## **EXPENSE TRACKER**

## **18CSC206J - SOFTWARE ENGINEERING AND PROJECT MANAGEMENT LAB**

## A Project Report

## *Submitted by*

## ADITI KHATRI [RA1911003010984]

## PRIYANSHA SINGH [RA1911003010982]

## TANYA SABARWAL [RA1911003010994]

## *Under the Guidance of*

## DR. MUTHU KUMARAN.A

## (Assistant Professor, Department of Computer Science and Engineering)

## 

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

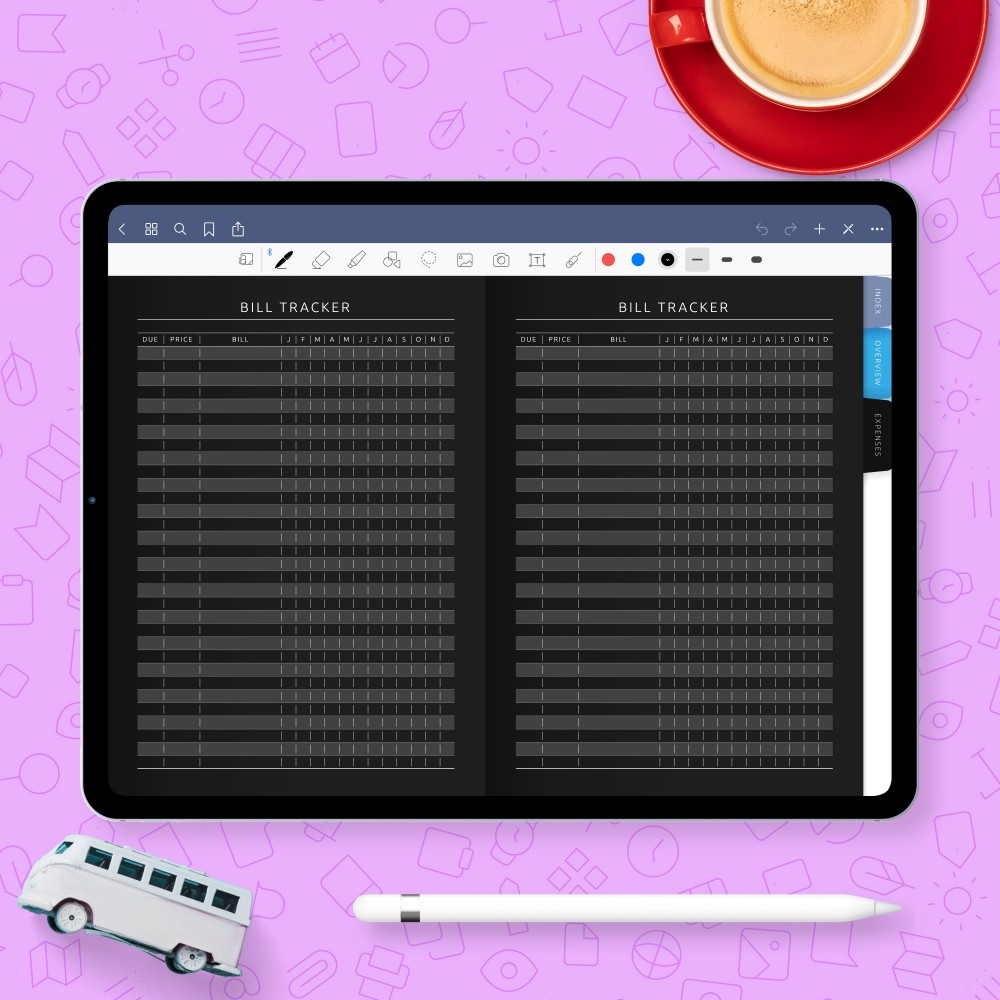
## **FACULTY OF ENGINEERING AND TECHNOLOGY**

## **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

## **KATTANKULATHUR - 603 203**

## **JUNE 2021**

## 



Expense Tracker JUNE 2021

**Team members :**

Priyansha singh (RA1911003010982 ) ,Aditi khatri (RA1911003010984) , Tanya sabarwal (RA1911003010994)

***Expensify is a platform which can be exclusively used by anyone who wants to keep track of their expenses. It helps you stick to your budget and reveal spending issues. Keeping record here will be much easier than using pen and paper. Being a smart expense tracker, it is a voice-based application where you need to give instructions verbally and it will categorize according to it.***

# PROBLEM STATEMENT

**Aim**

To frame a project team, analyze and identify a Software project.

**Project Description:**

* Expensify is a platform which can be exclusively used by anyone who wants to keep track of their expenses.
* It helps you stick to your budget and reveal spending issues. Keeping record here will be much easier than using pen and paper.
* Being a smart expense tracker, it is a voice-based application where you need to give instructions verbally and it will categorize according to it.
* It has a graphical description for income as well as expenses category giving user a clear and detailed analysis about the spending and budget.
* Tech stack used here will be  React , JavaScript , Some of the concepts are State Management in React, Context API, Local Storage, Material UI, and adding voice capabilities resulting in a scalable live application ready to be used on daily basis.

**Aim:**

To create a business case and Arrive at a Problem Statement for the Smart Expense Tracker.

**Business Case**

**Table of Contents**

[***1.***](#_gjdgxs) ***Problem Statement and Executive Summary 1***

[***2.***](#_30j0zll) ***Strategic Business Context 1***

[**2.1.**](#_1fob9te) **Business Need *1***

[**2.2.**](#_3znysh7) **Business Outcomes *1***

[***3.***](#_2et92p0) ***Detailed Business Problem 2***

[**3.1.**](#_tyjcwt) **Problem/Opportunity Statement 2**

[**3.2.**](#_3dy6vkm) **High Level Requirements 2**

[**3.3.**](#_1t3h5sf) **Assumptions 2**

[**3.4.** **Constraints 2**](https://docs.google.com/document/d/1rTjxD5oaAwrgKyFH_1SKQHjGghtgbgnSmPbQc_Lsx9c/edit#heading=h.4d34og8)

[**3.5.** **Dependencies**](https://docs.google.com/document/d/1rTjxD5oaAwrgKyFH_1SKQHjGghtgbgnSmPbQc_Lsx9c/edit#heading=h.2s8eyo1) **2**

[**3.6.** **Stakeholder Analysis 3**](https://docs.google.com/document/d/1rTjxD5oaAwrgKyFH_1SKQHjGghtgbgnSmPbQc_Lsx9c/edit#heading=h.17dp8vu)

[***4.*** ***Project Team Structure 3***](https://docs.google.com/document/d/1rTjxD5oaAwrgKyFH_1SKQHjGghtgbgnSmPbQc_Lsx9c/edit#heading=h.3rdcrjn)

[**4.1.** **Roles and Responsibilities 3**](https://docs.google.com/document/d/1rTjxD5oaAwrgKyFH_1SKQHjGghtgbgnSmPbQc_Lsx9c/edit#heading=h.26in1rg)

[***Reference***](https://docs.google.com/document/d/1rTjxD5oaAwrgKyFH_1SKQHjGghtgbgnSmPbQc_Lsx9c/edit#heading=h.2bn6wsx) ***4***

***Result***

**PROBLEM STATEMENT:**

In short, the main reason you should track your expenses is to identify and eliminate wasteful spending habits in your financial life. Moreover, consistently tracking your expenses will help you maintain control of your finances, and promote better financial habits like saving and investing.

# Executive Summary

This business case outlines how the Smart Expense Tracker Project will address current business concerns, the benefits of the project, and recommendations and justification of the project. The business case also discusses detailed project goals, performance measures, assumptions, constraints, and alternative options.

# Strategic Business Context

## Business Need

Employees or customers often require a means to hold oneself accountable, tracking expenses on a day-to-day basis and helps to see progress on the road to financial goals. SET can deliver a solution to track expenses, and take complete control over finances.

Our project aims on helping the user keep a track record of his/her expenditure which will ultimately lead to an organized budget.

Business Outcomes

When you make an effort to record every financial transaction you make, you are essentially, taking the reins on anything and everything involving your money. At any one time, you will know exactly how much money is sitting in your bank account, and how much you can spend. Plans on saving, investing, getting out of debt, or building wealth, holding you accountable, setting financial goals, and have financial dreams can all be achieved and implemented.

# Detailed Business Problem

## **Problem/Opportunity Statement**

## Expense Tracker aims to help everyone who are planning to know their expenses and save from it. It helps you stick to your budget and revel spending issues. Keeping record here will be much easier than using pen and paper. Being a smart expense tracker, it is a voice-based application where you need to give instructions verbally and it will categorize according to it. It has a graphical description for income as well as expenses category giving users a clear and detailed analysis about the expenditure and budget.

## **High Level Requirements**

Requirements:

1. Registration –where users can create an account directly from the website and save all the information.
2. Group creation – information can be grouped into different categories.
3. Log expense –the user will enter all expense details in application and the module will calculate the dues.
4. The data will then be sync with the online service and update expense log in main database
5. Internet connection is required

## **Assumptions**

|  |  |
| --- | --- |
| **S.No** | **Assumptions** |
| 1 | We will get the app ready in 4 months. |
| 2 | We will achieve 20 thousand users in our first year. |
| 3 | We will hire all our employees by the end of next month. |

## **Constraints**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Category** | **Constraints** |
| 1 | Monetary | The employees initially might not be up to the skills required. |
| 2 | Privacy | Need access to voice recognition. |
| 3 | Monetary | Requires a budget of 2 lakhs. |
| 4 | Time | The employees will have to be hired by the end of next month. |

## **3.5 Dependencies**

No such dependencies are needed in this project.

## **3.6 Stakeholder Analysis**

|  |  |  |
| --- | --- | --- |
| **Name** | **Designation** | **Role in Project** |
| Priyansha Singh | Corporate Head for Sales & Marketing | Executive Sponsor |
| Tanya Sabarwal | Chief Information Officer or Regional Head of Sales & Marketing | Project Sponsor |
| Aditi Khatri | Finance Head | Cost Approver |
| Priyansha Singh | Business User(s) | Validate the functionalities |
| Aditi Khatri | IT Head | To Develop AI shopping Assistant |

## **4. Project Team Structure**

## **4.1 Roles & Responsibilities**

|  |  |  |
| --- | --- | --- |
| **Project Role** | **Responsibilities** | **Assigned To** |
| Project Manager | Responsible for planning, organizing, and directing the  completion of project for  the organization while ensuring this is on time, on budget, and within  scope. | Priyansha Singh |
| Technical Lead | Plan, design, develop, and launch efficient business, financial, and operations systems in support of  core organizational functions and business processes | Aditi Khatri |
| Business Analyst | Helping businesses implement technology  solutions in a cost-effective way by determining the requirements of a project  or program, and communicating them clearly to stakeholders, facilitators and  partners. | Tanya Sabarwal |
| Developer | Professional who designs, creates, and  customizes software applications to meet the business and functional  requirements | Aditi Khatri  Priyansha Singh  Tanya Sabarwal |
| Tester | Creating Test Plan and Develop Test Cases | Tanya Sabarwal |

# 

# Reference

1. [https://www.wikipedia.org/](https://www.pmi.org/)
2. <https://www.projectmanagement.com/>
3. [https://www.google.com/](https://www.projectmanagement.com/)

**Result**

Thus, the business case was prepared and the problem statement arrived.

# 

# 

# 

# 

# 

# 

# REQUIREMENTS

**Aim**

To Identify and document the Requirements of a Software system

**Table of Contents**

[***1.***](#_gjdgxs) ***Executive Summary 2***

[***2.***](#_30j0zll) ***Project Scope 2***

[**2.1.**](#_1fob9te) **In Scope 2**

[**2.2.**](#_3znysh7) **Out of Scope 2**

[***3.***](#_2et92p0) ***Epics [Major Functions] 2***

[***4.***](#_tyjcwt) ***Requirements 2***

[**4.1.**](#_3dy6vkm) **Functional Requirements 2**

[**4.2.**](#_1t3h5sf) **Non-Functional Requirements 3**

[**4.3.**](#_4d34og8) **Infrastructure Requirements 3**

[**4.4.**](#_2s8eyo1) **Requirement definition in Agile [Optional … Use according to methodology chosen by student] 4**

[***Reference 4***](#_17dp8vu)

# Executive Summary

Expensify is a platform which can be exclusively used by anyone who wants to keep track of their expenses. It helps you stick to your budget and revel spending issues. Keeping record here will be much easier than using pen and paper. Being smart expense tracker, it is a voice-based application where you need to give instructions verbally and it will categorize according to it.

This section elaborates on how we can have the requirements in various aspects of the software.

# Project Scope

Expand on the scope definition and outline the major activities required to successfully complete the project (for example, develop module ABC, develop requirements document, etc.). Out of scope activities are identified to reduce ambiguity.

|  |  |
| --- | --- |
| **S. No** | **Activities in Scope** |
| 1 | Create distinct product users based on their roles and permissions. |
| 2 | Authenticate users at their login |
| 3 | Provide the spaces for filling the necessary details |
| 4 | Facility to scan through the information entered |
| 5 | A status page for all users to know if their details are correctly entered and are end-to-end encrypted |
| 6 | A page for the user to enter all expense details in the application. |
| 7 | Instant notifications will be given where the expenditure is about to cross the given budget. |
| 8 | A weekly or monthly expenditure report of expenses will be generated. |
| 9 | Mechanism to reset the password incase the user forgets it. |

## In Scope

Design, Develop, Test and Implement Expense Tracker

## Out of Scope

Adding multi-platform support as it would require a lot of codebases so initially, we are only going for a desktop-based release.

# Epics [Major Functions]

|  |  |
| --- | --- |
| **Epic (#)** | **Epic Description** |
| E1 | Add money, date, category manually |
| E2 | Registration of user |
| E3 | update items |
| E4 | remove items |
| E5 | add using speech recognition |
| E6 | adding expense /income in pie chart representation |

# Requirements

## Functional Requirements

Functional Requirements can also be expressed in the form of “user story” which is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement (#)** | **Requirement Specification** | **Department** | **Name of Business User** | **Status** |
| E1FR1 | create account through application and authenticate on every login | technical department | All users | active |
| E1FR2 | overall report expense details for logged in user | technical department | All users | active |

Functional requirements:

1. Provide weekly notifications by an email
2. Overall report expense details for logged in user
3. Update sync to web service and propagate to another member of the group
4. Creating account through application or web service
5. Authenticate on every login

## Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement (#)** | **Category of NFR** | **Requirement Specification** |
| 1 | Performance | The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submission. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen. |
| 2 | Performance | Search should bring the results less than 7 seconds |
| 3 | Availability | The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week. |
| 4 | Scalability | Registration Service of new users should scale to serve 1000 request per second over 5 minutes timespan |
| 5 | Confidentiality | all Student and faculty data shall be store within the organization to which they belong |
| 6 | Accuracy | The system should accurately provide real time information taking into consideration various concurrency issues. The system shall provide 100% access reliability |
| 7 | Reliability | The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week, 24 hours a day.   1. **Privacy statement** — Anyone who would decide to use the expense tracker wants to first check the privacy statement and its terms and conditions. Keep the word “sell” or “share” in the privacy statement. This will make sure the tracker won’t sell your sensitive personal data to third-parties for targeted ads. 2. **Security level-** The tracker deals with financial information, so it should be using financial institution-level security measures. Tracker’s encryption standards for user data should be 128-bit(good) or 256-bit (better) and it should have [two-factor authentication](https://blog.avast.com/5-privacy-protection-tips) as a login option (it’ll add an extra layer of security on top of your password). 3. **Tracker owner-** Many budgeting apps are owned by larger financial institutions, which should have experience handling user data in a safe and secure manner. 4. **Password protect your device —**To make sure that no one can take a peek at finances, a password or PIN or fingerprint to unlock the home screen of the device should be implemented. This is also important for protecting actual bank accounts. 5. **Antivirus** — [Malware](https://www.avast.com/c-malware) targeting financial accounts is rise, a [robust antivirus](https://www.avast.com/en-us/free-mobile-security) should be made available. |
| 8. | Security | 1. **Privacy statement** — Anyone who would decide to use the expense tracker wants to first check the privacy statement and its terms and conditions. Keep the word “sell” or “share” in the privacy statement. This will make sure the tracker won’t sell your sensitive personal data to third-parties for targeted ads. 2. **Security level-** The tracker deals with financial information, so it should be using financial institution-level security measures. Tracker’s encryption standards for user data should be 128-bit(good) or 256-bit (better) and it should have [two-factor authentication](https://blog.avast.com/5-privacy-protection-tips) as a login option (it’ll add an extra layer of security on top of your password). 3. **Tracker owner-** Many budgeting apps are owned by larger financial institutions, which should have experience handling user data in a safe and secure manner. 4. **Password protect your device —**To make sure that no one can take a peek at finances, a password or PIN or fingerprint to unlock the home screen of the device should be implemented. This is also important for protecting actual bank accounts. 5. **Antivirus** — [Malware](https://www.avast.com/c-malware) targeting financial accounts is rise, a [robust antivirus](https://www.avast.com/en-us/free-mobile-security) should be made available. |

## Infrastructure Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement (#)** | **Requirement Specification** | **Department** | **Name of Business User / Project Team Member** | **Status** |
| IR1 | Development Machine with 6 GB Ram and 4 Cores | Development | Tanya Sabarwal | Done |
| IR2 | Code Repository | Development | Aditi Khatri | Pending |
| IR3 | AWS S3 Bucket | Development | Priyansha Singh | Pending |
| IR4 | IDE – visual code studio | Development | Aditi Khatri | Pending |
|  |  |  |  |  |

## Requirement definition in Agile [Optional … Use according to methodology chosen by student]

User story is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

|  |  |  |
| --- | --- | --- |
| **User Story** | **Acceptance Criteria** | **Size of User Story** |
| As a student/faculty, I can view all my expenses in an organized manner. I can set limits for the next expense according to the available financial status in my account | View all expenses of 15 days period  Allow to put a tracker for the next specified time period  Allow to notify me about the incoming and outgoing money in the account | large |

# Reference

1. <https://www.pmi.org/>
2. <https://www.atlassian.com/agile/project-management/user-stories>

Result:

Thus, the requirements are identified, collected and documented.

**Aim:** To Prepare Project Plan based on scope, Find Job roles and responsibilities, Calculate Project effort based on resources

**Table of Contents**

[***1.***     ***Executive Summary   2***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.gjdgxs)

[***2.***     ***Project Management Plan 2***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.30j0zll)

[***3.***     ***Estimation  3***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.1fob9te)

[***3.1.***  ***Effort and Cost Estimation   3***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.3znysh7)

[***3.2.***  ***Infrastructure/Resource Cost [CapEx]     3***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.2et92p0)

[***4.***     ***Maintenance and Support Cost [OpEx]   4***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.tyjcwt)

[***5.***     ***Project Team Formation   4***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.3dy6vkm)

[***5.1.***  ***Identification Team members       4***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.1t3h5sf)

[***5.2.***  ***Responsibility Assignment Matrix 4***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.4d34og8)

[***Reference   5***](https://docs.google.com/document/d/1bwQoJqP_QYUGpqiibKtWpc0f9qlgIji--cYdrDD4n4k/edit#heading=h.2s8eyo1)

**1.** **Executive Summary**

 Project management plan is a detailed document in which the project executing, controlling and monitoring is defined. It further segregated in several parts such as Schedule Management, Stakeholder identification, Communication management, Risk management etc. Cost and effort estimation is the process of predicting the most realistic amount of cost and effort required. There are several techniques for estimating the number of efforts such as COCOMO I, COCOMO II etc. The team must consist of people of certain skill sets such as planning, monitoring, developing, executing and testing etc.

**2.**              **Project Management Plan**

|  |  |
| --- | --- |
| Focus Area | Details |
| Cost Management | Effort:   * Design and User interface: Tanya Sabarwal   build interfaces in software or computerized devices, focusing on looks or style.   * Database Implementation: Priyansha Singh   The implementation phase is where you install the DBMS on the required hardware, optimize the database to run best on that hardware and software platform, and create the database and load the data.   * Training Core ML Model: All members   on-device machine learning features, like object detection in images and video, language analysis, and sound classification with just a few lines of code.   * Integration of Frontend with Backend: Aditi Khatri   In this implementation, middleware or a similar integration solution intercepts interaction between applications. The middleware acts as a middle tier application server: transparently marshalling, managing, and directing underapplication requests and responses. Typically, the requesting application waits while the middleware requests services of another application.   * Final Deployment: All members   **BUDGET-**   * **Plan for Inflation** * **Other Unexpected Costs** * **Track in Real-Time** * **Size Accordingly** * **It is critical that figures be as precise and accurate as possible.** * **Establish how the** [**project performance will be measured**](https://www.projectmanager.com/blog/5-ways-to-measure-project-success) **in order to see if you’re meeting the goals and expectations of the project.** |
| Quality Management | The relevant methodologies, guidelines should be maintained for following items   * Standards of Risk Management must be maintained. * Records management, web publishing, information security, privacy, etc. * Relevant business domain driven standards   Quality control:   * Business Owner(s) - The Business Owners need to contribute resources to the project during their development to ensure that the outputs are being developed satisfactorily. * Advisory Groups: provide advice or technical expertise in relation to output development and quality assurance. * Reference Group: give a forum to achieve consensus among groups of stakeholders. * Working Group(s) - consist of small specialist work groups, which is dedicated to producing a well-defined output within a specific timeframe. * Consultants – which provide advice about the development of specific outputs. |
| Schedule  Management | Milestones denote the start and the finish of a project and mark the completion of a major phase of work.       ● Backend Core Deep Learning Models       ● Backend Development for App       ● Frontend Development for UI/UX       ● Database Implementation from Genuine Resources       ● Testing Process Schedule determines the current status and the influencing factors that cause schedule changes and accordingly manages the changes |
| Stakeholder | 1. Project Manager: Priyansha Singh  Will be responsible for planning and monitoring the project and accountable for the entire project.   2. Developer: Aditi  Responsible for developing the entire model and ensures the requirements of the project are fulfilled.  3. Business Analyst: Tanya  Creates a detailed business analysis and outlines problems and solutions for the project in a cost-effective way.  4. Beta-Tester: All members  Considering different aspects each one  will be making  sure App is viable in all its aspects .This plates important role in post development phase |
| Communication Management | A communication plan outlines who you need to communicate with about what, how you're going to do it, and how often.      1. A meeting is conducted daily.      2. Discussions will be held on progress and problems faced.      3. Further, discussions will be held on the plan of the next day.      4. If additional requirements are specified by the clients, they will be noted |
| Risk Management | It is the process of identifying and controlling threats to the software model.       1. Hacking.       2. Risk of uneasiness.       3. Model being unable to predict accurate data.       4. Unavailability of Data to Train core models |
| Procurement Management | Procurement is the act of obtaining goods, supplies, and/or services. Therefore, project procurement is obtaining all of the materials and services that are required for the project. Project procurement management encompasses the processes used for making sure project procurement is successful. Project procurement management processes. Project procurement management includes three primary processes. These are: Plan Procurements, Conduct Procurements, Administer or control the Procurements. |
| Scope management | **Scope management** is the process whereby the outputs, outcomes and benefits are identified, **defined** and controlled. '**Scope**' is the term used in the **management** of projects to refer to the totality of the outputs, outcomes and benefits and the work required to produce them.   * By determining the project scope, the time or cost that the project will take up can be estimated easily * Scope management builds control processes to address elements that may alter project during the project life-cycle * It helps to avoid the challenges that a project might encounter with every growing scope and unruly requirement list. |
| Resource management | In this phase, the Team manager will be pre-**planning**, scheduling, and allocating the available **resources** to maximize efficiency.  This benefits a team by Providing Unmatched visibility, Reduction in bench-time with consistent and persistent data.  Some of the pre-planned actions are as follows:   * Design and User interface: Tanya Sabarwal   build interfaces in software or computerised devices, focusing on looks or style.   * Database Implementation: Priyansha Singh   The implementation phase is where you install the DBMS on the required hardware, optimize the database to run best on that hardware and software platform, and create the database and load the data.   * Training Core ML Model: All members   on-device machine learning features, like object detection in images and video, language analysis, and sound classification with just a few lines of code.   * Integration of Frontend with Backend: Aditi Khatri   In this implementation, middleware or a similar integration solution intercepts interaction between applications. The middleware acts as a middle tier application server: transparently marshalling, managing, and directing inter-application requests and responses. Typically, the requesting application waits while the middleware requests services of another application. |
| Integration Management | **Governance Framework:**  The expense tracker Governance Framework is designed to assure adequate accountability to our many stakeholders and to encourage performance improvement while meeting our obligations and legislative requirements.  Statement: Expense Tracker aims to help everyone who is planning to know their expenses and save. It helps you stick to your budget and reduce expenditure. Being a smart expense tracker, it is a voice-based application where you need to give instructions verbally and it will categorize according to it.  It has a graphical description for income as well as expenses category giving users a clear and detailed analysis about the expenditure and budget.  **Project Team Structure**    **Roles and Responsibilities of Team**:   1. Project Manager: Priyansha Singh responsible for planning and   monitoring the project and accountable for the entire project.   2. Developer: Aditi responsible for developing the entire model and ensures the requirements of the project are fulfilled.   3. Business Analyst: Tanya creates a detailed business analysis and  outlines problems and solutions for the project in a cost-effective way.   4. Beta-Tester : All members in different aspects will be making  sure the App is viable in all its aspects .This plays an important role in the post development phase.  **Change Control:** Any changes to output specifications will be to controlled through a change process which includes:   * A structured process to facilitate the change to the system. * Complete assessment of the impact of the projected changed. * A method of authorizing a change.   **Management of changes (Issue management):**   * Planning for possible changes through proper risk analysis * Keeping track of all types of possible unanticipated issues * Using an iterative approach to make change within the scope of a single project. * Reflecting changes of project scope in Project Business Plan   **Project Closure:**   1. All required features and functionality were developed, tested and implemented.   2.      A service support process was established.  3.      A service management model and team was established and engaged. |

**3.**              **Estimation**

**3.1.** **Effort and Cost Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **WBS** | **Activity** | **Activity Description** | **Sub-Task** | **Sub-Task Description** | **Effort (in hours)** | **Cost in INR** |
| E1FR1 | E1R1A1 | Design the user screen | E1R1A1T1 | Confirm the user requirements (acceptance criteria) | 3 | 1500 |
| E1R1A1T2 | Making UI/UX | 5 | 3000 |
| E1FR2 | E1R1A2 | Database Implementation | E1R1A2T2 | Creating a database | 3 | 2500 |
| E1R1A2T3 | Linking the database to the ML mode | 2 | 1500 |
| E1FR3 | E1R1A3 | Training Core ML model | E1R1A3T1 | Collection Of Data From Viable Resources | 3 | 1500 |
| E1R1A3T2 | Pre-processing the data | 2 | 500 |
| E1R1A3T3 | Selection of a Required Highly Correlated Data | 5 | 3000 |
| E1R1A3T4 | Setting Up APIs to Work With and fetch data in real time | 1 | 500 |
| E1R1A3T5 | Training the model and evaluating the accuracy of the model | 4 | 3000 |
| E1FR4 | E1R1A4 | Integration Of Frontend with Backend | E1R1A4T1 | Connecting Flask App with data | 1 | 2000 |
| E1R1A4T2 | Integrate API with Front end | 1 | 500 |
| E1R1A4T3 | Improving UI/UX in Frontend | 4 | 1500 |
| E1FR5 | E1R1A5 | Final Deployment | E1R1A5T1 | Final Deployment on Heroku | 2 | 3000 |

|  |  |
| --- | --- |
| **Effort (hr)** | **Cost (INR)** |
| 36 | 24,000 |

**3.2.**              **Infrastructure/Resource Cost [CapEx]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Infrastructure Requirement** | **Qty** | **Cost per qty** | **Cost per item** |
| Laptop Systems | 5 | 80,000 | 4,00,000 |
| Stable Internet Connection | 3 | 5,000 | 15,000 |
| Cloud Memory(5TB) | 4 | 20,000 | 80,000 |

**4.**              **Maintenance and Support Cost [OpX]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Details** | **Qty** | **Cost per qty per annum** | **Cost per item** |
| People | Network, System, Middleware and DB admin    Developer, Support Consultant | 3    5 | 2,00,000    5,00,000 | 6,00,000    25,00,000 |
| License | Operating System  Database  Middleware  IDE | 10 | 10000 | 1,00,000 |
| Infrastructures | Server, Storage and Network | 20 | 20000 | 400,000 |

**5.**              **Project Team Formation**

**5.1.** **Identification Team members**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibilities** |
| Priyansha SIngh | Key Business User (Product Owner) | Provide clear business and user requirements |
| Priyansha Singh | Project Manager | Manage the project |
| Tanya Sabarwal | Business Analyst | Discuss and Document Requirements |
| Aditi Khatri | Technical Lead | Design the end-to-end architecture |
| Tanya Sabarwal | UX Designer | Design the user experience |
| Aditi Khatri | Frontend Developer | Develop user interface |
| Priyansha Singh | Backend Developer | Design, Develop and Unit Test Services/API/DB |
| Tanya Sabarwal | Cloud Architect | Design the cost effective, highly available and scalable architecture |
| Tanya Sabarwal | Cloud Operations | Provision required Services |
| Priyansha Singh | Tester | Define Test Cases and Perform Testing |

**5.2.**              **Responsibility Assignment Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RACI Matrix** | **Team Members** | | | |
| **Activity** | **Name (BA)** | **Name (Developer)** | **Name (Project Manager)** | **Key Business User** |
| User Requirement Documentation | A | C/I | I | R |
| Design the user interface | A | I | C | I |
| Training ML model | R | C | A | I/C |
| Integration Of Frontend or Backend | R | R/C | C/A | I |

|  |  |
| --- | --- |
| A | Accountable |
| R | Responsible |
| C | Consult |
| I | Inform |

**Reference**

1.     <https://www.pmi.org/>

2.     <https://www.projectmanagement.com/>

Result: Thus, the Project Plan was documented successfully.

**1. Executive Summary**

In this document, we have decided on a work breakdown structure with planned start and end dates for each activity/deliverable discussed in the previous documents. We have assigned a member of our team to each sub task. Our planned end date for finishing this project is the beginning of May i.e 7 May 2021. The planned end date for our last activity is 30 th April which gives us one week of buffer time to go over any problems in our project. Along with the work breakdown structure, we have identified the risks in our project and have created a risk plan to deal with the same

**2. WBS With Project Schedule**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Modul  e (#) | Activity (#) | Sub-Task(#) | Assignee( s) | Planne  d Start Date | Planne  d End  Date | Actu  al  Start  Date | Actu  al  End  Date | Statu  s |
| E1 | Design  Landing  Page | Prepare a  digital  design for  dashboard | Tanya  Sabarwal | 7-3-21 | 10-3-  21 |  |  | Not  starte  d |
| Convert the design in  the App. | Priyansha Singh | 11-3-  21 | 16-3-  21 |  |  | Not  starte  d |
| E2 | Design  registration page | Design the registration page | Tanya  Sabarwal | 17-3-  21 | 19-3-  21 |  |  | Not  starte  d |
| Create  database for storing user details | Aditi  khatri | 20-3-  21 | 23-3-  21 |  |  | Not  starte  d |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E3 | Database  Implementati on | Gathering  the data i.e. text mining | Aditi  Khatri | 20-3-  21 | 30-3-  21 |  |  | Not  starte  d |
| Linking the database to the website. | Aditi  Khatri | 31-3-  21 | 6-4-21 |  |  | Not  starte  d |
| E4 | Implementin g Model | Creating  model | Priyansha Singh | 7-4-21 | 15-4-  21 |  |  | Not  starte  d |
| Training the model on a dataset. | Priyansha Singh | 15-4-  21 | 30-4-  21 |  |  | Not  starte  d |
| E5 | Documentati on | Completing detailed  documentati on for entire project | Tanya  Sabarwal | 5-5-21 | 7-5-21 |  |  | Not  starte  d |

**3. Risk Identification**

**3.1.List (Describe) Register**

|  |  |  |
| --- | --- | --- |
| **Risk**  **ID (#)** | **Risk Description** | **Impact**  **Description** |
| R01 | Extra lines of codes or redundant algorithms may cause wastage of memory. | High |

|  |  |  |
| --- | --- | --- |
| R02 | Poor quality of the source image may cause incorrect output | High |
| R03 | Without extracting features the character complexity of ANN will be increased. | Low |
| R04 | If a customer provides irrelevant information then it may generate some unknown risk. | High |
| R05 | If a customer asks for change or gives some unexpected modification in later stages of development then it is difficult to alter the entire system design in accordance with that change. | High |

**3.2 Managing risks**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk**  **ID**  **(#)** | **Status**  **[Open /**  **Closed]** | **Risk Appetite [ Accept/**  **Mitigate/**  **Transfer/Avoid]** | **Action** | **Action**  **Owner** | **Target Date** | **Remarks** |
| R01 | Open | Avoid | We can avoid this risk by using  redundant code. | Aditi Khatri | 20-4-  2021 | In  Process |
| R02 | Open | Mitigate | Before processing for segmentation  using suitable noise removal algorithms will reduce this risk. | Priyansha Singh | 25-4-  20121 | In  Process |
| R03 | Open | Mitigate | Extract the Feature of character to  reduce complexity of ANN. | Aditi  Khatri | 4-5-  2021 | In  Process |
| R04 | Open | Transfer | We will try to improve the quality of the  model by training it on a large data set. | Tanya  Sabarwal | 15-4-  2021 | In  Process |
| R05 | Open | Transfer | Properly design the system with a  flexibility to adopt changes in later | Priyansha Singh,  Tanya  Sabarwal | 6-5-  2021 | In  Process |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | stages and also  maintain all  necessary  documentation for the same. |  |  |  |

**Reference**

1. https://www.pmi.org/

**DESIGN**

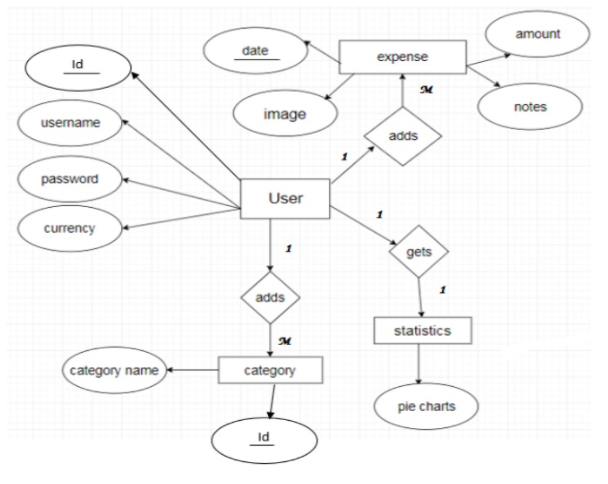
**Use case diagram**:

There are two actors in this system i.e the user and the administrator. The user can add expenses, date, category, bills,reports and also view expenses. On the basis of these inputs our application will show the trends.

* Used to gather the requirements of a system.
* Used to get an outside view of a system.
* Identify the external and internal factors influencing the system.
* Show the interaction among the requirements are actors.

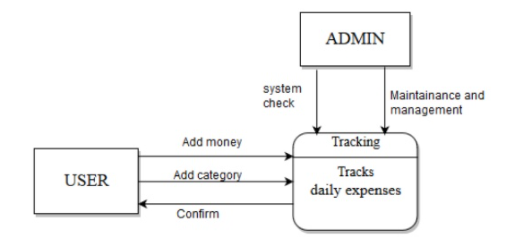


**Entity Relationship Diagram:**



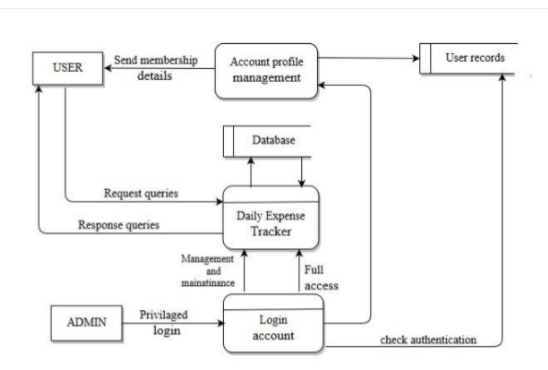
The above diagram explains the relationship between the databases where rectangle represents entity, oval represents attributes and diamond represents relationships.There are four entities with their respective attributes.

**DFD(Data Flow Diagram) LEVEL-0**



The user adds money to the tracking system, the system then updates it to our local database and then the database will retrieve it to the system. The data is then confirmed by the user.

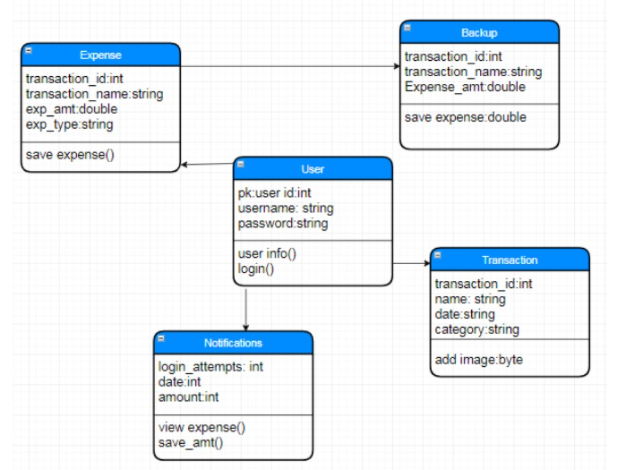
**DFD LEVEL 1**



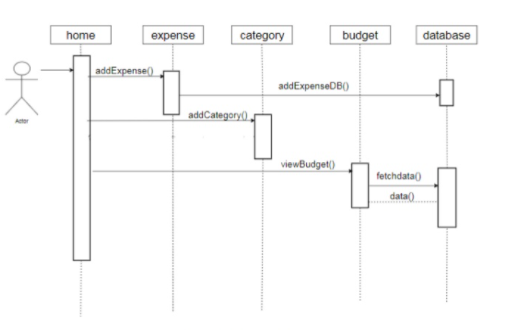
Here we can track our data(expenses).

**CLASS DIAGRAM:**

* Describing the static view of the system.
* Showing the collaboration among the elements of the static view.
* Describing the functionalities performed by the system.
* Construction of software applications using object oriented languages.



**SEQUENCE DIAGRAM:**

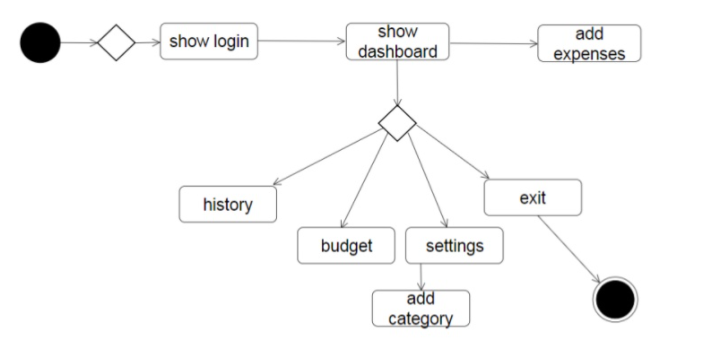


## The lifelines here are home, expense, category, budget and database.

## Purpose:

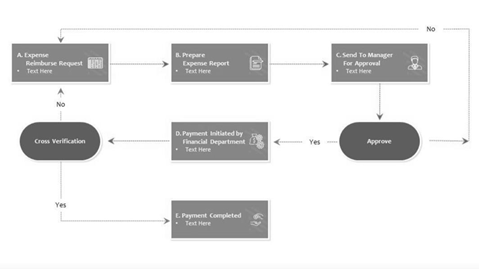
1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.

**STATECHART DIAGRAM:**



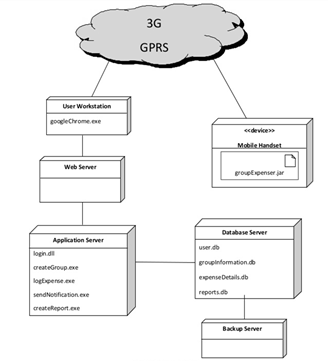
When the user visits the site, we transit from the initial state to login state.The user dashboard appears which shows history, budget and settings.After the user is done editing the details we transit to the final state.

**COLLABORATION DIAGRAM:**



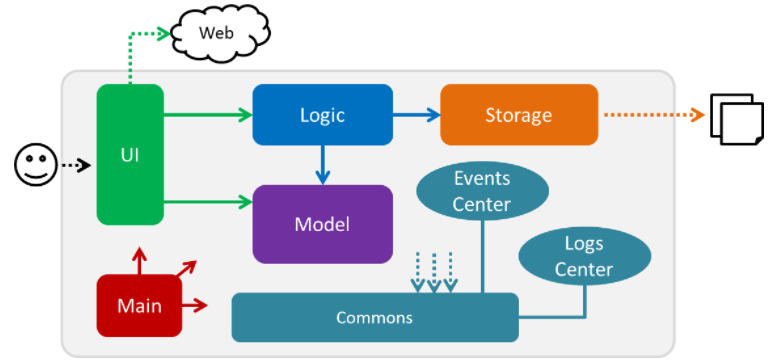
Shows how objects interact to perform the behaviour of a particular use case, or a part of a use case. First the user enters his expenses i.e A then a report is generated i.e B and the other steps follow.

**DEPLOYMENT DIAGRAM:**

****

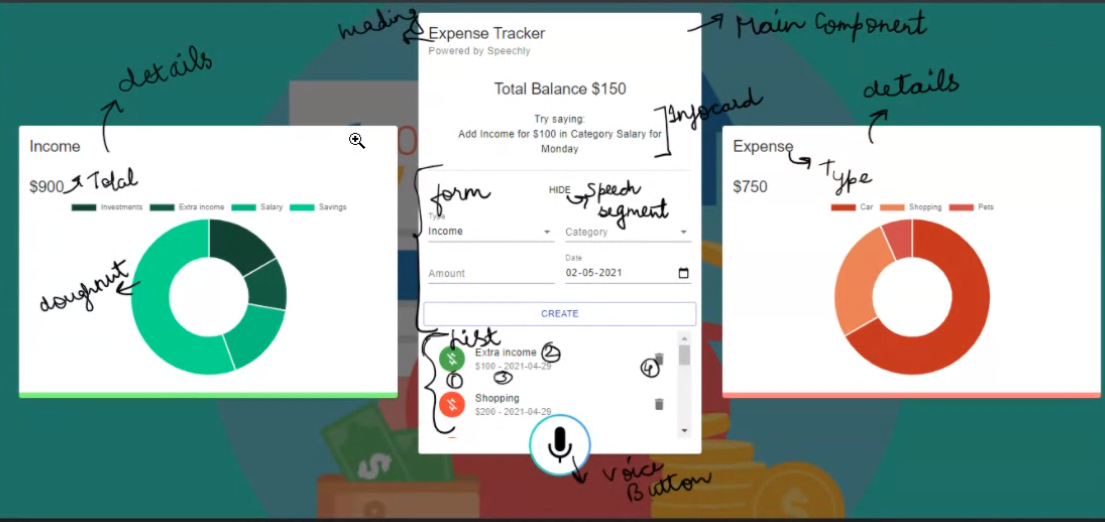
It shows the configuration of run time processing nodes and the components that live on them. Here the nodes are user workstation, application server, web server,mobile handset, database server and backup server.

**ARCHITECTURE DIAGRAM:**



* MAIN is responsible for:
  + Initializing the components in the correct sequence and connecting them up with each other when the app is launched.
  + Shutting down the components and invoking cleanup methods.
* COMMONS represents a collection of classes used by multiple components.
* LOGIC is the command executor.It defines the API.
* MODEL holds the data of the app in-memory.
* STORAGE reads the data from and writes data to the local data storage.

# UI DESIGN

****



# MODULES

**IMPLEMENTATION**

# **FRONT END TOOLS**:

**HTML5**:

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets.

**CSS3:**

Cascading Style Sheets (CSS) is a language that is used to illustrate the look, style, and format of a document written in any markup language. In simple words, it is used to style and organize the layout of Web pages.

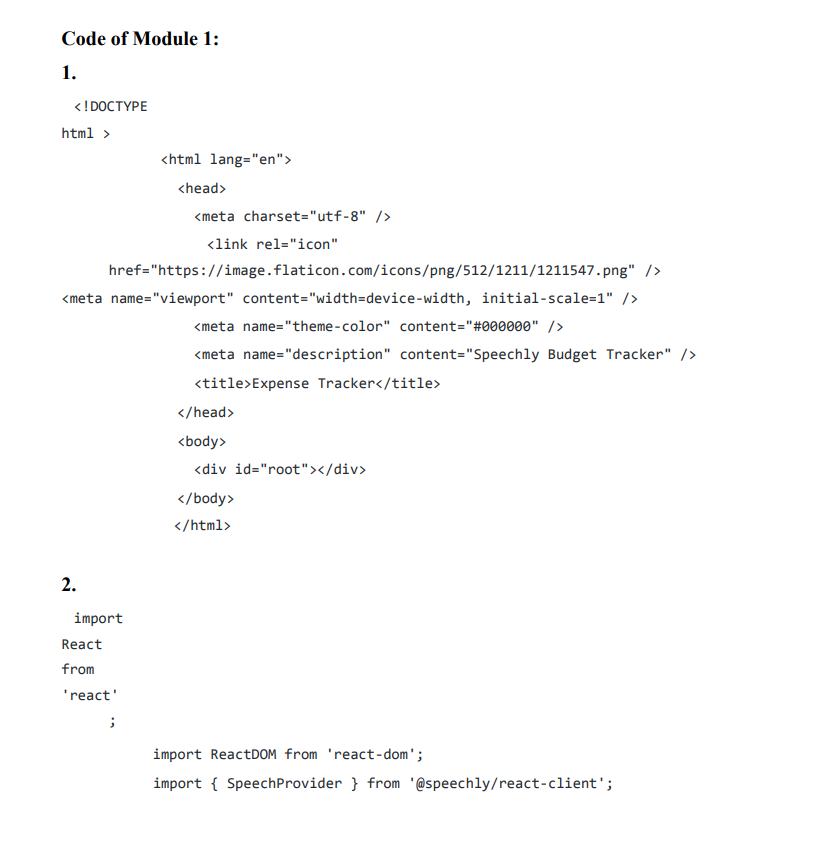
**JAVASCRIPT:**

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

**REACT:**

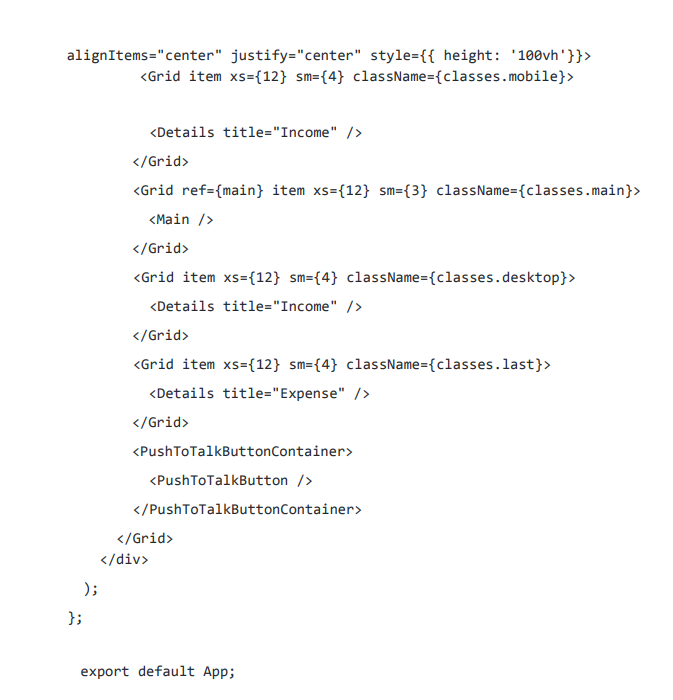
React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components.

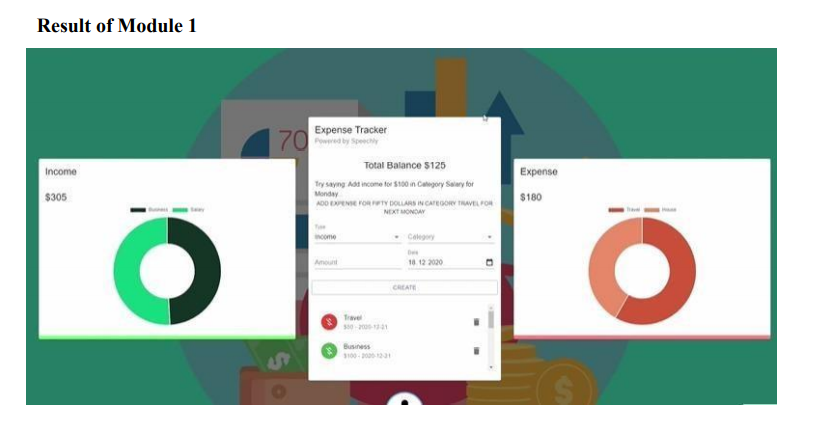
# DATABASE: Local storage.











# Code of Module 2

**THIS MODULE SHOWS THE CODE FOR THE FRONTEND DEVELOPMENT**

1. MAIN FILE

import React, { useEffect, useRef } from 'react';

import { Grid } from '@material-ui/core';

//grid to position the cards

import { SpeechState, useSpeechContext } from "@speechly/react-client";

import { PushToTalkButton, PushToTalkButtonContainer } from '@speechly/react-ui';

import { Details, Main } from './components';

import useStyles from './styles';

const App = () => {

const classes = useStyles();

// to auto go up while speaking inmobile.

const { speechState } = useSpeechContext();

const main = useRef(null)

const executeScroll = () => main.current.scrollIntoView()

useEffect(() => {

if (speechState === SpeechState.Recording) {

executeScroll();

}

}, [speechState]);

return (

<div>

<Grid className={classes.grid} container spacing={0} alignItems="center" justify="center" style={{ height: '100vh'}}>

<Grid item xs={12} sm={4} className={classes.mobile}>

<Details title="Income" />

</Grid>

<Grid ref={main} item xs={12} sm={3} className={classes.main}>

<Main />

</Grid>

<Grid item xs={12} sm={4} className={classes.desktop}>

<Details title="Income" />

</Grid>

<Grid item xs={12} sm={4} className={classes.last}>

<Details title="Expense" />

</Grid>

<PushToTalkButtonContainer>

<PushToTalkButton />

</PushToTalkButtonContainer>

</Grid>

</div>

);

};

export default App;

//useref for scrolling up auto while speaking on phone

1. DETAILS COMPONENT

import React from 'react';

import { Card, CardHeader, CardContent, Typography } from '@material-ui/core';

import { Doughnut } from 'react-chartjs-2';

import useStyles from './styles';

import useTransactions from '../../useTransactions';

//hooks

const DetailsCard = ({ title, subheader }) => {

// getting chart data from usetrans.js

const { total, chartData } = useTransactions(title);

const classes = useStyles();

return (

<Card className={title === 'Income' ? classes.income : classes.expense}>

<CardHeader title={title} subheader={subheader} />

<CardContent>

{/\* total \*/}

<Typography variant="h5">${total}</Typography>

{/\* piechart :use a hook namedusetrans \*/}

<Doughnut data={chartData} />

</CardContent>

</Card>

);

};

export default DetailsCard;

// details of 2 card income and expense ;topo :text ,doughnut :piechart

// title is a prop to get diffrent headings that we are getting from const deltails

1. MAIN COMPONENT

import React, { useState, useEffect, useContext } from 'react';

import { Card, CardHeader, CardContent, Typography, Grid, Divider } from '@material-ui/core';

import { useSpeechContext } from '@speechly/react-client';

import { ExpenseTrackerContext } from '../../context/context';

import useStyles from './styles';

import Form from './Form/Form';

import List from './List/List';

import InfoCard from '../InfoCard';

const ExpenseTracker = () => {

const classes = useStyles();

const { balance } = useContext(ExpenseTrackerContext);

return (

<Card className={classes.root}>

<CardHeader title="Expense Tracker" subheader="Powered by Speechly" />

<CardContent>

<Typography align="center" variant="h5">Total Balance ${balance}</Typography>

<Typography variant="subtitle1" style={{ lineHeight: '1.5em', marginTop: '20px' }}>

<InfoCard />

</Typography>

<Divider className={classes.divider} />

<Form />

</CardContent>

<CardContent className={classes.cartContent}>

<Grid container spacing={2}>

<Grid item xs={12}>

<List />

</Grid>

</Grid>

</CardContent>

</Card>

);

};

export default ExpenseTracker;

//inside main

// create infor card, form, list

1. FORM COMPONENT

import React, { useState, useContext, useEffect } from 'react';

import { TextField, Typography, Grid, Button, FormControl, InputLabel, Select, MenuItem } from '@material-ui/core';

import { v4 as uuidv4 } from 'uuid';

import { useSpeechContext } from '@speechly/react-client';

import Snackbar from '../../Snackbar/Snackbar';

import formatDate from '../../../utils/formatDate';

import { ExpenseTrackerContext } from '../../../context/context';

import { incomeCategories, expenseCategories } from '../../../constants/categories';

import useStyles from './styles';

// type, catagory ,(sec) amount ,date

const initialState = {

amount: '',

category: '',

type: 'Income',

date: formatDate(new Date()),

};

const NewTransactionForm = () => {

const classes = useStyles();

const { addTransaction } = useContext(ExpenseTrackerContext);

const [formData, setFormData] = useState(initialState);

const { segment } = useSpeechContext();//segement of voice

const [open, setOpen] = React.useState(false);

const createTransaction = () => {

//corner case

if (Number.isNaN(Number(formData.amount)) || !formData.date.includes('-')) return;

if (incomeCategories.map((iC) => iC.type).includes(formData.category)) {

setFormData({ ...formData, type: 'Income' });

} else if (expenseCategories.map((iC) => iC.type).includes(formData.category)) {

setFormData({ ...formData, type: 'Expense' });

}

//snackbar by default shows

setOpen(true);

addTransaction({ ...formData, amount: Number(formData.amount), id: uuidv4() });

setFormData(initialState);

};

// take inpput using speachly

useEffect(() => {

if (segment) {

if (segment.intent.intent === 'add\_expense') {

setFormData({ ...formData, type: 'Expense' });// if we say add expense change type to expense

} else if (segment.intent.intent === 'add\_income') {

setFormData({ ...formData, type: 'Income' });

} else if (segment.isFinal && segment.intent.intent === 'create\_transaction') {

return createTransaction();

} else if (segment.isFinal && segment.intent.intent === 'cancel\_transaction') {

return setFormData(initialState);

}

segment.entities.forEach((s) => {

// change only 1 word to upper case and rest lower case

const category = `${s.value.charAt(0)}${s.value.slice(1).toLowerCase()}`;

switch (s.type) {

case 'amount':

setFormData({ ...formData, amount: s.value });

break;

case 'category':

if (incomeCategories.map((iC) => iC.type).includes(category)) {

setFormData({ ...formData, type: 'Income', category });

} else if (expenseCategories.map((iC) => iC.type).includes(category)) {

setFormData({ ...formData, type: 'Expense', category });

}

break;

case 'date':

setFormData({ ...formData, date: s.value });

break;

default:

break;

}

});

// to automatically create tans if data is filled

if (segment.isFinal && formData.amount && formData.category && formData.type && formData.date) {

createTransaction();

}

}

}, [segment]);

const selectedCategories = formData.type === 'Income' ? incomeCategories : expenseCategories;

//shows the list of catagories

return (

<Grid container spacing={2}>

{/\* snackbar \*/}

<Snackbar open={open} setOpen={setOpen} />

<Grid item xs={12}>

<Typography align="center" variant="subtitle2" gutterBottom>

{/\* see what words we are speaking \*/}

{segment ? (

<div className="segment">

{segment.words.map((w) => w.value).join(" ")}

</div>

) : null}

{/\* {isSpeaking ? <BigTranscript /> : 'Start adding transactions'} \*/}

</Typography>

</Grid>

<Grid item xs={6}>

<FormControl fullWidth>

{/\* type \*/}

<InputLabel>Type</InputLabel>

<Select value={formData.type} onChange={(e) => setFormData({ ...formData, type: e.target.value })}>

<MenuItem value="Income">Income</MenuItem>

<MenuItem value="Expense">Expense</MenuItem>

</Select>

</FormControl>

</Grid>

<Grid item xs={6}>

<FormControl fullWidth>

{/\* catagory \*/}

<InputLabel>Category</InputLabel>

<Select value={formData.category} onChange={(e) => setFormData({ ...formData, category: e.target.value })}>

{/\* render catagories dynamically \*/}

{selectedCategories.map((c) => <MenuItem key={c.type} value={c.type}>{c.type}</MenuItem>)}

</Select>

</FormControl>

</Grid>

<Grid item xs={6}>

{/\* amount \*/}

<TextField type="number" label="Amount" value={formData.amount} onChange={(e) => setFormData({ ...formData, amount: e.target.value })} fullWidth />

</Grid>

<Grid item xs={6}>

{/\* date \*/}

<TextField fullWidth label="Date" type="date" value={formData.date} onChange={(e) => setFormData({ ...formData, date: formatDate(e.target.value) })} />

</Grid>

{/\* create button \*/}

<Button className={classes.button} variant="outlined" color="primary" fullWidth onClick={createTransaction}>Create</Button>

</Grid>

);

};

export default NewTransactionForm;

1. LIST COMPONENT

import React, { useContext } from 'react';

import { List as MUIList, ListItem, ListItemAvatar, Avatar, ListItemText, ListItemSecondaryAction, IconButton, Slide } from '@material-ui/core';

import { Delete, MoneyOff } from '@material-ui/icons';

import { ExpenseTrackerContext } from '../../../context/context';

import useStyles from './styles';

const List = () => {

const classes = useStyles();

//transactions is being fetched up from context and displaied in list

const { transactions, deleteTransaction } = useContext(ExpenseTrackerContext);

//using context :expensetracker

return (

<MUIList dense={false} className={classes.list}>

{transactions.map((transaction) => (

<Slide direction="down" in mountOnEnter unmountOnExit key={transaction.id}>

<ListItem>

<ListItemAvatar>

{/\* icon \*/}

<Avatar className={transaction.type === 'Income' ? classes.avatarIncome : classes.avatarExpense}>

<MoneyOff />

</Avatar>

</ListItemAvatar>

{/\* primary will be catagory name, secondary is for amount and date \*/}

<ListItemText primary={transaction.category} secondary={`$${transaction.amount} - ${transaction.date}`} />

<ListItemSecondaryAction>

{/\* delete buttom \*/}

<IconButton edge="end" aria-label="delete" onClick={() => deleteTransaction(transaction.id)}>

<Delete />

</IconButton>

</ListItemSecondaryAction>

</ListItem>

</Slide>

))}

</MUIList>

);

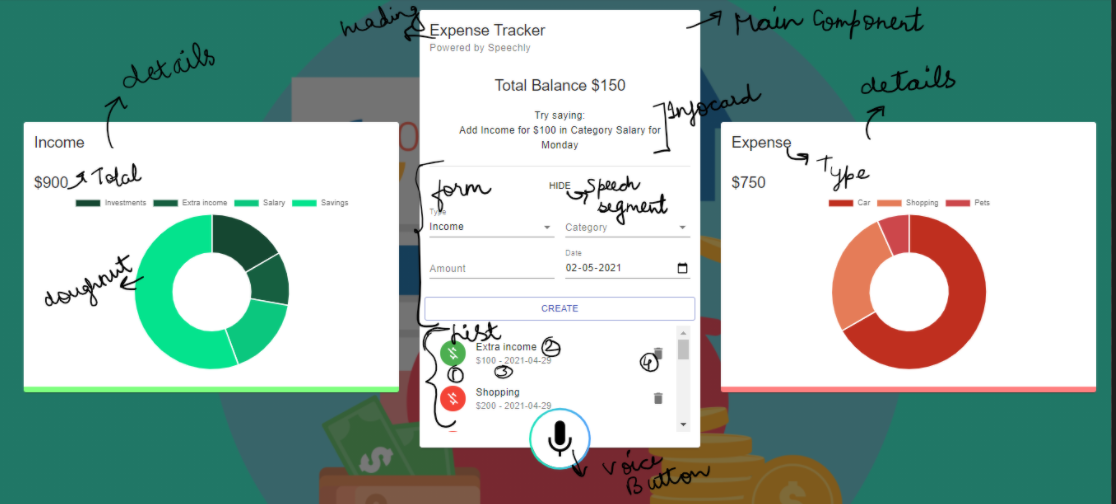
};

export default List;

// list to show transactions:which is an array

// trans: id , type, catagory ,(sec) amount ,date

**Result of Module 2**



# 

# Code of Module 3

**THIS MODULE SHOWS THE CODE FOR THE BACKEND DEVELOPMENT**

1. CONTEXT USED FOR LOCAL STORAGE

import React, { useReducer, createContext } from 'react';

import contextReducer from './contextReducer';

// use locally stored data or if not give this

const initialState = JSON.parse(localStorage.getItem('transactions')) || [{ amount: 500, category: 'Salary', type: 'Income', date: '2020-11-16', id: '44c68123-5b86-4cc8-b915-bb9e16cebe6a' }, { amount: 225, category: 'Investments', type: 'Income', date: '2020-11-16', id: '33b295b8-a8cb-49f0-8f0d-bb268686de1a' }, { amount: 50, category: 'Salary', type: 'Income', date: '2020-11-13', id: '270304a8-b11d-4e16-9341-33df641ede64' }, { amount: 123, category: 'Car', type: 'Expense', date: '2020-11-16', id: '0f72e66e-e144-4a72-bbc1-c3c92018635e' }, { amount: 50, category: 'Pets', type: 'Expense', date: '2020-11-13', id: 'c5647dde-d857-463d-8b4e-1c866cc5f83e' }, { amount: 500, category: 'Travel', type: 'Expense', date: '2020-11-13', id: '365a4ebd-9892-4471-ad55-36077e4121a9' }, { amount: 50, category: 'Investments', type: 'Income', date: '2020-11-23', id: '80cf7e33-fc3e-4f9f-a2aa-ecf140711460' }, { amount: 500, category: 'Savings', type: 'Income', date: '2020-11-23', id: 'ef090181-21d1-4568-85c4-5646232085b2' }, { amount: 5, category: 'Savings', type: 'Income', date: '2020-11-23', id: '037a35a3-40ec-4212-abe0-cc485a98aeee' }];

export const ExpenseTrackerContext = createContext(initialState);

export const Provider = ({ children }) => {

const [transactions, dispatch] = useReducer(contextReducer, initialState);

//reducer is a fxn that defines how we will change our state so we create contextreducer

//action creator

const deleteTransaction = (id) => {

dispatch({ type: 'DELETE\_TRANSACTION', payload: id });

};

const addTransaction = (transaction) => {

dispatch({ type: 'ADD\_TRANSACTION', payload: transaction });

};

// we will call the functions somewhere else(form) but return them in ExpenseTrackerContext.provider for universal excess

const balance = transactions.reduce((acc, currVal) => (currVal.type === 'Expense' ? acc - currVal.amount : acc + currVal.amount), 0);

// to update balance

return (

<ExpenseTrackerContext.Provider value={{

transactions,

balance,

deleteTransaction,

addTransaction,

}}

>

{children}

</ExpenseTrackerContext.Provider>

);

};

// context can be used in place of redux ,wrap our application with provider and all

//of our components will have excess to -- value property whatever is inside it

// call provider inside index.js and wrap app div inside it.

1. CONTEXT REDUCER

const contextReducer = (state, action) => {

let transactions;

switch (action.type) {

case 'DELETE\_TRANSACTION':

transactions = state.filter((transaction) => transaction.id !== action.payload);

//storing locally

localStorage.setItem('transactions', JSON.stringify(transactions));

return transactions;

case 'ADD\_TRANSACTION':

transactions = [action.payload, ...state];

// add data and keep rest same,...state is to spread data

localStorage.setItem('transactions', JSON.stringify(transactions));

return transactions;

default:

return state;

}

};

export default contextReducer;

// for backend => Localstorage api :localStorage stores key-value pairs. So to store a entire javascript object we need to serialize it first

//Then to retrieve it from the store and convert to an object again

//using hooks to mainatain state with usestate ,benefits if using hooks is helps in sharing stateful logic without rendering props .

//using context api to share data among components

//consuming context data with usecontext hook.

1. FETCHING PIE DATA

import { useContext } from 'react';

import { ExpenseTrackerContext } from './context/context';

// to add data in pie chart

import { incomeCategories, expenseCategories, resetCategories } from './constants/categories';

const useTransactions = (title) => {

resetCategories();

const { transactions } = useContext(ExpenseTrackerContext);//using context to acess

const rightTransactions = transactions.filter((t) => t.type === title);

const total = rightTransactions.reduce((acc, currVal) => acc += currVal.amount, 0);

const categories = title === 'Income' ? incomeCategories : expenseCategories;

rightTransactions.forEach((t) => {

const category = categories.find((c) => c.type === t.category);

if (category) category.amount += t.amount;

});

//filter catagory haveing 0 amount

const filteredCategories = categories.filter((sc) => sc.amount > 0);

const chartData = {

datasets: [{

data: filteredCategories.map((c) => c.amount),

backgroundColor: filteredCategories.map((c) => c.color),

}],

labels: filteredCategories.map((c) => c.type),

};

return { filteredCategories, total, chartData };

};

export default useTransactions;

1. CATEGORIES

const incomeColors = ['#123123', '#154731', '#165f40', '#16784f', '#14915f', '#10ac6e', '#0bc77e', '#04e38d', '#00ff9d'];

const expenseColors = ['#b50d12', '#bf2f1f', '#c9452c', '#d3583a', '#dc6a48', '#e57c58', '#ee8d68', '#f79d79', '#ffae8a', '#cc474b', '#f55b5f'];

export const incomeCategories = [

{ type: 'Business', amount: 0, color: incomeColors[0] },

{ type: 'Investments', amount: 0, color: incomeColors[1] },

{ type: 'Extra income', amount: 0, color: incomeColors[2] },

{ type: 'Deposits', amount: 0, color: incomeColors[3] },

{ type: 'Lottery', amount: 0, color: incomeColors[4] },

{ type: 'Gifts', amount: 0, color: incomeColors[5] },

{ type: 'Salary', amount: 0, color: incomeColors[6] },

{ type: 'Savings', amount: 0, color: incomeColors[7] },

{ type: 'Rental income', amount: 0, color: incomeColors[8] },

];

export const expenseCategories = [

{ type: 'Bills', amount: 0, color: expenseColors[0] },

{ type: 'Car', amount: 0, color: expenseColors[1] },

{ type: 'Clothes', amount: 0, color: expenseColors[2] },

{ type: 'Travel', amount: 0, color: expenseColors[3] },

{ type: 'Food', amount: 0, color: expenseColors[4] },

{ type: 'Shopping', amount: 0, color: expenseColors[5] },

{ type: 'House', amount: 0, color: expenseColors[6] },

{ type: 'Entertainment', amount: 0, color: expenseColors[7] },

{ type: 'Phone', amount: 0, color: expenseColors[8] },

{ type: 'Pets', amount: 0, color: expenseColors[9] },

{ type: 'Other', amount: 0, color: expenseColors[10] },

];

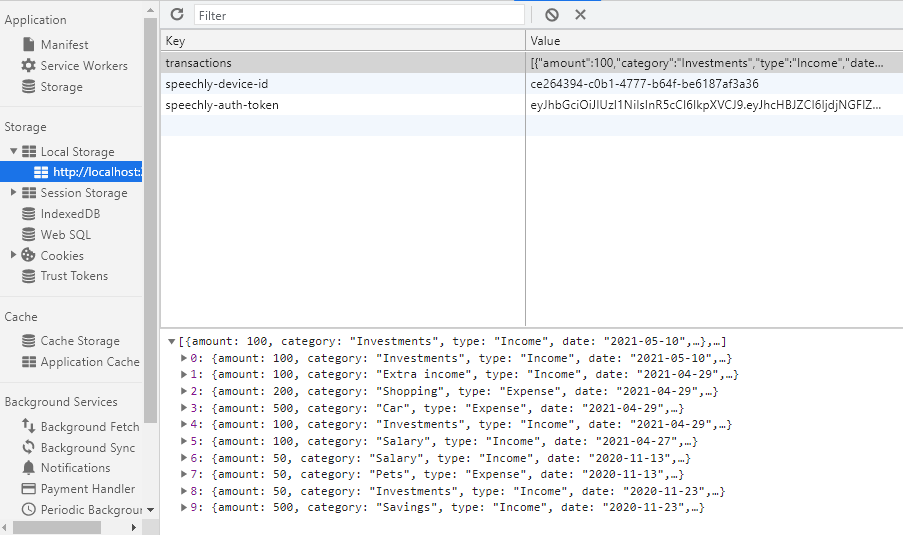
export const resetCategories = () => {

incomeCategories.forEach((c) => c.amount = 0);

expenseCategories.forEach((c) => c.amount = 0);

};

**Result of Module 3**



# TEST PLAN ,TEST CASE

# Executive Summary

The testing process is done to find defects which may get created by the programmer while developing the software and to make sure that the end result meets the business and user requirements.

To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications and also to gain the confidence of the customers by providing them a quality product.

# Test Plan

To prepare a master test plan and conduct manual test using test cases then prepare test report for the project

# Scope of Testing

**Functional:** All the Functional requirements are covered

**Non-Functional:** All the Non-Functional requirements are covered

# Types of Testing , Methodology , Tools

|  |  |  |
| --- | --- | --- |
| Category | Methodology | Tools Required |
| Functional Requirements | Manual | Word Template |

# 

# Test Case

# Functional Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID (#)** | **Test Scenario** | **Test Case** | **Execution Steps** | **Expected Outcome** | **Actual Outcome** | **Status** | **Remarks** |
|  | Taking the input values | Accept input | 1. Take input in different catagories 2. Assign input 3. Match in database | User views the message if successful | User views the message if successful | pass | success |
|  | Taking voice input | Check for voice actions | Taking input and assigining to different catagories | If wrong it will ask again with some sample text | If wrong it will ask again with some sample text | pass | success |
|  | Adding new entry | Accept the new data | After clicking create button entry is made  Adding new entry into database | Message displayed as  Try this” “for new entry | Message displayed as  Try this” “for new entry  Entry made successfully | pass | success |
|  | Enter valid values | Accepts valid values | 1) enter different paramenters  2) Checking in database | Message displayed successfully added | Message displayed as successfully added | pass | success |

# Non-Functional Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID (#)** | **Test Scenario** | **Test Case** | **Status** | **Remarks** |
|  | Performance testing | Adding values and displaying should not excess more than 5 sec | Pass | Completed |
|  | Volume testing | Analyzing the system performance by increasing the volume of data in database | Pass | Completed |
|  | Usability testing | To identify any usability problems, collect qualitative and quantitative data and determine the participant's satisfaction with the product. | Pass | Completed |
|  | Reliability testing | To check whether the software can perform a failure-free operation for a specified time period in a particular environment. | Pass | Completed |

**Manual testing with report**

# Executive Summary

This program describes the Expense Tracker system. Testing provides a system that uses a variety of test results. Testing helps in predicting the any future errors or the debugs in the program. Then the errors are detected and corrected using test steps and corrections are made for future reference. Thus, a sequence of checks is carried out on the software before being applied.

# Test Plan

# Scope of Testing

The aim of the system testing process was to determine all defects in our project.

**Functional:** All the Functional requirements are covered

**Non-Functional:** All the Non-Functional requirements are covered

# Types of Testing , Methodology , Tools

|  |  |  |
| --- | --- | --- |
| Category | Methodology | Tools Required |
| Functional Requirements | Manual | Word Template |

# Test Deliverables

<< Test Case Documentation, Defect Log, Test Report >>

# Test Case

# Functional Test Cases

Functional testing is a type of testing which verifies that each function of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing, and it is not concerned about the source code of the application.

Every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results. This testing involves checking of User Interface, APIs, Database, security, client/ server applications and functionality of the Application Under Test. The testing can be done either manually or using automation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID (#) | Test Scenario | Test Case | Execution Steps | Expected Outcome | Actual Outcome | Status | Remarks |
|  | Taking the input values | Accept input | 1. Take input in different catagories  2. Assign input  3. Match in database | User views the message if successful | User views the message if successful | pass | success |
|  | Taking voice input | Check for voice actions | Taking input and assigining to different catagories | If wrong it will ask again with some sample text | If wrong it will ask again with some sample text | pass | success |
|  | Adding new entry | Accept the new data | After clicking create button entry is made    Adding new entry into database | Message displayed as  Try this” “for new entry | Message displayed as  Try this” “for new entry  Entry made successfully | pass | success |
|  | Enter valid values | Accepts valid values | 1) enter different paramenters  2) Checking in database | Message displayed successfully added | Message displayed as successfully added | pass | success |

# 

# Non-Functional Test Cases:

Non-functional testing is a type of testing to check non-functional aspects (performance, usability, reliability, etc.) of a software application. It is explicitly designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID (#) | Test Scenario | Test Case | Status | Remarks |
|  | Performance testing | Adding values and displaying should not excess more than 5 sec | Pass | Completed |
|  | Volume testing | Analyzing the system performance by increasing the volume of data in database | Pass | Completed |
|  | Usability testing | To identify any usability problems, collect qualitative and quantitative data and determine the participant's satisfaction with the product. | Pass | Completed |
|  | Reliability testing | To check whether the software can perform a failure-free operation for a specified time period in a particular environment. | Pass | Completed |

**Defect Log:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement # | Defect ID # | Defect Description | Assignee | Status |
| M1R1 | RA1 | If the person once speak and enter the data and something goes wrong it wont do anything ,unless we press the voice button to speak again | Priyansha | completed |

# Test Report:

|  |  |  |
| --- | --- | --- |
| **Category** | **Progress Against Plan** | **Status** |
| Functional Testing | Green | Completed |
| Non-Functional Testing | Green | Completed |

|  |  |  |
| --- | --- | --- |
| **Functional** | **Test Case Coverage (%)** | **Status** |
| M1R1 | 50% | Completed |
| M1R2 | 50% | Completed |
| M1R3 | 50% | Completed |