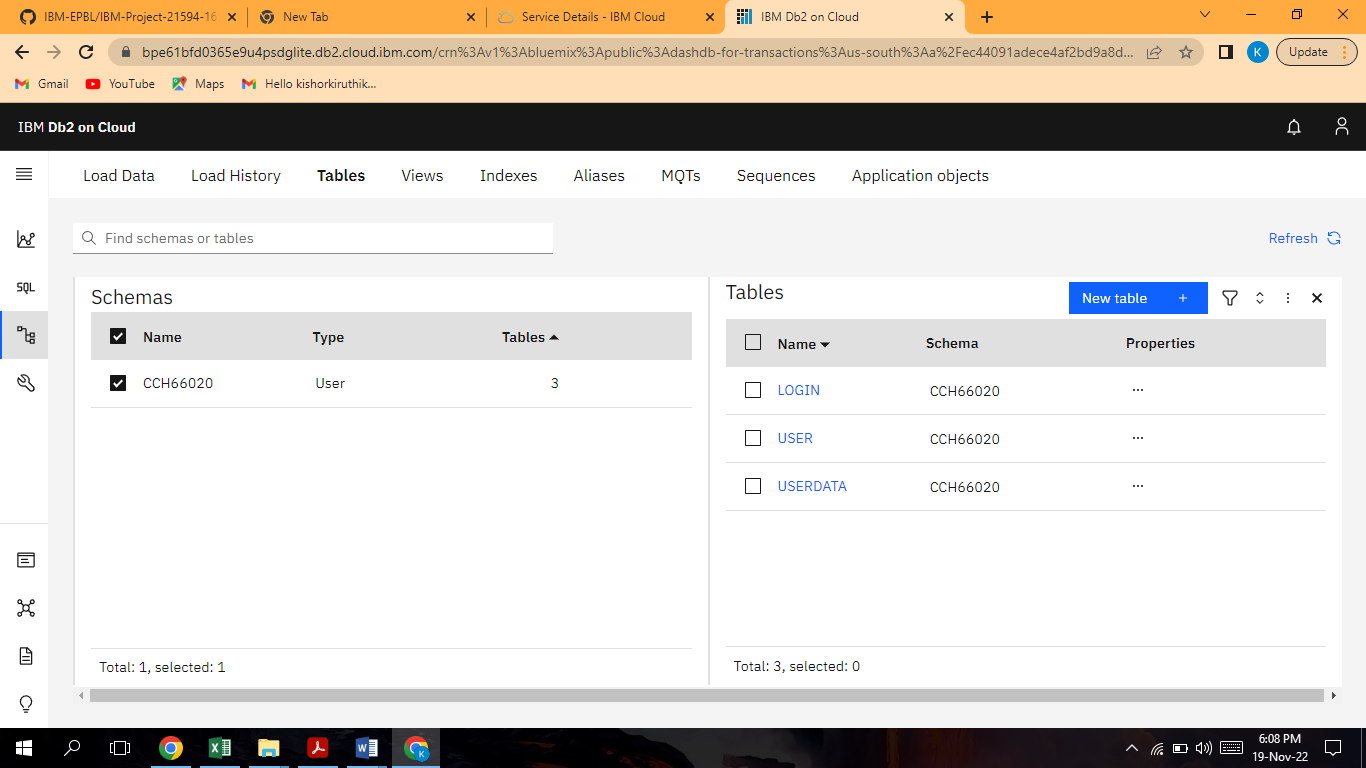
Swarm is a clustering and scheduling tool for Docker containers. At the front end, it uses the Docker API, which helps us to use various tools to control it. It is a self-organizing group of engines that enables pluggable backends.

### Services:

Services is a list of tasks that specifies the state of a container inside a cluster. Each task in the Services lists one instance of a container that should be running, while Swarm schedules them across the nodes.

## **Kubernetes Features:**

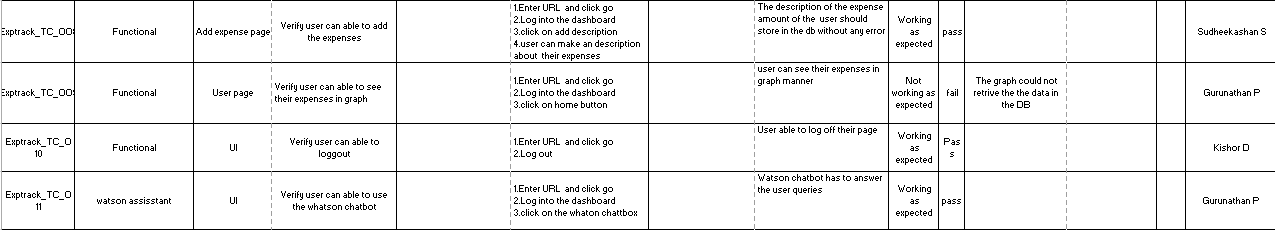
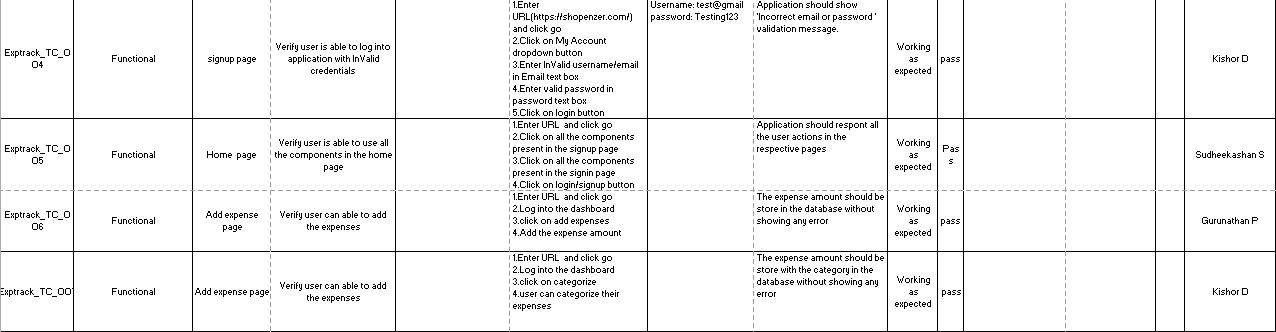
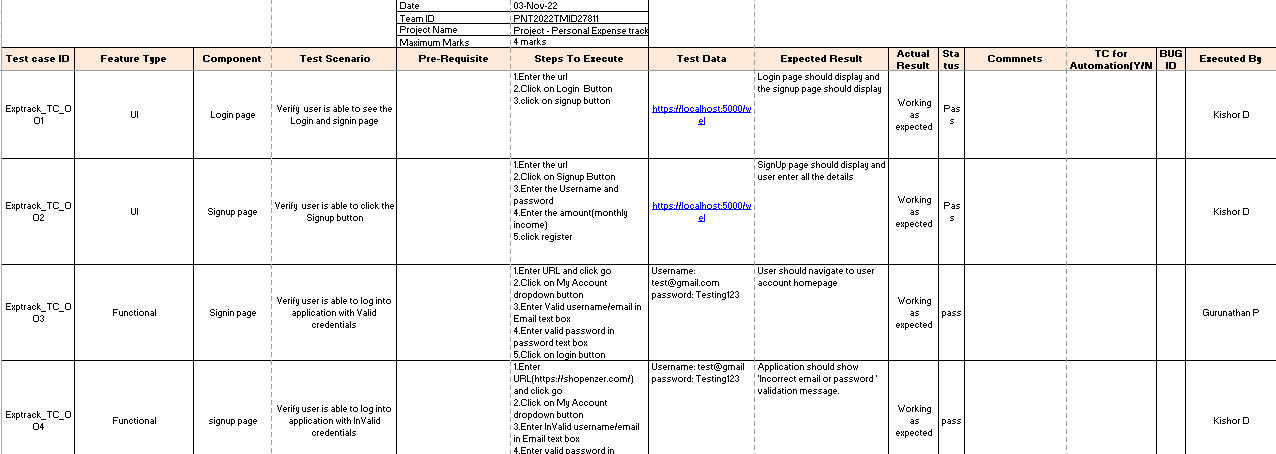
* 1. **Auto-scaling**. Automatically scale containerized applications and their resources up or down based on usage
  2. **Lifecycle management**. Automate deployments and updates with the ability to:
     1. Rollback to previous versions
     2. Pause and continue a deployment
  3. **Declarative model**. Declare the desired state, and K8s works in the background to maintain that state and recover from any failures.
  4. **Resilience and self-healing**. Auto placement, auto restart, auto replication and auto scaling provide application self-healing
  5. **Persistent storage**. Ability to mount and add storage dynamically
  6. **Load balancing**. Kubernetes supports a variety of internal and external load balancing options to address diverse needs.
  7. **DevSecOps support**. [DevSecOps](https://tanzu.vmware.com/devsecops) is an advanced approach to security that simplifies and automates container operations across clouds, integrates security throughout the container lifecycle, and enables teams to deliver secure, high-quality software more quickly. Combining DevSecOps practices and Kubernetes improves developer productivity.
  8. **DATABASE SCHEMA:**



1. **TESTING**
   1. **TEST CASES**

|  |  |
| --- | --- |
|  | **Test Scenarios** |
| 1 | Verify user is able to see login page |
| 2 | Verify user is able to loginto application or not? |
| 3 | Verify user is able to navigate to create your account page? |
| 4 | Veriify login page elements |
|  |  |
|  | **Sign uppage** |
| 1 | verify user can see the register page |
| 2 | verify user can enter the details |
| 3 | verify user can enter the amount |
| 4 | verify user can come back to the login form |
|  |  |
|  | **User page** |
| 1 | Verify user can see and open the home page in the dashboard |
| 2 | Verify user can see and open the add expense in the dashboard |
| 3 | Verify user can see and open the Expense Log in the dashboard |
|  |  |
|  |  |
|  | **HomePage** |
| 1 | user can verify their expenses and balance |
| 2 | User can verify their expenses in graph |
|  |  |
|  | **Add expenses** |
| 1 | User can enter the expense amount |
| 2 | User can select the category |
| 3 | User can mention their description about the expenses |
|  |  |
|  |  |
|  | **Expense log** |
| 1 | User can verify the entered datails |

* 1. **USER ACCEPTING TESTING**



**UAT Execution & Report**

# **Purpose Document**

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

# **Defect Analysis**

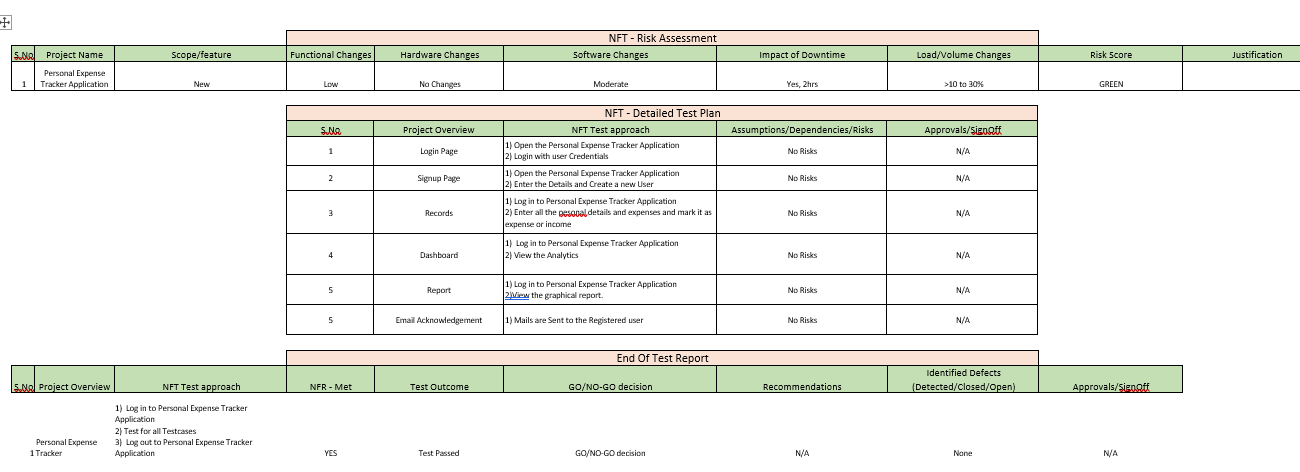
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| By Design | 6 | 3 | 1 | 2 | 12 |
| Duplicate | 1 | 1 | 0 | 0 | 2 |
| External | 2 | 1 | 0 | 0 | 3 |
| Fixed | 4 | 0 | 1 | 0 | 5 |
| Not Reproduced | 0 | 2 | 0 | 0 | 2 |
| Skipped | 1 | 1 | 1 | 1 | 3 |
| Won't Fix | 0 | 3 | 0 | 2 | 5 |
| Totals | 14 | 11 | 3 | 5 | 33 |

# **3 Test Case Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Total Cases** | **Not Tested** | **Fail** | **Pass** |
| Print Engine | 5 | 0 | 0 | 5 |
| Client Application | 35 | 0 | 0 | 35 |
| Security | 8 | 0 | 0 | 8 |
| Outsource Shipping | 10 | 0 | 0 | 10 |
| Exception Reporting | 3 | 0 | 0 | 3 |
| Final Report Output | 15 | 0 | 0 | 15 |
| Version Control | 4 | 0 | 0 | 4 |

1. **PERFORMANCE TESTING**
   1. **PERFORMANCE METRICES**

****

1. **ADVANTAGES AND DISADVANTAGES**
   1. **ADVANTAGES**

The best organizations have a way of tracking and handling these reimbursements. This ideal practice guarantees that the expenses tracked are accurately and in a timely manner. From a company perspective, timely settlements of these expenses when tracked well will certainly boost employees' morale Financially Aware and Improve Money Management tracking your expenditures ensures you achieve your project financial targets. How is that? By clearly understanding your project spending using project budget limits, you can aptly make the necessary changes to complete your project within time and budget. Effective expense tracking and reporting to avoid conflict. As a project manager or business owner, you can set clear policies for the expense types and reimbursement limits to avoid misunderstandings are about costs. Tracking the project expenses by asking team members to provide receipts is helpful to avoid conflict and maintain compliance also. An excellent reporting mechanism is extremely helpful to support the amount to be reimbursed to your team and also invoicing to your customer.

* 1. **DISADVANTAGES**

If a person first makes a budget plan, then places money in savings before spending any each new pay period or month, the tracking goal can help. In this way, tracking spending and making sure all receipts are accounted for only needs to be done once or twice a month. Even with constant tracking of one's spending habits, there is no guarantee that financial goals will be met.

1. **CONCLUSION**

In conclusion, developing a personal budget and tracking all expenses and spending is a crucial aspect of personal finances. Set aside a fixed amount in a savings account, they say you should always have three months work of your living expenses in a savings account in case of emergencies. Lastly, educating the children early in life about personal finances should be a mandatory class in every school. Parents need to proactive in teaching their children about banking, credit card, interest rates, and credit. The importance of actually seeing my spending on my budget sheet was enlightening. However, I know now where I can trim the fat, and by changing just a few items and cutting back on others, I will see a substantial increase in money that I can put into savings.

1. **FUTURE SCOPE**

In future in order to do the things in an efficient manner so that in this application has to be made of high features such as direct bank account linking, representation of expenses in the graphical or pie chart manner and the receipt could be print for each user.

1. **APPENDIX**

**SOURCE CODE:**

**LOGIN PAGE:**

<html lang="en" dir="ltr">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- <title>Login Form | CodingLab</title> -->

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.2/css/all.min.css"/>

    <title>Perssonal Expense Tracker</title>

    <style>

       @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');

\*{

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: 'Poppins',sans-serif;

}

body{

  background-image: url('coins.jpg');

  background-size: cover;

  background-repeat: no-repeat;

  background-attachment: fixed;

  object-fit: cover;

  overflow: hidden;

}

::selection{

  background: rgba(26,188,156,0.3);

}

.container{

  max-width: 440px;

  padding: 0 20px;

  margin: 100px auto;

}

.wrapper{

  width: 100%;

  background: #fff;

  border-radius: 5px;

  box-shadow: 0px 4px 10px 1px rgba(0,0,0,0.1);

}

.wrapper .title{

  height: 90px;

  border-radius: 5px 5px 0 0;

  text-align: center;

  position: relative;

  background: #838d2f8a;

  font-size: 25px;

  font-weight: 550;

  display: flex;

  align-items: center;

  justify-content: center;

}

.wrapper form{

  padding: 30px 25px 25px 25px;

}

.wrapper form .row{

  height: 45px;

  margin-bottom: 15px;

  position: relative;

}

.wrapper form .row input{

  height: 100%;

  width: 100%;

  outline: none;

  padding-left: 60px;

  border-radius: 5px;

  border: 1px solid lightgrey;

  font-size: 16px;

  transition: all 0.3s ease;

}

form .row input:focus{

  border-color: #16a085;

  box-shadow: inset 0px 0px 2px 2px rgba(26,188,156,0.25);

}

form .row input::placeholder{

  color: #999;

}

.wrapper form .row i{

  position: absolute;

  width: 47px;

  height: 100%;

  color: #fff;

  font-size: 18px;

  background: #838d2f8a;

  border: 1px solid #16a085;

  border-radius: 5px 0 0 5px;

  display: flex;

  align-items: center;

  justify-content: center;

}

.wrapper form .pass{

  margin: -8px 0 20px 0;

}

.wrapper form .pass a{

  color: #16a085;

  font-size: 17px;

  text-decoration: none;

}

.wrapper form .pass a:hover{

  text-decoration: underline;

}

.wrapper form .button input{

  color: rgb(0, 0, 0);

  font-size: 20px;

  font-weight: 500;

  padding-left: 0px;

  background: #838d2f8a;

  border: 1px solid #16a085;

  cursor: pointer;

}

form .button input:hover{

  background: #437cd9;

}

.wrapper form .signup-link{

  text-align: center;

  margin-top: 20px;

  font-size: 17px;

}

.wrapper form .signup-link a{

  color: #16a085;

  text-decoration: none;

}

form .signup-link a:hover{

  text-decoration: underline;

}

    </style>

  </head>

  <body>

    <div class="container">

      <h2 style="font-size: 30px; font-weight: 600;"><center>Personal Expense Tracker</center><h2>

        <br>

      <div class="wrapper">

        <div class="title"><span>Login Form</span></div>

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

          <div class="row">

            <i class="fas fa-user"></i>

            <input name="email" type="text" placeholder="Email" required>

          </div>

          <div class="row">

            <i class="fas fa-lock"></i>

            <input name="pass" type="password" placeholder="Password" required>

          </div>

          <div class="row button">

            <input type="submit" value="Login">

          </div>

          <div class="signup-link">Not a member? <a href="{{url\_for('registerpage')}}">Signup now</a></div>

        </form>

      </div>

    </div>

  </body>

</html>

**HOME PAGE:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <title>Personal Expense Tracker</title>

  <link rel="stylesheet" href="static/styles.css" />

  <!-- Font Awesome Cdn Link -->

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css"/>

</head>

<body>

  <h1 style="text-align: center;   background: #838d2f8a; padding: 7px;">Personal Expense tracker</h1>

  <div class="container">

    <nav>

        <ul>

          <li><a href="#">

            <i class="fas fa-bars" style="color: black;"></i>

            <span class="nav-item" Style="color:black; font-size:20px; font-weight:600" >Dashboard</span>

          </a></li>

        <li><a class="active">

          <i class="fas fa-home"></i>

          <span class="nav-item">Home</span>

        </a></li>

        <li><a href="{{url\_for('expense')}}">

          <i class="fas fa-wallet"></i>

          <span class="nav-item">Add Expense</span>

        </a></li>

        <li><a href="{{url\_for('report')}}">

          <i class="fas fa-receipt"></i>

            <span class="nav-item">Expense log</span>

          </a></li>

        <li><a href="{{url\_for('wel')}}">

          <i class="fas fa-sign-out-alt"></i>

          <span class="nav-item">Log out</span>

        </a></li>

      </ul>

    </nav>

    <section class="main">

      <div class="main-top">

        <h1>Your Expenses</h1>

        <div class="homecard">

            <div>

             <h4 style="text-align: center;">INITIAL AMOUNT</h4 style="text-align: center;">

              <h1>1000.0</h1>

            </div>

            <div>

              <h4 style="text-align: center;">EXPENSE MADE</h4 style="text-align: center;">

                <h1>103.0</h1>

             </div>

            <div>

              <h4 style="text-align: center;">BALANCE</h4 style="text-align: center;">

                <h1>897.0</h1>

             </div>

        </div>

      </div>

  </div>

</body>

<script>

  window.watsonAssistantChatOptions = {

    integrationID: "7f157622-db65-4138-8cca-60e657ef7461", // The ID of this integration.

    region: "us-south", // The region your integration is hosted in.

    serviceInstanceID: "5ee64e34-e58d-45b6-8d1b-e739162fb7cf", // The ID of your service instance.

    onLoad: function(instance) { instance.render(); }

  };

  setTimeout(function(){

    const t=document.createElement('script');

    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

    document.head.appendChild(t);

  });

</script>

</html>

**Expense.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <title>Personal Expense Tracker</title>

  <link rel="stylesheet" href="static/styles.css" />

  <!-- Font Awesome Cdn Link -->

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css"/>

</head>

<body>

  <h1 style="text-align: center;   background: #838d2f8a; padding: 7px;">Personal Expense tracker</h1>

  <div class="container">

    <nav>

        <ul>

          <li><a href="#">

            <i class="fas fa-bars" style="color: black;"></i>

            <span class="nav-item" Style="color:black; font-size:20px; font-weight:600" >Dashboard</span>

          </a></li>

        <li><a href="{{url\_for('home')}}">

          <i class="fas fa-home"></i>

          <span class="nav-item">Home</span>

        </a></li>

        <li class="active"><a href="#">

          <i class="fas fa-wallet"></i>

          <span class="nav-item">Add Expense</span>

        </a></li>

        </a></li>

        <li><a href="{{url\_for('display')}}">

          <i class="fas fa-receipt"></i>

            <span class="nav-item">Expense log</span>

          </a></li>

        <li><a href="{{url\_for('wel')}}">

          <i class="fas fa-sign-out-alt"></i>

          <span class="nav-item">Log out</span>

        </a></li>

      </ul>

    </nav>

    <section class="main">

      <div class="main-top">

        <h1>Add Expense</h1>

        <div>

          <form action="/adding" method="post">

            <label for="amnt">Expense Amount</label>

            <input type="number" step="5" id="amnt" name="amount" placeholder="Expense amount" required>

            <label for="category">Category</label>

            <select id="category" name="category">

              <option value="food">Food</option>

              <option value="clothes">Clothes</option>

              <option value="rent">Rent</option>

              <option value="grocery">Grocery</option>

              <option value="Fuel">fuel</option>

              <option value="stationary">Stationary</option>

              <option value="other">Others</option>

            </select>

            <label for="note">Description</label>

            <input type="text" id="note" name="note" placeholder="Decribe your expense" maxlength="50">

            <input type="submit" value="Add expense">

          </form>

        </div>

      </div>

  </div>

</body>

<script>

  window.watsonAssistantChatOptions = {

    integrationID: "7f157622-db65-4138-8cca-60e657ef7461", // The ID of this integration.

    region: "us-south", // The region your integration is hosted in.

    serviceInstanceID: "5ee64e34-e58d-45b6-8d1b-e739162fb7cf", // The ID of your service instance.

    onLoad: function(instance) { instance.render(); }

  };

  setTimeout(function(){

    const t=document.createElement('script');

    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

    document.head.appendChild(t);

  });

</script>

</html>

**Registration.html**

<!DOCTYPE html>

<html lang="en" dir="ltr">

  <head>

    <meta charset="UTF-8">

    <!---<title> Responsive Registration Form | CodingLab </title>--->

     <meta name="viewport" content="width=device-width, initial-scale=1.0">

   </head>

   <style>

     @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');

\*{

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: 'Poppins',sans-serif;

}

body{

  height: 100vh;

  display: flex;

  justify-content: center;

  align-items: center;

  padding: 5px;

  background-image: url('coins.jpg');

  background-size: cover;

  background-repeat: no-repeat;

  background-attachment: fixed;

  object-fit: cover;

  overflow: hidden;

}

.container{

  max-width: 600px;

  width: 50%;

  background-color:white;

  border-radius: 5px;

  box-shadow: 0 5px 10px rgba(0,0,0,0.15);

}

.container .title{

  height: 70px;

  display: flex;

  align-items: center;

  justify-content: center;

  text-align: center;

  position: relative;

  background: #838d2f8a;

  font-size: 30px;

  font-weight: 600;

}

.content form .user-details{

  display: flex;

  flex-direction: column;

  justify-content: center;

  align-items: center;

}

form .user-details .input-box{

  margin-bottom: 15px;

  width: calc(100% / 2 - 20px);

}

form .input-box span.details{

  text-align: center;

  display: block;

  font-weight: 500;

  margin-bottom: 5px;

}

.user-details .input-box input{

  height: 45px;

  width: 100%;

  outline: none;

  font-size: 16px;

  border-radius: 5px;

  padding-left: 15px;

  border: 1px solid #ccc;

  border-bottom-width: 2px;

  transition: all 0.3s ease;

}

.user-details .input-box input:focus,

.user-details .input-box input:valid{

  border-color: #9b59b6;

}

 form .gender-details .gender-title{

  font-size: 20px;

  font-weight: 500;

 }

 form .category{

   display: flex;

   width: 80%;

   margin: 14px 0 ;

   justify-content: space-between;

 }

 form .category label{

   display: flex;

   align-items: center;

   cursor: pointer;

 }

 form .button{

   height: 45px;

   margin: 20px 0 20px;

 }

 form .button input{

  display: flex;

  justify-content: center;

  margin: auto;

   height: 100%;

   width: 50%;

   border-radius: 5px;

   border: none;

   color: rgb(0, 0, 0);

   font-size: 18px;

   font-weight: 500;

   letter-spacing: 1px;

   cursor: pointer;

   transition: all 0.3s ease;

   background: linear-gradient(135deg,  #838d2f8a, #b6b659);

 }

 form .button input:hover{

  /\* transform: scale(0.99); \*/

  background: #597ab6;

  }

 @media(max-width: 584px){

 .container{

  max-width: 100%;

}

form .user-details .input-box{

    margin-bottom: 15px;

    width: 100%;

  }

  form .category{

    width: 100%;

  }

  .content form .user-details{

    max-height: 300px;

    overflow-y: scroll;

  }

  .user-details::-webkit-scrollbar{

    width: 5px;

  }

  }

  @media(max-width: 459px){

  .container .content .category{

    flex-direction: column;

  }

}

   </style>

<body>

  <div class="container">

    <div class="title">Registration Form</div><br>

    <div class="content">

      <form action="/register" method="post">

        <div class="user-details">

          <div class="input-box">

            <span class="details">Username</span>

            <input name="username" type="text" placeholder="Enter your username" required>

          </div>

          <div class="input-box">

            <span class="details">Email</span>

            <input name="email" type="text" placeholder="Enter your email" required>

          </div>

          <div class="input-box">

            <span class="details">Password</span>

            <input name="password" type="text" placeholder="Enter your password" required>

          </div>

          <div class="input-box">

            <span class="details">Confirm Password</span>

            <input name="cpassword" type="text" placeholder="Confirm your password" required>

          </div>

          <div class="input-box">

            <span class="details">Initial Wallet amount</span>

            <input name="amount" type="number" step="5" placeholder="Amount of money" required>

          </div>

        </div>

        <div class="button">

          <input type="submit" value="Register">

        </div>

      </form>

    </div>

  </div>

</body>

</html>

**Styles.css**

/\*  import google fonts \*/

@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap");

\*{

  margin: 0;

  padding: 0;

  outline: none;

  border: none;

  text-decoration: none;

  box-sizing: border-box;

  font-family: "Poppins", sans-serif;

}

body{

  background: rgb(226, 226, 226);

}

nav{

  position: sticky;

  top: 0;

  bottom: 0;

  height: 100vh;

  left: 0;

  width: 280px;

  /\* width: 280px; \*/

  background: #fff;

  overflow: hidden;

  transition: 1s;

}

nav:hover{

  width: 280px;

  transition: 1s;

}

.logo{

  text-align: center;

  display: flex;

  margin: 10px 0 0 10px;

  padding-bottom: 3rem;

}

.logo img{

  width: 45px;

  height: 45px;

  border-radius: 50%;

}

.logo span{

  font-weight: bold;

  padding-left: 15px;

  font-size: 18px;

  text-transform: uppercase;

}

a{

  position: relative;

  width: 280px;

  font-size: 14px;

  color: rgb(85, 83, 83);

  display: table;

  padding: 10px;

}

nav .fas{

  position: relative;

  width: 70px;

  height: 40px;

  top: 20px;

  font-size: 20px;

  text-align: center;

}

.nav-item{

  position: relative;

  top: 12px;

  margin-left: 10px;

}

a:hover{

  background: #eee;

}

a:hover i{

  color: #34AF6D;

  transition: 0.5s;

}

.logout{

  position: absolute;

  bottom: 0;

}

.container{

  display: flex;

}

/\* MAin Section \*/

.main{

  position: relative;

  padding: 20px;

  width: 100%;

}

.main-top{

  display: flex;

  flex-direction: column;

  justify-content: center;

  align-items: center;

  gap: 30px;

  width: 100%;

}

.main-top i{

  position: absolute;

  right: 0;

  margin: 10px 30px;

  color: rgb(110, 109, 109);

  cursor: pointer;

}

/\*active page\*/

.active{

  background-color: #4AD489;

}

.active a:hover{

  background-color: #4AD489;

  color: #000;

}

.active a:hover i{

  color: #000;

}

.active i{

color: #000;

}

.active span{

  color: #000;

  font-weight: 600;

}

/\*input tags\*/

input[type=text], select {

    width: 100%;

    padding: 12px 20px;

    margin: 8px 0;

    display: inline-block;

    border: 1px solid #ccc;

    border-radius: 4px;

    box-sizing: border-box;

  }

  input[type=number], select {

    width: 100%;

    padding: 12px 20px;

    margin: 8px 0;

    display: inline-block;

    border: 1px solid #ccc;

    border-radius: 4px;

    box-sizing: border-box;

  }

  input[type=date], select {

    width: 100%;

    padding: 12px 20px;

    margin: 8px 0;

    display: inline-block;

    border: 1px solid #ccc;

    border-radius: 4px;

    box-sizing: border-box;

  }

  input[type=submit] {

    width: 100%;

    background-color: #4CAF50;

    color: white;

    padding: 14px 20px;

    margin: 8px 0;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    font-size: larger;

  }

  input[type=submit]:hover {

    background-color: #45a049;

    font-size: larger;

  }

label{

    font-weight: 700;

    font-size: larger;

}

  /\* expense table\*/

  #report {

    border-collapse: collapse;

    margin-left: -25%;

    width: 150%;

  }

  #report td, #report th {

    background-color: white;

    border: 1px solid rgb(231, 231, 231);

    padding: 8px;

  }

  #report tr:hover td {background-color: rgba(211, 211, 211, 0.88);}

  #report th {

    padding-top: 12px;

    padding-bottom: 12px;

    text-align: left;

    background-color: #04AA6D;

    color: white;

  }

  .homecard{

    display: flex;

    flex-direction: row;

    justify-content: center;

    align-items: center;

    gap: 50px;

  }

  .homecard div{

    display: flex;

    flex-direction: column;

    border: 1px solid black;

    border-radius:5px ;

    width: 250px;

    height: 250px;

    padding: 20px;

  }

  .homecard div h1{

    font-size: 50px;

    font-weight: 500;

    color: #297b2e;

    align-items: center;

    margin: auto;

  }

**Report.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <title>Personal Expense Tracker</title>

  <link rel="stylesheet" href="static/styles.css" />

  <!-- Font Awesome Cdn Link -->

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css"/>

</head>

<body>

  <h1 style="text-align: center;   background: #838d2f8a; padding: 7px;">Personal Expense tracker</h1>

  <div class="container">

    <nav>

        <ul>

          <li><a href="#">

            <i class="fas fa-bars" style="color: black;"></i>

            <span class="nav-item" Style="color:black; font-size:20px; font-weight:600" >Dashboard</span>

          </a></li>

          <li><a href="{{url\_for('home')}}">>

            <i class="fas fa-home"></i>

            <span class="nav-item">Home</span>

          </a></li>

        <li><a href="{{url\_for('expense')}}">

          <i class="fas fa-wallet"></i>

          <span class="nav-item">Add Expense</span>

        </a></li>

        <li class="active"><a href="#">

            <i class="fas fa-clipboard-list"></i>

            <span class="nav-item">Expense log</span>

          </a></li>

        <li><a href="{url\_for('wel')}}">

          <i class="fas fa-sign-out-alt"></i>

          <span class="nav-item">Log out</span>

        </a></li>

      </ul>

    </nav>

    <section class="main">

      <div class="main-top">

        <h1>Expense log</h1>

        <div>

            <table id="report">

                <tr>

                  <th>Date</th>

                  <th>Amount</th>

                  <th>Category</th>

                  <th>Description</th>

                </tr>

                {% for acc in account %}

                <tr>

                  <td>{{ acc[4] }}</td>

                  <td>{{ acc[5] }}</td>

                  <td>{{ acc[6] }}</td>

                  <td>{{ acc[7] }}</td>

                </tr>

                {% endfor %}

              </table>

        </div>

      </div>

  </div>

</body>

<script>

  window.watsonAssistantChatOptions = {

    integrationID: "7f157622-db65-4138-8cca-60e657ef7461", // The ID of this integration.

    region: "us-south", // The region your integration is hosted in.

    serviceInstanceID: "5ee64e34-e58d-45b6-8d1b-e739162fb7cf", // The ID of your service instance.

    onLoad: function(instance) { instance.render(); }

  };

  setTimeout(function(){

    const t=document.createElement('script');

    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

    document.head.appendChild(t);

  });

</script>

</html>

**App.py**

from flask import Flask,url\_for,redirect,render\_template,request,session,sessions

import ibm\_db

import ibm\_db\_dbi as db2

import socket

import re

hostname = socket.gethostname()

ip = socket.gethostbyname(hostname)

app=Flask(\_\_name\_\_)

app.secret\_key='a'

con=db2.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=cch66020;PWD=lI1IGk5RBsX1sjkG",'','')

cur=con.cursor()

conn=ibm\_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=cch66020;PWD=lI1IGk5RBsX1sjkG",'','')

@app.route('/wel')

def wel():

    return render\_template("loginpage.html")

@app.route('/registerpage')

def registerpage():

    return render\_template("registration.html")

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

def register():

    msg=''

    if request.method=='POST':

        email=request.form['email']

        passwrd=request.form['password']

        name=request.form['username']

        amount=request.form['amount']

        insert\_sql="INSERT INTO LOGIN VALUES(?,?,?,?)"

        prep\_stmt=ibm\_db.prepare(conn,insert\_sql)

        ibm\_db.bind\_param(prep\_stmt,1,email)

        ibm\_db.bind\_param(prep\_stmt,2,passwrd)

        ibm\_db.bind\_param(prep\_stmt,3,name)

        ibm\_db.bind\_param(prep\_stmt,4,amount)

        ibm\_db.execute(prep\_stmt)

        msg='Ticket is raised successfully!'

        return render\_template('expense.html',msg=msg)

    else:

        msg='Query submission Unsuccessful! Please try again later!'

        return render\_template('registration.html',msg=msg)

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

def login():

    global userid

    msg=''

    if request.method=='POST':

        username=request.form['email']

        pwd=request.form['pass']

        sql="SELECT \* FROM LOGIN WHERE EMAIL=? AND PASSWORD=?"

        stmt=ibm\_db.prepare(conn,sql)

        ibm\_db.bind\_param(stmt,1,username)

        ibm\_db.bind\_param(stmt,2,pwd)

        ibm\_db.execute(stmt)

        acc=ibm\_db.fetch\_assoc(stmt)

        if acc:

            session['loggedin'] = True

            session['id'] = acc['USERNAME']

            session['amnt']=acc['AMOUNT']

            session['exp']=acc['EXPENSE']

            userid = acc["USERNAME"]

            session['email'] = acc["EMAIL"]

            msg='Login successful!'

            return render\_template("expense.html")

        else:

            msg="Invalid username/password!"

            return render\_template("loginpage.html")

@app.route('/report')

def report():

    return render\_template("report.html")

@app.route('/expense')

def expense():

    return render\_template("expense.html")

@app.route('/home')

def home():

    return render\_template("home.html")

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

def adding():

    if request.method=='POST':

        oldexpense=session['exp']

        expense=(float)(request.form['amount'])

        cat=request.form['category']

        des=request.form['note']

        name=session['id']

        amnt=session['amnt']

        insert\_sql="UPDATE LOGIN SET EXPENSE=?,BALANCE=?,CATEGORY=?, DESCRIPTION=? WHERE USERNAME=?;"

        prep\_stmt=ibm\_db.prepare(conn,insert\_sql)

        ibm\_db.bind\_param(prep\_stmt,1,(oldexpense+expense))

        ibm\_db.bind\_param(prep\_stmt,2,(amnt-expense))

        ibm\_db.bind\_param(prep\_stmt,3,cat)

        ibm\_db.bind\_param(prep\_stmt,4,des)

        ibm\_db.bind\_param(prep\_stmt,5,name)

        ibm\_db.execute(prep\_stmt)

        msg='Expense added'

        return render\_template('expense.html',msg=msg)

@app.route('/display')

def display():

    cur.execute("SELECT \* FROM CCH66020.LOGIN where NAME=?")

    account = cur.fetchall()

    return render\_template('report.html',account=account)

if \_\_name\_\_ == '\_\_main\_\_' :

    app.run(debug=True)

**deploy.yaml**

 apiVersion: apps/v1

 kind: Deployment

 metadata:

   name: flask-node-deployment

 spec:

   replicas: 1

   selector:

     matchLabels:

       app: flasknode

   template:

     metadata:

       labels:

         app: flasknode

     spec:

       containers:

       - name: flasknode

         image: us.icr.io/personalexpensetracker/personalexpensetracker

         imagePullPolicy: Alw

         ports:

         - containerPort: 5000

**Dockerfile:**

FROM python:3.6

WORKDIR /app

ADD . /app

COPY requirements.txt /app

RUN python3 -m pip install -r requirements.txt

RUN python3 -m pip install ibm\_db

EXPOSE 5000

CMD ["python","app.py"]

**GITHUB AND PROJECT DEMO LINK**

**GitHub Link:** [**https://github.com/IBM-EPBL/IBM-Project-21594-1659785541**](https://github.com/IBM-EPBL/IBM-Project-21594-1659785541)

**Project Demo Link:**