1. Title: Expense Management System

2.Project Statement:

Expense management software can guarantee efficiency and transparency while assisting companies in adhering to corporate standards and tax laws. Additionally, it can save time by eliminating the need for manual expense monitoring, giving staff members more time to participate in reporting, budget planning, and spending analysis.

Outcomes:

The Expense Management System is designed to help organizations track and manage their expenses efficiently. The system is developed using a Monolithic architecture to ensure scalability, flexibility, and maintainability. The application is divided into multiple services, each responsible for a specific aspect of expense management.

3. Modules to be implemented

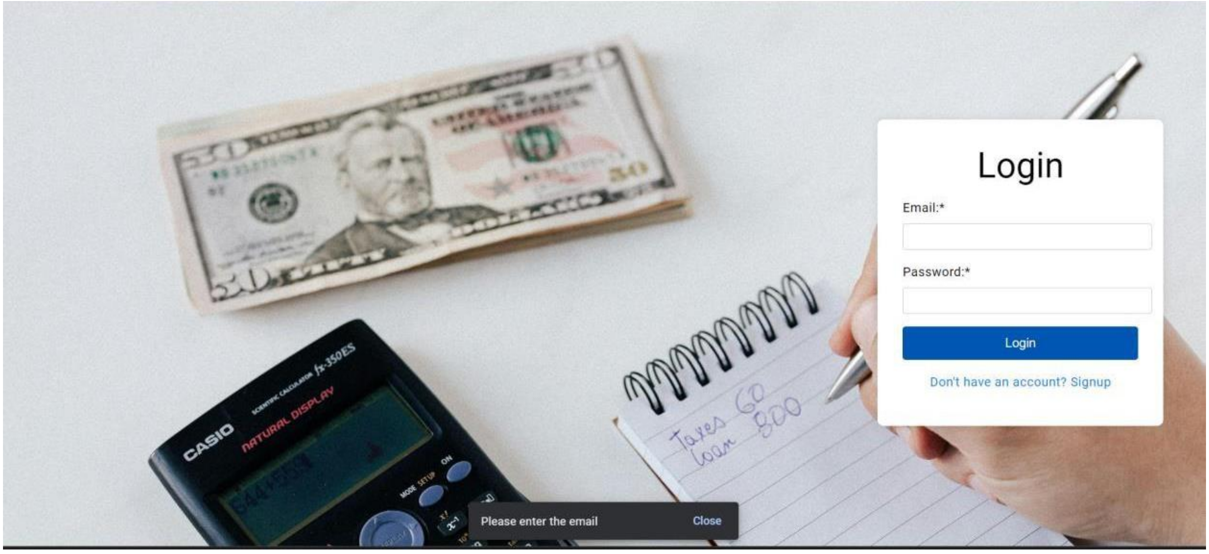
1. User Module
2. Expense Module
3. Category Module
4. Report Module
5. Notification Module
6. Feedback Module

4. **Week-wise module implementation and high-level requirements with output screenshots**

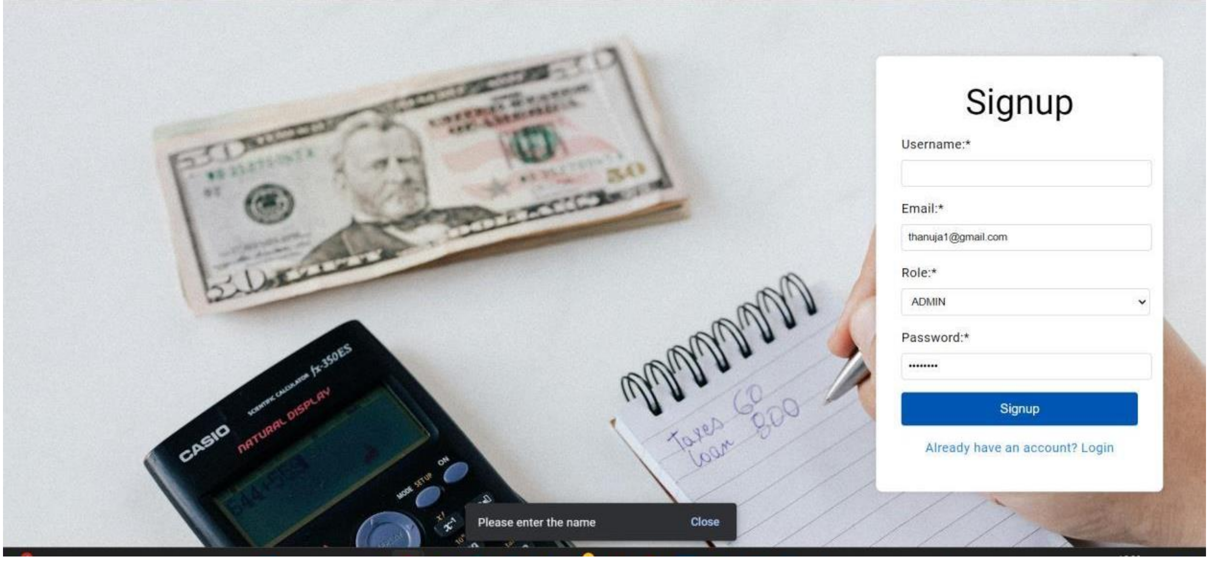
Milestone 1: Weeks 1-3

Module 1: User Authentication and Registration

- Implement user registration functionality.
- Develop user login mechanism.
- Spring security mechanism for redirecting to the respective dashboard. Ex: admin, student, and parent dashboards separately.
- After login successfully display the profile section with picture.
- Integrate email verification and password reset features.
- **Output screenshot:**



Please enter the email



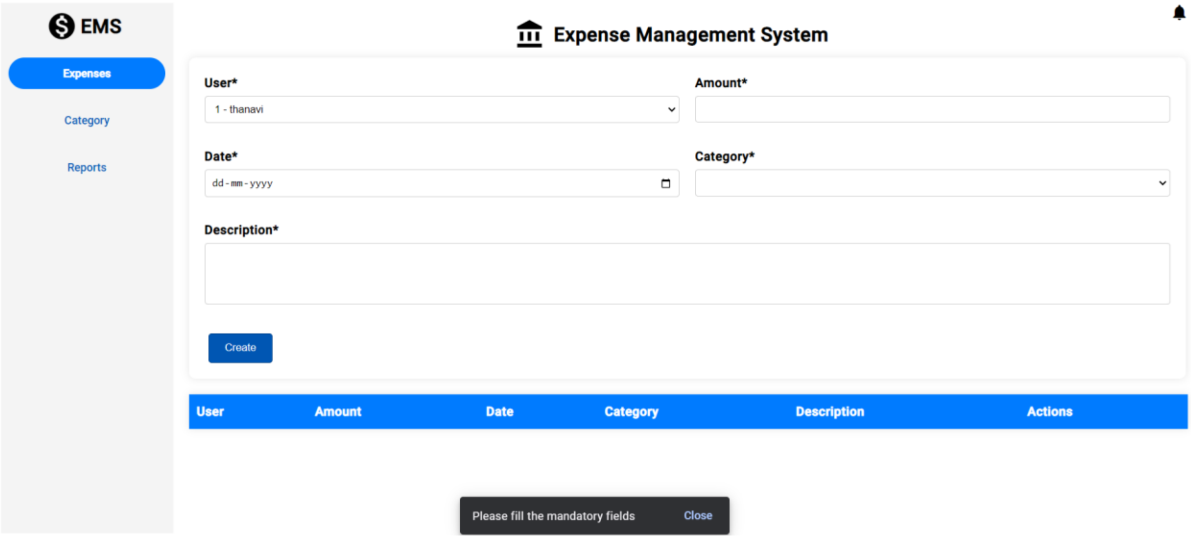
Email:* thanuja1@gmail.com

Role:* ADMIN

Password:* ••••••••

Please enter the name

- **Module 2: Expense Management:**
- Design database schema for storing expense information.
- Develop an interface for expense to add the details.
- Implement backend functionality to manage and update expense information.
- Write test cases using JUnit and Mockito for all the Services with code coverage.
- Implement respective module frontend using Angular/React JS and write test cases for frontend logic.
- Form validations must be implemented for all the services.
- **Output screenshot:**

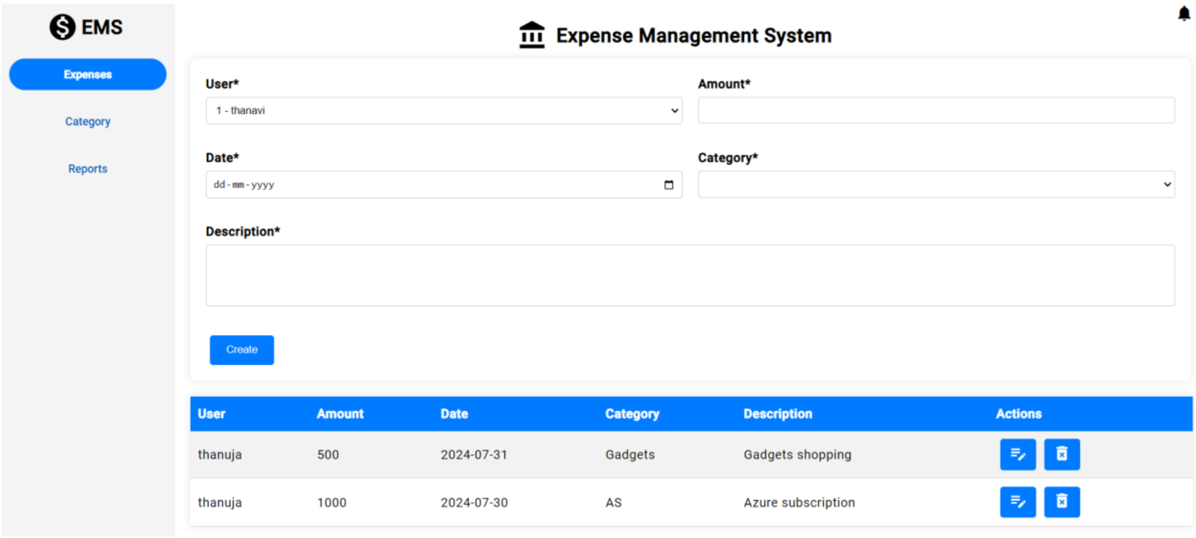


User* 1 - thanavi

dd-mm-yyyy

User	Amount	Date	Category	Description	Actions

Please fill the mandatory fields



User* 1 - thanavi

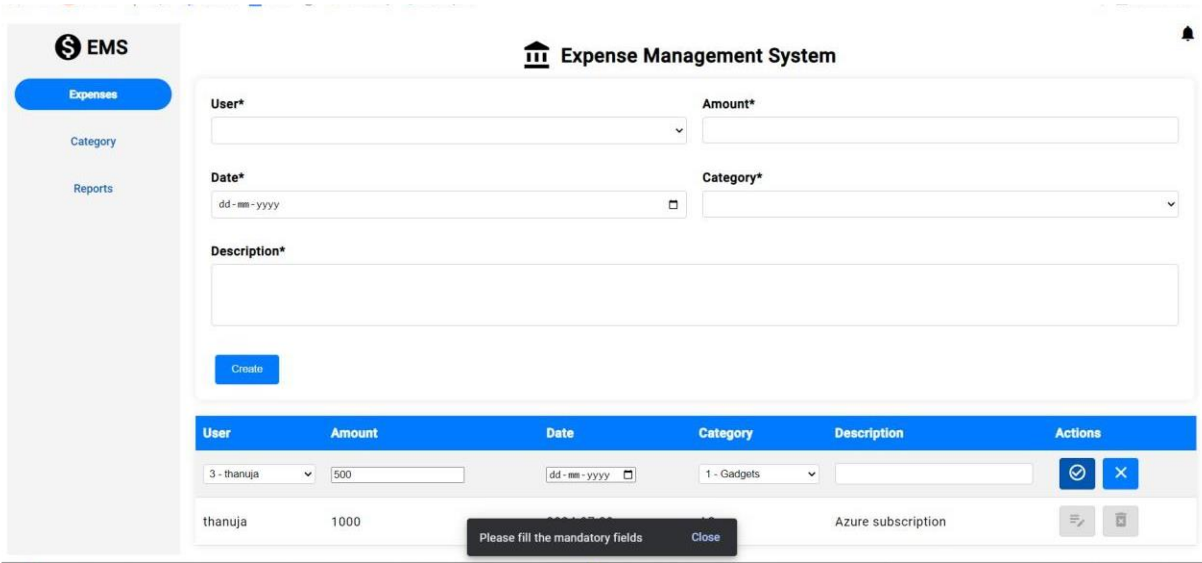
Amount*

Date* dd-mm-yyyy

Category*

Description*

User	Amount	Date	Category	Description	Actions
thanuja	500	2024-07-31	Gadgets	Gadgets shopping	
thanuja	1000	2024-07-30	AS	Azure subscription	



User*

Amount*

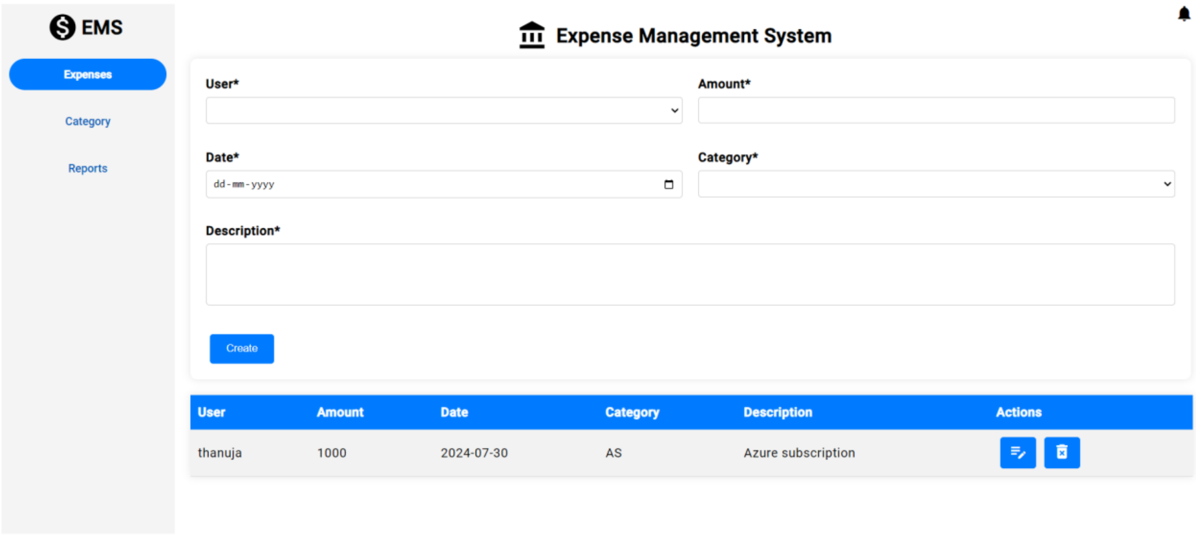
Date* dd-mm-yyyy

Category*

Description*

User	Amount	Date	Category	Description	Actions
3 - thanuja	500	dd-mm-yyyy	1 - Gadgets		
thanuja	1000			Azure subscription	

Please fill the mandatory fields



User*

Amount*

Date* dd-mm-yyyy

Category*

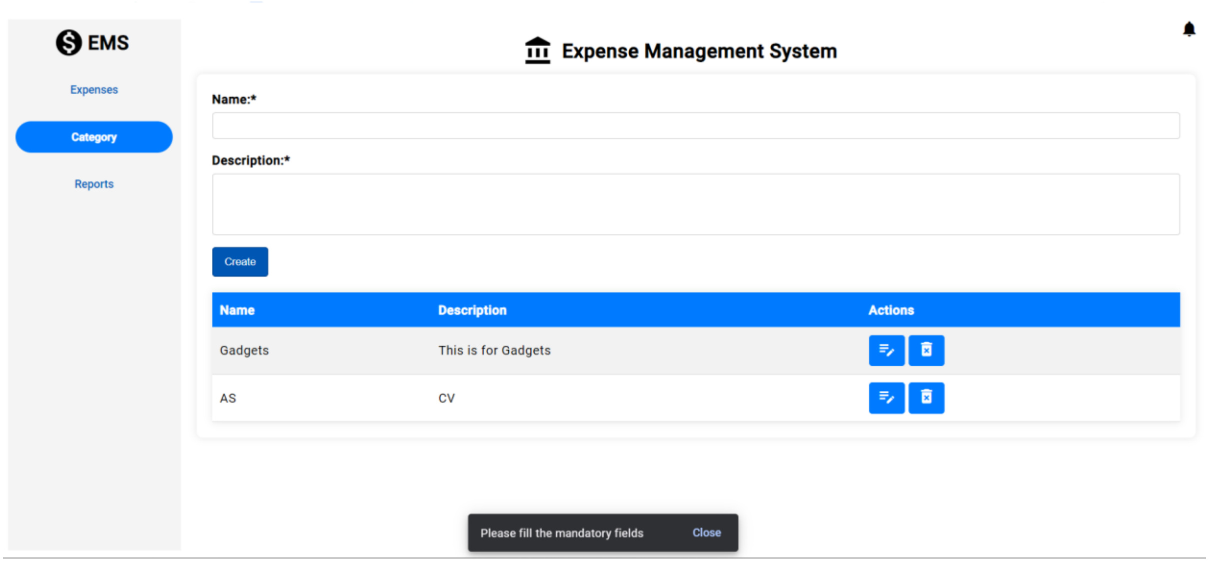
Description*

User	Amount	Date	Category	Description	Actions
thanuja	1000	2024-07-30	AS	Azure subscription	

Milestone 2: Weeks 4-5

Module 3: Category Module

- Design and develop the interface for Categories to display.
- Implement functionality for users to select Categories and provide the category wise expense details.
- Integrate AI tools with Spring boot to generate content for expense and display the highest category on top to the application.
- After selecting the category, it will accept all input data to it.
- Once the customer/user adds expenses, display the expense data in the category section and expense data.
- Integrate the expense interface with the backend for storing category wise expense details.
- In this module, we need to provide a dashboard for respective roles: Ex-admin, customer role with respective dashboards.
- The admin dashboard displays all services that are accessible for CRUD operations. we can search for specific candidates to check the status of the customer, and the admin replies with remarks to the customer for expense data.
- Write test cases using JUnit and Mockito for all the Services with code coverage.
- In dashboard display booking status report in graphical representation for finance and count for shows with percentage.
- Implement respective module frontend using Angular/React JS and write test cases for frontend logic.
- Form validations must be implemented for all the services.
- **Output screenshot:**



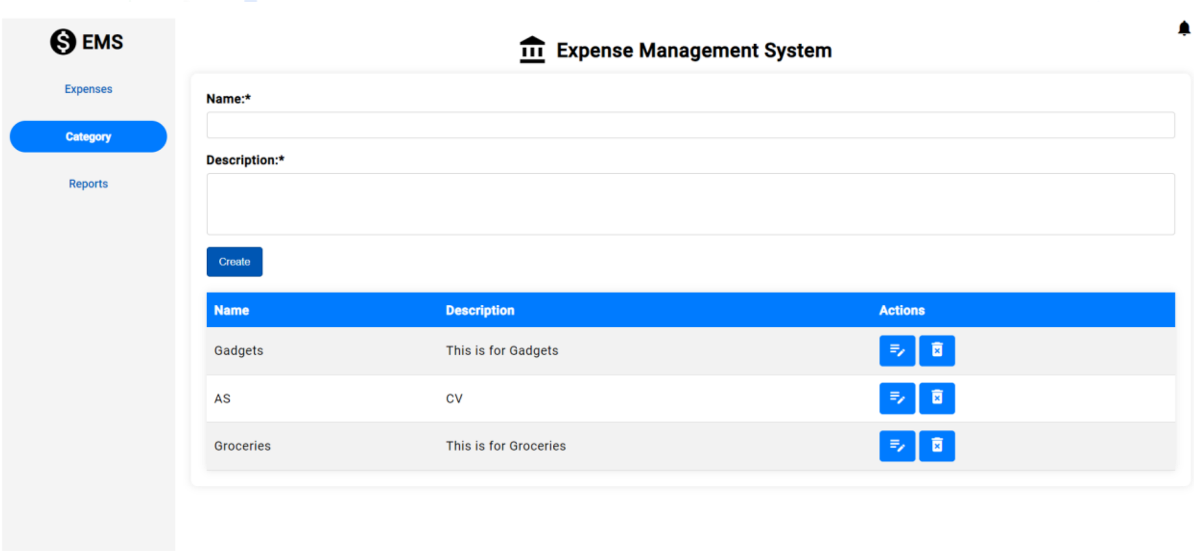
Expense Management System

Name:*

Description:*

Name	Description	Actions
Gadgets	This is for Gadgets	
AS	CV	

Please fill the mandatory fields

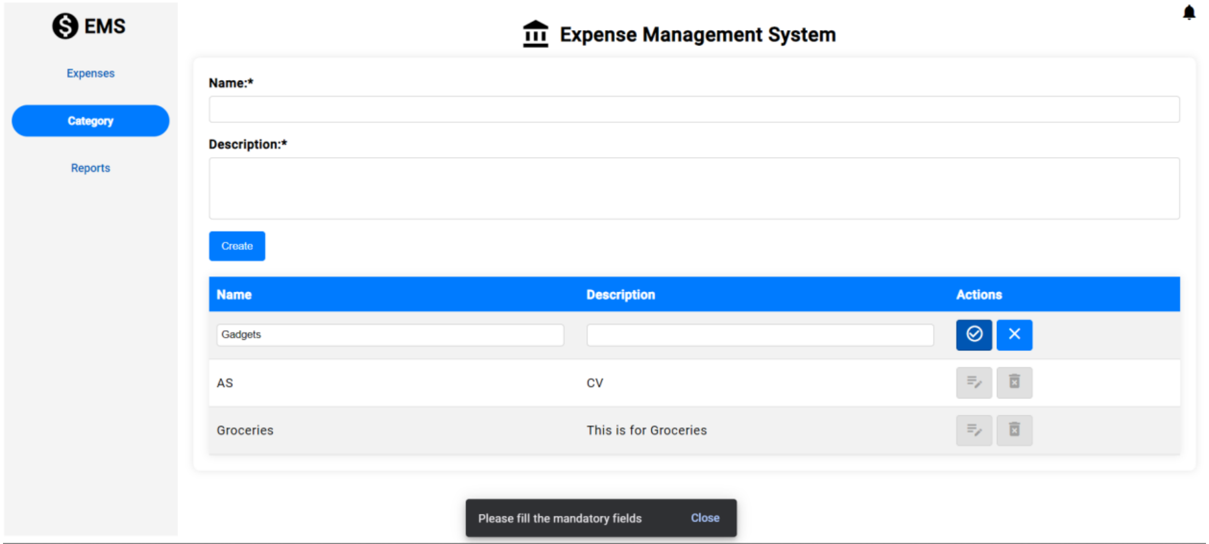


Expense Management System

Name:*

Description:*

Name	Description	Actions
Gadgets	This is for Gadgets	
AS	CV	
Groceries	This is for Groceries	



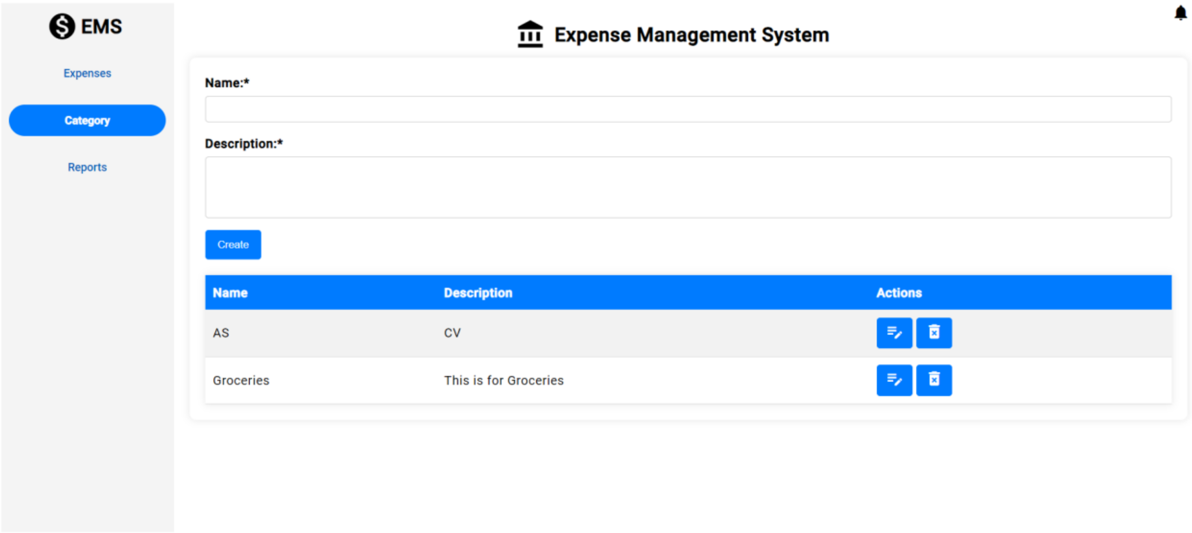
Expense Management System

Name:*

Description:*

Name	Description	Actions
Gadgets		
AS	CV	
Groceries	This is for Groceries	

Please fill the mandatory fields



Name:*

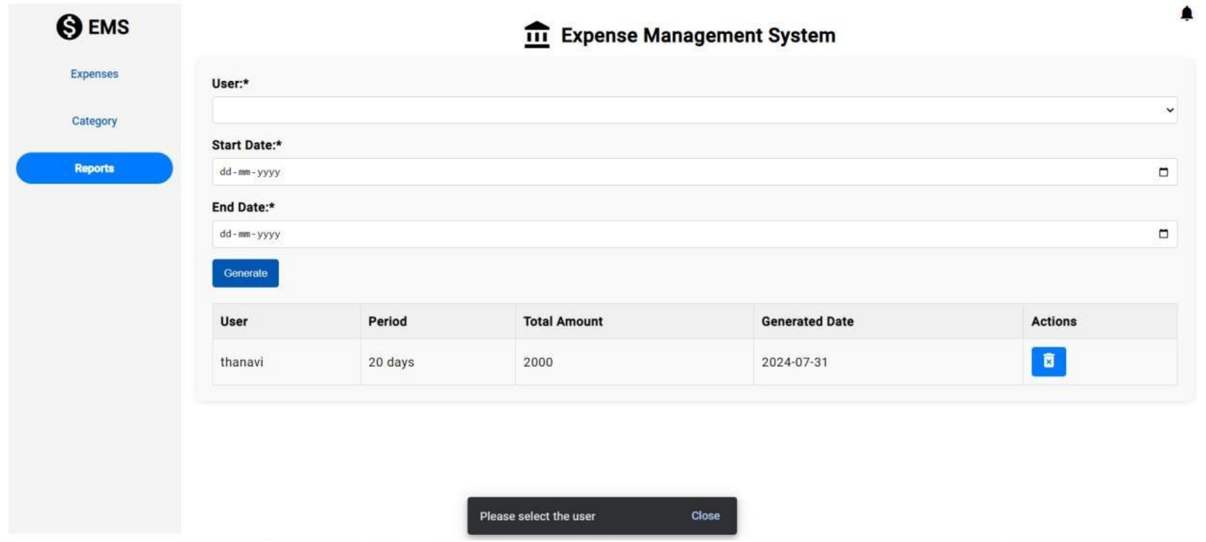
Description:*

Name	Description	Actions
AS	CV	
Groceries	This is for Groceries	

Milestone 3: Weeks 6-7

Module 4: Report Module

- Design database schema for storing report details.
- Implement backend functionality for validating the report information.
- Develop an interface for users and admin to view and manage reports.
- For report information automatically send alerts to respective mail or notifications in profile section with alert symbol.
- Develop a code for displaying report policy before going to modify expense data.
- Write test cases using JUnit and Mockito for all the Services with code coverage.
- Implement respective module frontend using Angular/React JS and write test cases for frontend logic.
- Form validations must be implemented for all the services.
- **Output Screenshot:**



Expense Management System

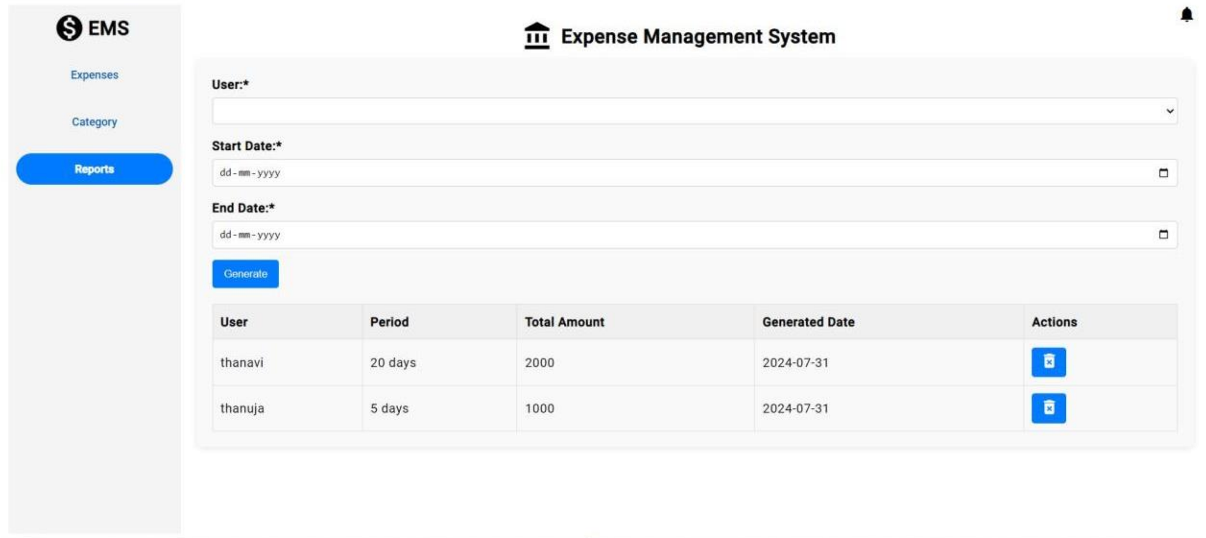
User:*

Start Date:* dd-mm-yyyy

End Date:* dd-mm-yyyy

User	Period	Total Amount	Generated Date	Actions
thanavi	20 days	2000	2024-07-31	

Please select the user



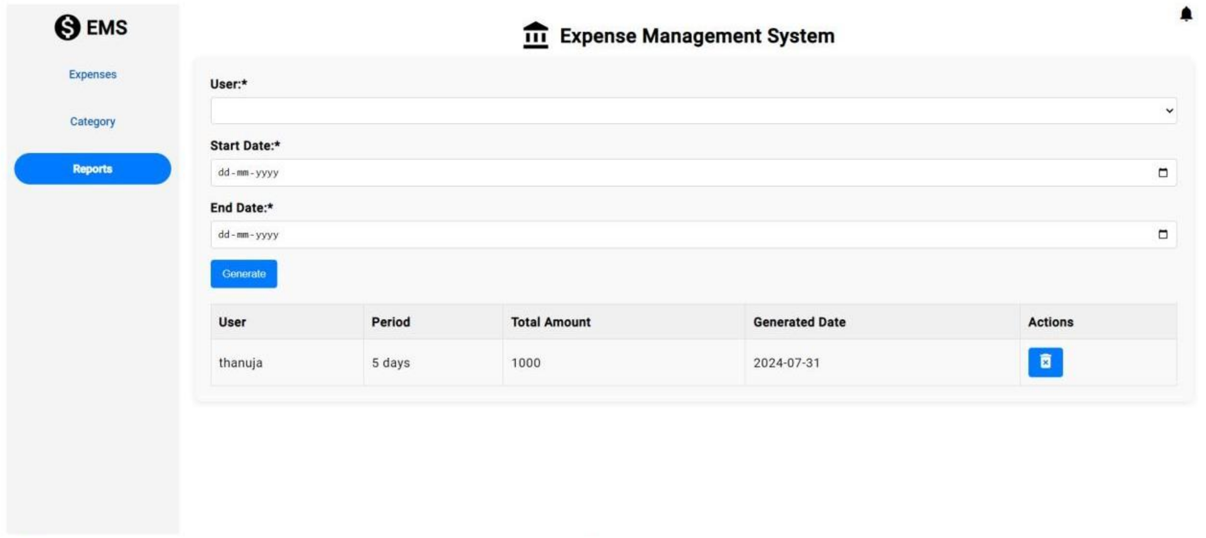
Expense Management System

User:*

Start Date:* dd-mm-yyyy

End Date:* dd-mm-yyyy

User	Period	Total Amount	Generated Date	Actions
thanavi	20 days	2000	2024-07-31	
thanuja	5 days	1000	2024-07-31	



Expense Management System

User:*

Start Date:* dd-mm-yyyy

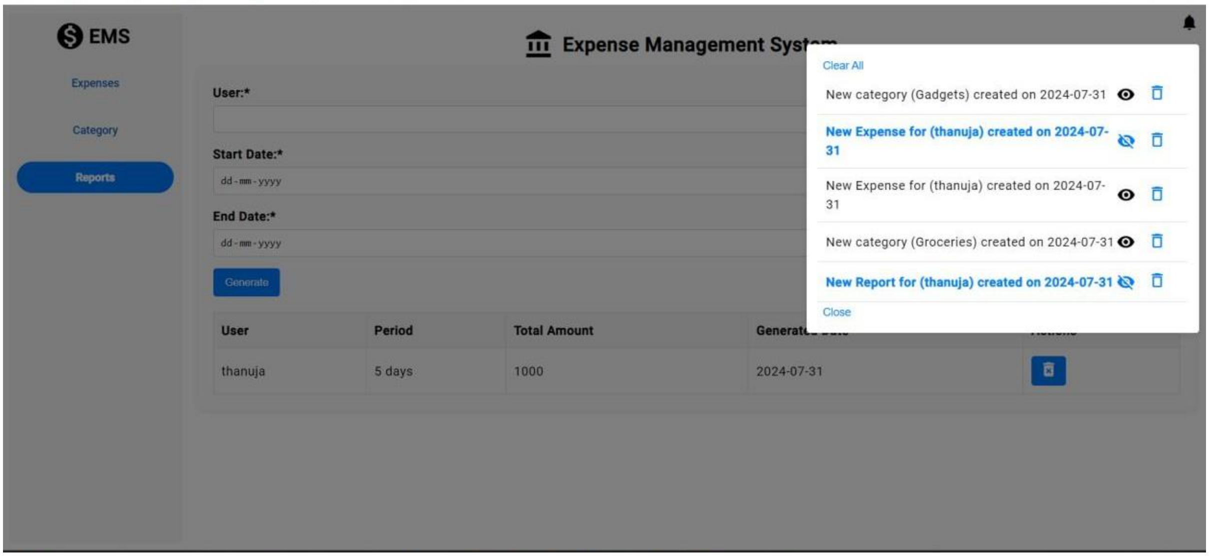
End Date:* dd-mm-yyyy

User	Period	Total Amount	Generated Date	Actions
thanuja	5 days	1000	2024-07-31	

Milestone 4: Weeks 8-10

Module 5: In App Notification Module

- Design database schema for storing expense and category information securely.
- Develop functionality for users to provide and update their expense information.
- Implement backend logic for accessing and managing notification information securely.
- Users can manage their notifications either by clicking or removing the notification from the dashboard.
- Admin and user manage notifications in this module.
- Admin can remove User data once the user notification is deleted.
- The removed users can’t watch notifications from the dashboard.
- Write test cases using JUnit and Mockito for all the Services with code coverage.
- Implement respective module frontend using Angular/React JS and write test cases for frontend logic.
- Form validations must be implemented for all the services.
- **Output Screenshot:**



User:*

Start Date:* dd-mm-yyyy

End Date:* dd-mm-yyyy

User	Period	Total Amount	Generat	
thanuja	5 days	1000	2024-07-31	

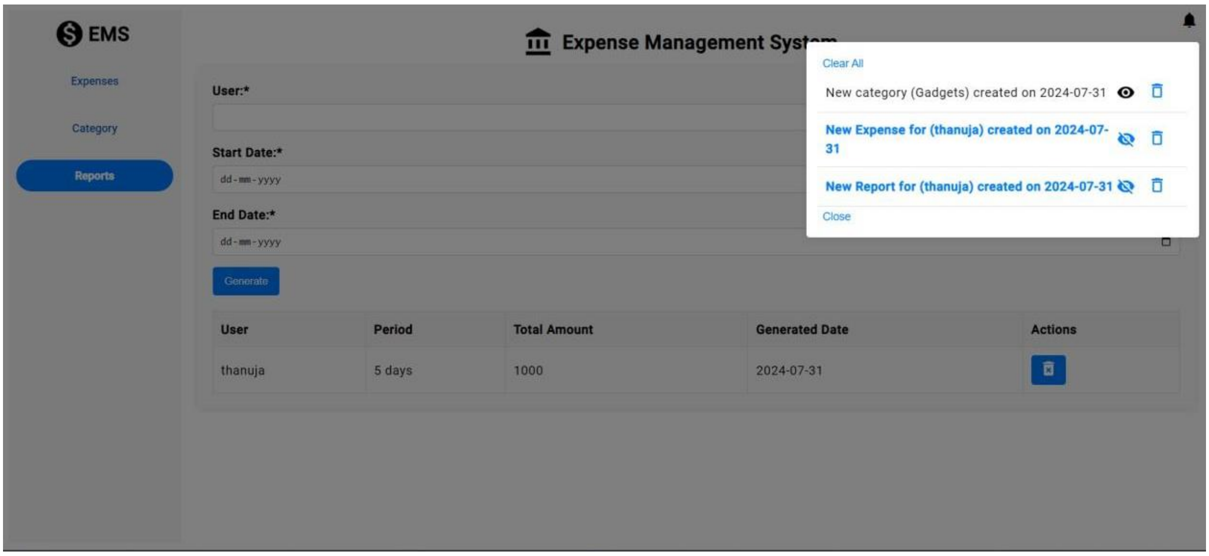
New category (Gadgets) created on 2024-07-31

New Expense for (thanuja) created on 2024-07-31

New Expense for (thanuja) created on 2024-07-31

New category (Groceries) created on 2024-07-31

New Report for (thanuja) created on 2024-07-31



New category (Gadgets) created on 2024-07-31

New Expense for (thanuja) created on 2024-07-31

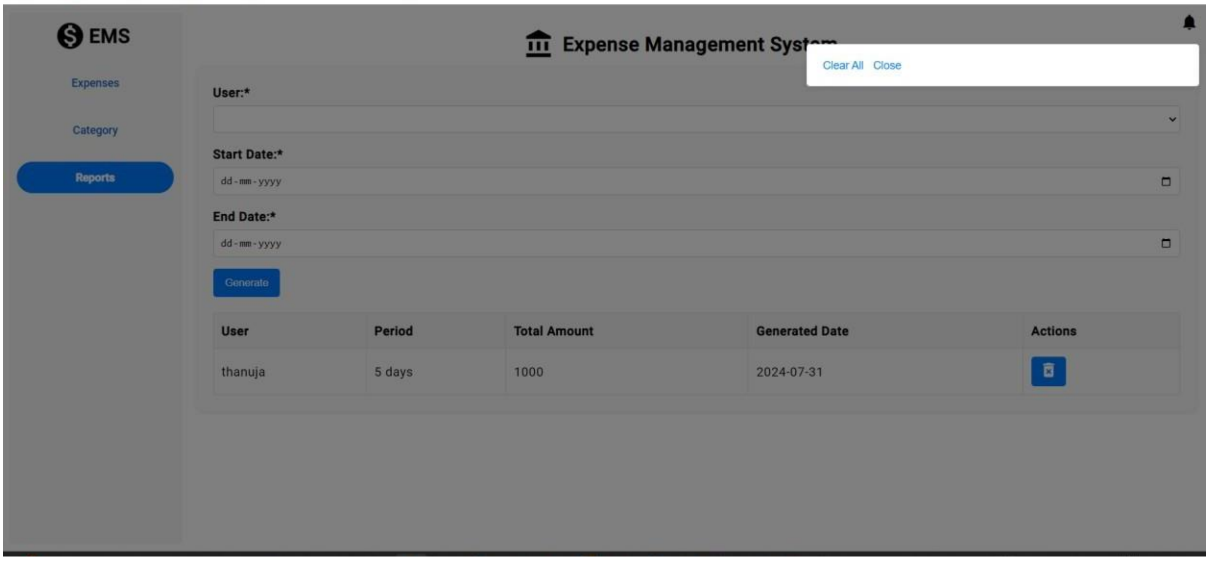
New Report for (thanuja) created on 2024-07-31

User:*

Start Date:* dd-mm-yyyy

End Date:* dd-mm-yyyy

User	Period	Total Amount	Generated Date	Actions
thanuja	5 days	1000	2024-07-31	



User:*

Start Date:* dd-mm-yyyy

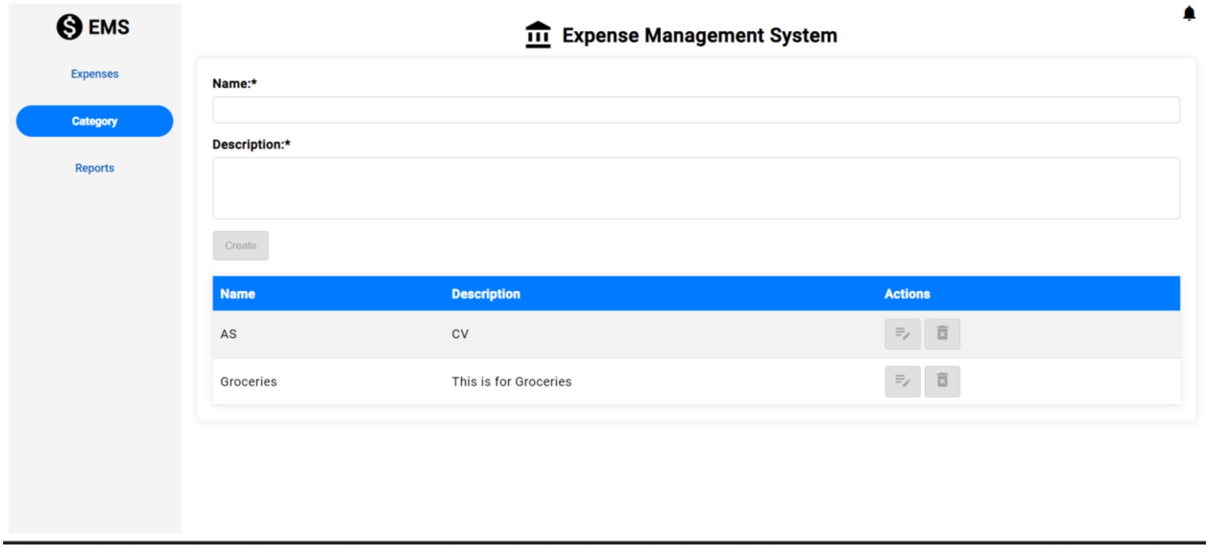
End Date:* dd-mm-yyyy

User	Period	Total Amount	Generated Date	Actions
thanuja	5 days	1000	2024-07-31	

1. **Module 6: Email Notification System and Feedback**

- Integrate email and/or SMS notification services.
- Develop triggers and notifications for subscription reminders, new categories, and expenses exhaust based on target money.
- Implement a system for handling and tracking notifications.
- Integrate AI tools with Spring boot to generate automatically message content and send to the respective candidates on their booking services.
- Implement validations for email and mobile to avoid duplicate entries.
- Write test cases using JUnit and Mockito for all the Services with code coverage.
- Implement respective module frontend using Angular/React JS and write test cases for frontend logic.

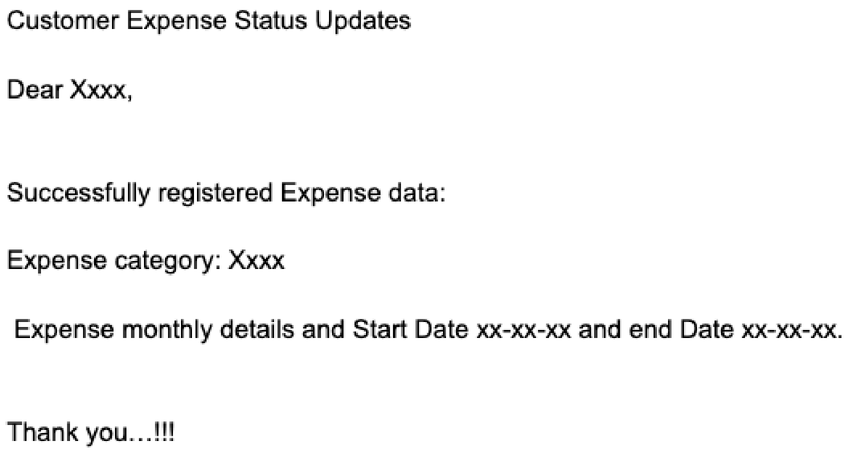
- Form validations must be implemented for all the services.
- **Output screenshot:**



Name:*

Description:*

Name	Description	Actions
AS	CV	
Groceries	This is for Groceries	



Customer Expense Status Updates

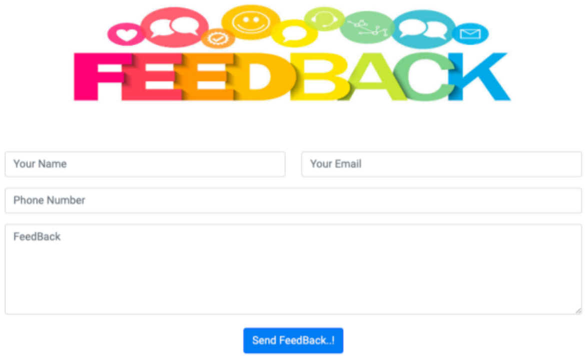
Dear Xxxx,

Successfully registered Expense data:

Expense category: Xxxx

Expense monthly details and Start Date xx-xx-xx and end Date xx-xx-xx.

Thank you…!!!



Your Name

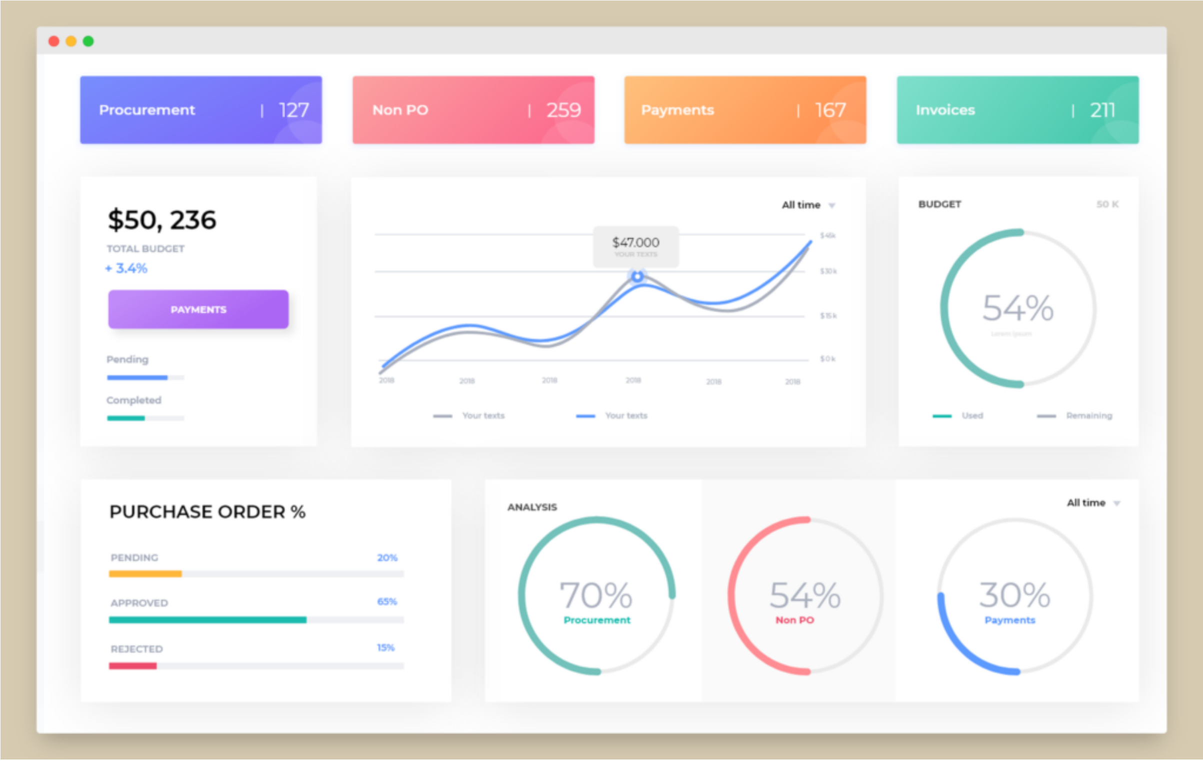
Your Email

Phone Number

FeedBack

Module 7: Admin Dashboard

- Design the layout and functionality of the admin dashboard.
- Implement user management features for admins to add, edit, and delete users.
- Develop modules for managing payments, wallets, user, and expense information.
- Write test cases using JUnit and Mockito for all the Services with code coverage.
- Implement respective module frontend using Angular/React JS and write test cases for frontend logic.
- Form validations must be implemented for all the services.
- **Output Screenshot:**



Evaluation Criteria:

Milestone 1 Evaluation (Week 3):

- Completion of user authentication and registration.
- Successful implementation of movie functionalities.

Milestone 2 Evaluation (Week 5):

- The Expense and report interface is fully developed and integrated with the front end and back end.

Milestone 3 Evaluation (Week 7):

- The category management system is operational, allowing for choosing and viewing expense services or respective details.

Milestone 4 Evaluation (Week 10):

- The notification system, feedback and admin dashboard are fully implemented and functional.

5. Design diagrams

Class Diagram: Entity classes

1. User Module

• User

o UserID (Primary Key)

o Name

o Email

o Role (Admin, User)

o Password (hashed)

Relationships:

• Each user can have multiple expenses.

• Users can have different roles affecting their permissions within the system.

2. Expense Module

• Expense

o ExpenseID (Primary Key)

o UserID (Foreign Key)

o Amount

o Date

o CategoryID (Foreign Key)

o Description

Relationships:

- Each expense is linked to a user.
- Each expense falls under a specific category.

3. Category Module

- Category

o CategoryID (Primary Key)

o Name

o Description

Relationships:

- Categories are used to classify expenses.
- Categories are predefined and can be managed by admin users.

4. Report Module

- Report

o ReportID (Primary Key)

o UserID (Foreign Key)

o Period (Month, Year)

o TotalAmount

o GeneratedDate

Relationships:

- Reports are generated for users based on their expenses.

5. Notification Module

- Notification

o NotificationID (Primary Key)

o UserID (Foreign Key)

o Message

o Date

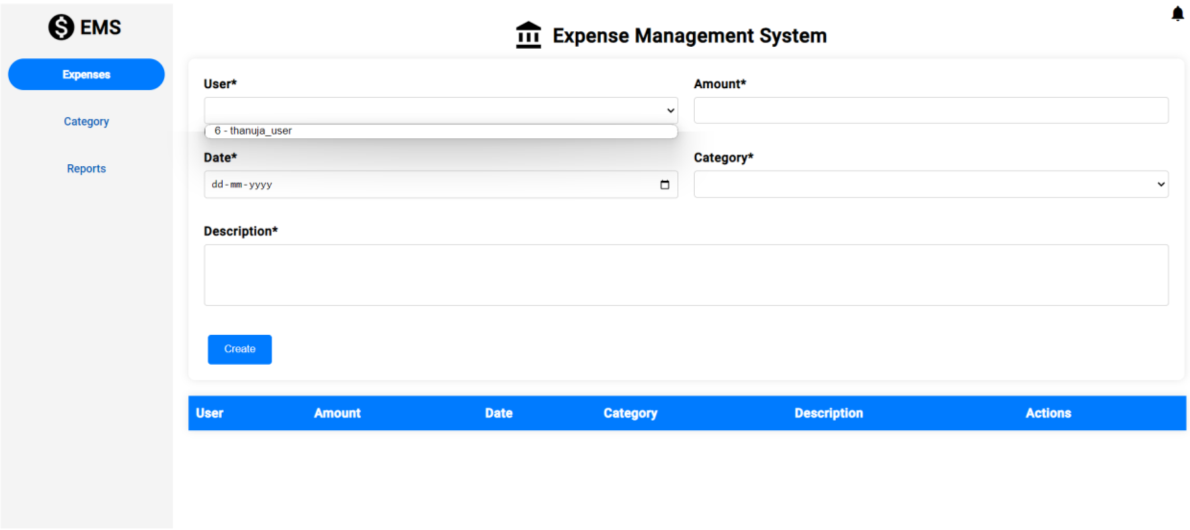
o Status (Read, Unread)

Relationships:

- Notifications are sent to users for various events (e.g., expense submission, report generation).

Output Screenshots:

1. Dashboard User:



User*

6 - thanuja_user

Amount*

Date*

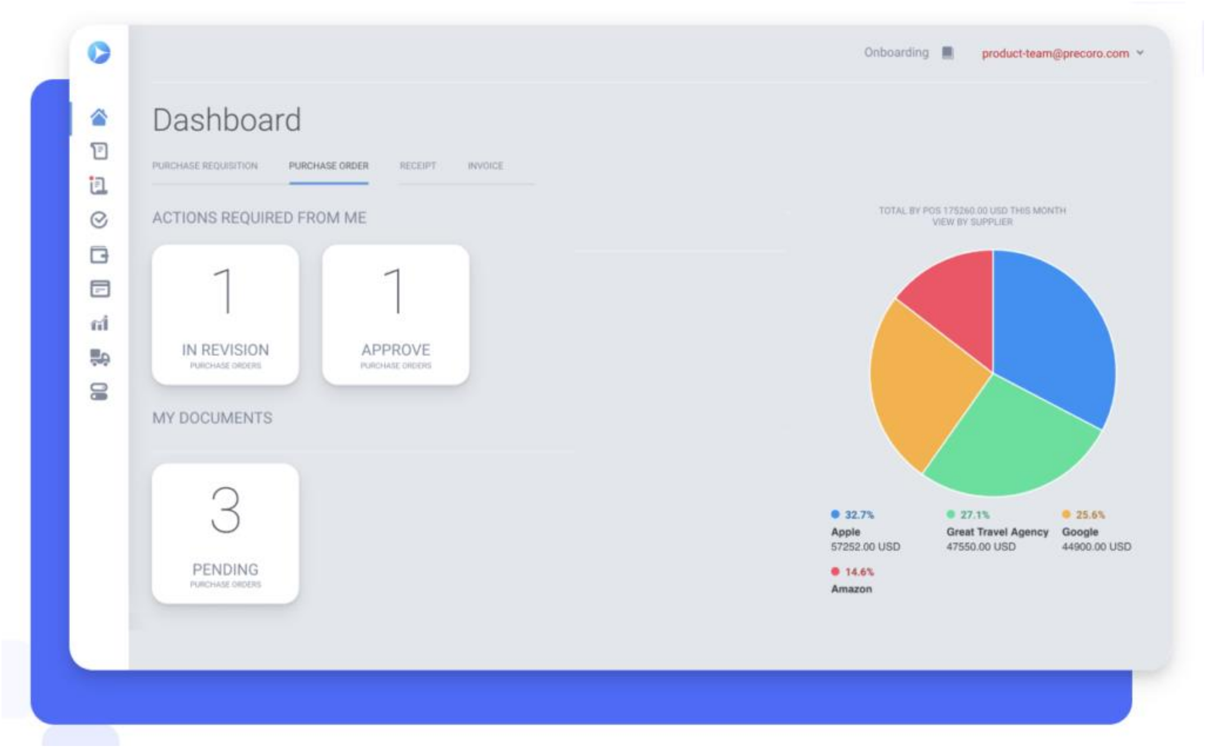
dd-mm-yyyy

Category*

Description*

User	Amount	Date	Category	Description	Actions

2. Admin Dashboard


Onboarding
product-team@precoro.com
Dashboard
PURCHASE REQUISITION
PURCHASE ORDER
RECEIPT
INVOICE
ACTIONS REQUIRED FROM ME
1
IN REVISION
PURCHASE ORDERS
1
APPROVE
PURCHASE ORDERS
MY DOCUMENTS
3
PENDING
PURCHASE ORDERS
TOTAL BY POS 175260.00 USD THIS MONTH
VIEW BY SUPPLIER
● 32.7%
Apple
57252.00 USD
● 27.1%
Great Travel Agency
47550.00 USD
● 25.6%
Google
44900.00 USD
● 14.6%
Amazon