# Expense Tracker Documentation

## 1. Introduction

The Expense Tracker is a web-based application designed to help companies manage and monitor their financial transactions efficiently. The system records expenses, categorizes them, and provides detailed insights through reports and analytics. It is built using Flask for the backend and React for the frontend, with PostgreSQL as the database.

## 2. Features

* **User Authentication**: Secure login/logout system with role-based access.
* **Expense Entry**: Users can record expenses with details like date, category, amount, and description.
* **Expense Categories**: Customizable categories for better financial organization.
* **Expense Approval System**: Allows managers to review and approve expenses.
* **Reporting & Analytics**: Provides graphical and tabular insights into spending patterns.
* **Export & Import**: Supports CSV and PDF export for reporting purposes.
* **Multi-user Support**: Supports different user roles such as Admin, Employee, and Manager.
* **Notifications & Alerts**: Sends reminders for pending approvals and expense limits.

## 3. Technology Stack

* **Frontend**: React, Redux, Material-UI
* **Backend**: Flask, Flask-RESTful, Flask-JWT
* **Database**: PostgreSQL
* **Storage**: Amazon S3 (for receipts and invoices)
* **Authentication**: JWT-based authentication
* **Hosting**: AWS/GCP/DigitalOcean

## 4. System Architecture

1. **Frontend (React)**: Handles the UI and user interactions.
2. **Backend (Flask API)**: Provides RESTful API endpoints for data processing.
3. **Database (PostgreSQL)**: Stores user details, expenses, and categories.
4. **File Storage (S3)**: Stores uploaded receipts and invoices.
5. **Security**: JWT authentication and role-based access control.

## 5. API Endpoints

### User Management

* **POST /register** – Register a new user
* **POST /login** – Authenticate and get JWT token
* **GET /user/{id}** – Get user details

### Expense Management

* **POST /expense** – Add new expense
* **GET /expenses** – Retrieve all expenses
* **PUT /expense/{id}** – Update an expense
* **DELETE /expense/{id}** – Delete an expense

### Reporting

* **GET /report/summary** – Get expense summary
* **GET /report/category** – Get expense report by category

## 6. Installation & Setup

### Prerequisites:

* Python 3.x
* Node.js and npm
* PostgreSQL

### Backend Setup:

1. Clone the repository: git clone <repo-url>
2. Navigate to the backend directory: cd backend
3. Install dependencies: pip install -r requirements.txt
4. Set up environment variables in .env
5. Run migrations: flask db upgrade
6. Start the server: flask run

### Frontend Setup:

1. Navigate to the frontend directory: cd frontend
2. Install dependencies: npm install
3. Start the React app: npm start

## 7. Security Measures

* **Data Encryption**: Uses HTTPS and encrypted JWT tokens.
* **Role-based Access Control**: Ensures restricted access to sensitive features.
* **Input Validation**: Prevents SQL injection and XSS attacks.

## 8. Future Enhancements

* AI-based expense predictions
* Mobile application
* Integration with accounting software

## 9. Conclusion

The Expense Tracker system provides a comprehensive solution for tracking, managing, and analyzing expenses for organizations. It enhances transparency, improves decision-making, and streamlines financial operations.