

# Project Report: Expense Management System

## Overview & Features

### 1. Introduction

We have developed an Expense Management System with role-based access control, allowing **Admins** and **Users** to track and manage expenses efficiently.

### 2. Features Implemented

#### Admin Dashboard:

- View **total users**, **total categories**, **total transactions**, and **total amount spent**.
- Export statistics as a **PDF report**.

#### User Dashboard:

- View **personal expense statistics**, including total transactions and total amount spent.
- Export user-specific statistics as a **PDF report**.

#### Expense Chart:

- Displays categorized spending trends.
- Integrated **MUI charts** for visualization.

#### Data Fetching & State Management:

- Used `fetchStats` API to dynamically fetch statistics.
- Managed state efficiently with `useState` and `useEffect`.

#### PDF Export Functionality:

- Implemented using **jsPDF** and **autoTable** for structured reports.

- Data formatted into tables and exported with a **single click**.

## Implementation Details & Technologies Used

### 1. Technologies Used:

- **Frontend:** React (Vite), Material-UI
- **State Management:** React Hooks (useState, useEffect)
- **PDF Generation:** jsPDF, autoTable
- **Icons & Styling:** Material-UI Icons & Theme
- **API Integration:** Fetch API for retrieving statistics

### 2. Key Implementations

#### Reusable PDF Export Utility

- Created a separate exportToPDF.js file for code reusability.
- Accepts dynamic **columns, data, title, and filename**.

#### Role-Based Dashboard Rendering

- **Admins** see global statistics.
- **Users** see only their own expenses.
- Dashboard dynamically updates based on **localStorage role**.

#### Dynamic UI Components

- Used **Material-UI Cards** to display statistics with intuitive icons.

- Implemented a **centralized ExpenseChart** for better data visualization.

### **Error Handling & Loading State**

- Implemented proper **error messages** for API failures.
- Used a **loading indicator** while fetching data.

### **Outcome & Next Steps**

**Fully functional** dashboard with PDF export.

**Optimized & scalable** frontend with reusable code.

### **Future Enhancements:**

- Implement user-specific filtering for expense tracking.
- Add advanced analytics for expense trends.