

EXPENSE MANAGEMENT

A MINI-PROJECT REPORT

Submitted by:

THULASI.S 241901118

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING(Cyber Security)



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project "**EXPENSE MANAGEMENT**" is the bona fide work of "**THULASI.S**" who carried out the project work under my supervision.

SIGNATURE

MS.M. FOWZIA SIHANA

ASSISTANT PROFESSOR (SS)

Department of Computer Science and Engineering (Cyber Security)

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voice examination to be held on

_____ .

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

In today's competitive business world, efficient project expense management is vital for organisational success. Companies often struggle to track expenses, manage vendors, monitor budgets, and generate reports. This project aims to develop a comprehensive database system that centralises expense tracking across multiple projects, vendors, and cost categories. The system enables organisations to control spending, prevent budget overruns, and produce insightful financial reports, ultimately optimising financial operations and improving decision-making.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honourable chairman, **MR. S.MEGANATHAN** and the chairperson, **Dr M.THANGAM MEGANATHAN**, for their timely support and encouragement.

We are greatly indebted to our respected and honourable principal **Dr S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Mr.Benedict Jayaprakash Nicholas** for being an ever-supportive force during our project work.

We also extend our sincere and hearty thanks to our internal guide **MS.M. FOWZIA SIHANA**, for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering (Cyber Security).

TABLE OF CONTENTS

i. ABSTRACT

1. INTRODUCTION

 1.1 INTRODUCTION

 1.2 SCOPE OF THE WORK

 1.3 PROBLEM STATEMENT

 1.4 AIM AND OBJECTIVES OF THE PROJECT

2. SYSTEM SPECIFICATIONS

 2.1 SOFTWARE SPECIFICATIONS

 2.2 HARDWARE SPECIFICATIONS

3. MODULE DESCRIPTION

4. DATABASE DESIGN

 4.1 ENTITY RELATIONSHIP DIAGRAM

 4.2 TABLE STRUCTURE

5. SQL IMPLEMENTATION

6. SCREENSHOTS

7. CONCLUSION AND FUTURE ENHANCEMENT

8. REFERENCE

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION:

The Project Expense Management System automates expense tracking, budgeting, and reporting for projects. It ensures accuracy, security, and real-time financial insights through role-based access and secure data management.

1.2 SCOPE OF THE WORK:

The Project Expense Management System encompasses the following functionalities:

User Management: Register and manage multiple users with different Business

Management: Store and organize multiple business entities Account Management:

Maintain financial accounts with balance tracking Vendor Management: Store vendor

information and contact details Customer Management: Track customer information and

contact details Category Management: Define and manage expense categories with type

classification Transaction Management: Record all financial transactions with date,

amount, and descriptions Invoice Management: Generate and track customer invoices

with payment status Budget Management: Set monthly or yearly budgets for expense

categories with limit tracking Reporting: Generate expense reports, budget vs. actual

analysis, and vendor payment summaries.

1.3 PROBLEM STATEMENT:

Organisations struggle with manual expense management due to scattered tracking, budget overruns, poor vendor management, weak audit trails, time-consuming reporting, and a lack of authorisation control.

1.4 AIM AND OBJECTIVES OF THE PROJECT:

The project aims to build a user-friendly system for efficient expense tracking and analysis. Its objectives include centralised data storage, secure role-based access, real-time budget monitoring, vendor management, automated reporting, audit trails, and multi-project tracking.

KEY OBJECTIVES:

1. Provide a centralised database for storing all expense-related information.
2. Implement role-based access control for enhanced security.
3. Enable real-time budget monitoring and alert generation for budget overruns.
4. Facilitate efficient vendor and customer payment management.
5. Generate comprehensive expense reports and financial analytics.
6. Maintain audit trails for compliance and verification.
7. Automate expense categorisation and allocation.
8. Provide multi-project expense tracking capabilities.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 SOFTWARE SPECIFICATIONS:

Operating System	:	WINDOWS 11
Front – End	:	Python
Back - End	:	SqlLite
Language	:	python,SQL

2.2 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

CHAPTER 3

MODULE DESCRIPTION

Module 1: User Authentication & Management

This module handles user registration, login, and role-based access control. Users can have roles such as Admin (full access), Manager (create and view expenses), Accountant (view and process payments), and Viewer (read-only access).

Key Features:

- User registration with email verification
- Secure password authentication
- Role-based access control (RBAC)
- Session management
- Password recovery and reset

Module 2: Business & Account Management:

This module allows organizations to manage multiple businesses and their associated financial accounts.

Key Features:

- Create and manage business profiles
- Track multiple bank/financial accounts per business
- Real-time account balance updates
- Account statement generation

Module 3: Expense & Transaction Management:

This module is the core of the system, enabling users to record and categorize expenses.

Key Features:

- Record transactions with category assignment
- Support for multiple vendors and customers
- Transaction history and audit trails
- Transaction search and filtering

- Receipt attachment capability

Module 4: Budget Management:

Allows setting and monitoring budgets for different expense categories.

Key Features:

- Set budget limits per category and time period
- Real-time budget vs. actual tracking
- Budget alert notifications for overruns
- Budget variance analysis

Module 5: Reporting & Analytics:

Generates comprehensive financial reports and analytics.

Key Features:

- Expense reports by category, vendor, or time period
- Budget variance reports
- Cash flow analysis
- Vendor payment summaries
- Export to CSV/PDF formats

Module 6: Vendor & Customer Management:

Manages vendor and customer information with payment tracking.

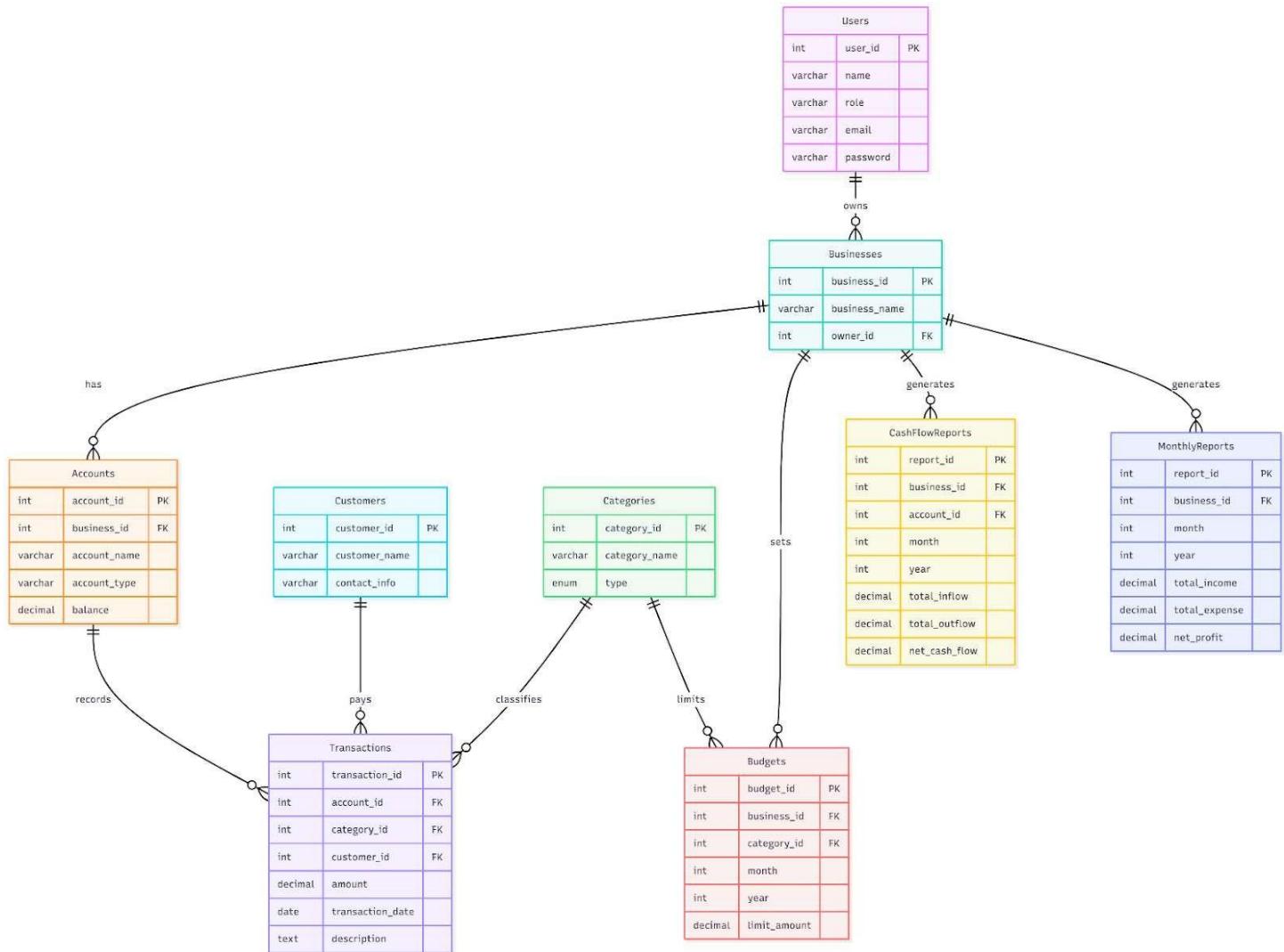
Key Features:

- Vendor profile management
- Customer invoice tracking
- Payment status monitoring
- Customer communication history

CHAPTER 4

DATABASE DESIGN

4.1 ENTITY RELATIONSHIP DIAGRAM



4.2 TABLE STRUCTURE

Users Table:

Users		
int	user_id	PK
varchar	name	
varchar	role	
varchar	email	
varchar	password	

Businesses Table:

Businesses		
int	business_id	PK
varchar	business_name	
int	owner_id	FK

Accounts Table:

Accounts		
int	account_id	PK
int	business_id	FK
varchar	account_name	
varchar	account_type	
decimal	balance	

Customers Table:

Customers		
int	customer_id	PK
varchar	customer_name	
varchar	contact_info	

Categories Table:

Categories		
int	category_id	PK
varchar	category_name	
enum	type	

Transactions Table:

Transactions		
int	transaction_id	PK
int	account_id	FK
int	category_id	FK
int	customer_id	FK
decimal	amount	
date	transaction_date	
text	description	

Budgets Table:

Budgets		
int	budget_id	PK
int	business_id	FK
int	category_id	FK
int	month	
int	year	
decimal	limit_amount	

CashFlowReports Table:

CashFlowReports		
int	report_id	PK
int	business_id	FK
int	account_id	FK
int	month	
int	year	
decimal	total_inflow	
decimal	total_outflow	
decimal	net_cash_flow	

MonthlyReports Table:

MonthlyReports		
int	report_id	PK
int	business_id	FK
int	month	
int	year	
decimal	total_income	
decimal	total_expense	
decimal	net_profit	

CHAPTER 5

SQL IMPLEMENTATION

Creation of Table:

```
expense_management > apps > users > 🐻 models.py > 📁 Users
You, 4 weeks ago | 1 author (You)
1 from django.db import models
2
3
4
5 You, 4 weeks ago | 1 author (You)
6 class Users(models.Model):      You, 4 weeks ago • Initial commit
7     user_id = models.IntegerField(primary_key = True)
8     name = models.CharField(max_length = 100)
9     role = models.CharField(max_length = 50)
10    email = models.CharField(max_length = 100)
11    password = models.CharField(max_length = 100)
12
13    def __str__(self):
14        return f"{self.name} ({self.role})"
```

Registrations Of The Table:

```
expense_management > apps > users > 🐻 admin.py
You, 1 second ago | 1 author (You)
1 from django.contrib import admin
2 from .models import Users
3
4 admin.site.register(Users)
5
6
7
8 |
```

Insertion Of Values Into Table:

	customer_id	name	email	user_id
1	1	kumar	kumar@gmail.com	NULL
2	2	rahul	rahul@gmail.com	NULL
3	3	sundar	sundar@gmail.com	NULL
4	4	janr	dd@gmail.com	NULL
	5			

Logic Behind The Code:

```

from django.shortcuts import render, redirect, get_object_or_404
from apps.users.decorators import customer_required
from .models import Customer
from .forms import CustomerForm

@customer_required
def customer_list(request):
    customers = Customer.objects.all()
    return render(request, 'customers/customer_list.html', {'customers': customers})

def customer_create(request):
    if request.method == 'POST':
        form = CustomerForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('customer_list')
    else:
        form = CustomerForm()
    return render(request, 'customers/customer_form.html', {'form': form})

def customer_update(request, pk):
    customer = get_object_or_404(Customer, pk=pk)
    if request.method == 'POST':
        form = CustomerForm(request.POST, instance=customer)
        if form.is_valid():
            form.save()
            return redirect('customer_list')
    else:
        form = CustomerForm(instance=customer)
    return render(request, 'customers/customer_form.html', {'form': form})

def customer_delete(request, pk):
    customer = get_object_or_404(Customer, pk=pk)
    if request.method == 'POST':
        customer.delete()
        return redirect('customer_list')      You, 4 weeks ago • Initial commit ...
    return render(request, 'customers/customer_confirm_delete.html', {'customer': customer})

```

Frontend For The Code:

Customers_list HTML:

```

customer_list.html × customer_confirm_delete.html customer_form.html base.html ...
expense_management > apps > customers > templates > customers > customer_list.html
You, 3 weeks ago | 1 author (You)
1  {% extends 'customers/base.html' %}      You, 3 weeks ago • Updated quiz feature logic and UI
2  {% load static %}
3
4  {% block title %}Customers{% endblock %}
5
6  {% block content %}
7    <h2>Customers</h2>
8    <p><a class="btn" href="{% url 'customer_create' %}">Add customer</a></p>
9    <table class="table">
10      <thead>
11        <tr>
12          <th>ID</th>
13          <th>Name</th>
14          <th>Email</th>
15          <th>Actions</th>
16        </tr>
17      </thead>
18      <tbody>
19        {% for c in customers %}
20          <tr>
21            <td>{{ c.customer_id }}</td>
22            <td>{{ c.customer_name }}</td>
23            <td>{{ c.email }}</td>
24            <td>
25              <a href="{% url 'customer_update' c.customer_id %}">Edit</a> |
26              <a href="{% url 'customer_delete' c.customer_id %}">Delete</a>
27            </td>
28          </tr>
29        {% empty %}
30        <tr><td colspan="4">No customers yet.</td></tr>
31      {% endfor %}
32    </tbody>
33  </table>
34  {% endblock %}
35

```

Customers_Delete HTML:

```

customer_confirm_delete.html
expense_management > apps > customers > templates > customers > customer_confirm_delete.html
You, 3 weeks ago | 1 author (You)
1  {% extends 'customers/base.html' %}      You, 3 weeks ago • Updated quiz feature logic and UI
2  {% load static %}
3
4  {% block title %}Delete Customer{% endblock %}
5
6  {% block content %}
7    <h2>Delete Customer</h2>
8    <p>Are you sure you want to delete customer {{ customer.customer_name }} ({{ customer.customer_email }})?</p>
9    <form method="post">
10      {% csrf_token %}
11      <button type="submit" class="btn">Yes, delete</button>
12      <a href="{% url 'customer_list' %}">Cancel</a>
13    </form>
14  {% endblock %}
15

```

Customers Form:

```
expense_management > apps > customers > templates > customers > dj customer_form.html
You, 3 weeks ago | 1 author (You)
1  {% extends 'customers/base.html' %}          You, 3 weeks ago • Updated quiz feature logic and UI
2  {% load static %}
3
4  {% block title %}Customer Form{% endblock %}
5
6  {% block content %}
7      <h2>Customer</h2>
8      <form method="post">
9          {% csrf_token %}
10         <table>
11             {{ form.as_table }}
12         </table>
13         <button type="submit" class="btn">Save</button>
14         <a href="{% url 'customer_list' %}">Cancel</a>
15     </form>
16  {% endblock %}
17
```

Urls For The Site:

```
expense_management > apps > customers > py urls.py > ...
You, 4 weeks ago | 1 author (You)
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('', views.customer_list, name='customer_list'),
6      path('add/', views.customer_create, name='customer_create'),
7      path('edit/<int:pk>/', views.customer_update, name='customer_update'),
8      path('delete/<int:pk>/', views.customer_delete, name='customer_delete'),
9  ]
10
```

CHAPTER 6:

SCREENSHOTS

Home Page:

The screenshot shows the homepage of an "Expense Management" application. At the top, there is a navigation bar with the following items: a three-line menu icon, the "Expense Management" logo, the text "Logged in as Jeffery", and links for "Home", "Businesses", "Accounts", "Customers", "Categories", "Transactions", "Budgets", "Reports", "Dashboard", and "Logout". Below the navigation bar, the main content area has a light gray background. In the center, the text "Welcome to Users" is displayed in a large, bold, dark font. Below this, a smaller text states: "This dashboard shows a quick summary of recent activity and links to reports generated from your data." Underneath, a section titled "Current month (11/2025)" lists the following activity: "Transactions this month: 0", "Total income: 0", and "Total expenses: 0". At the bottom of the content area, there is a section titled "Quick links" containing four hyperlinks: "Reports home", "Monthly report", "Cashflow report", and "View all users".

20

20

User Page:

The screenshot shows the 'Expense Management' application interface. At the top left, it says 'Expense Management' and 'Logged in as Jeffery'. On the right, there's a navigation bar with links: Home, Businesses, Accounts, Customers, Categories, Transactions, Budgets, Reports, Dashboard, and Logout. Below the navigation is a large white box titled 'Users'. Inside, there's a sub-header 'Add new user' and a table with three rows of data. The table has columns for ID, Name, Email, and Actions (Edit | Delete). The data is as follows:

ID	Name	Email	Actions
1	Tom (Sales)	Tom@gmail.com	Edit Delete
2	Sam (Finances)	Sam@gmail.com	Edit Delete
3	Max (Toys Business)	Max@gmail.com	Edit Delete

At the bottom right of the main page area, it says '© Expense Management'.

Businesses:

The screenshot shows the 'Expense Management' application interface. At the top left, it says 'Expense Management' and 'Logged in as Jeffery'. On the right, there's a navigation bar with links: Home, Businesses, Accounts, Customers, Categories, Transactions, Budgets, Reports, Dashboard, and Logout. Below the navigation is a large white box titled 'Businesses'. It contains four separate card-like boxes, each representing a business:

- Soap company**: Owner: Tom (Sales). Buttons: Edit (white), Delete (blue).
- Little Toys**: Owner: Max (Toys Business). Buttons: Edit (white), Delete (blue).
- Helping Finances**: Owner: Sam (Finances). Buttons: Edit (white), Delete (blue).
- Iron export**: Owner: Nandy (Business). Buttons: Edit (white), Delete (blue).

Below these, there's another card:

- Software As A Service**: Owner: George (Business). Buttons: Edit (white), Delete (blue).

21

21

Accounts For the Businesses Created:

The screenshot shows a web-based expense management application. At the top, there is a navigation bar with links for Home, Businesses, Accounts, Customers, Categories, Transactions, and Budgets. Below the navigation bar, it says "Logged in as Jeffery". On the left, there is a sidebar with a "Expense Management" logo and a "Add account" button. The main content area displays a table of accounts, each with an ID, business name, name, type, balance, and edit/delete links.

ID	Business	Name	Type	Balance	Actions
1	Soap company (Owner: Tom)	Business account	debit card	1000000.0	Edit Delete
2	Little Toys (Owner: Max)	Business account	credit	500000.0	Edit Delete
3	Helping Finances (Owner: Sam)	Business account	debit card	300000.0	Edit Delete
4	Iron export (Owner: Nandy)	Business account	credit	100000.0	Edit Delete
5	Software As A Service (Owner: George)	Business account	debit card	250000.0	Edit Delete

Customers Accounts For The Businesses:

The screenshot shows a web-based expense management application. At the top, there is a navigation bar with links for Home, Businesses, Accounts, Customers, Categories, Transactions, and Budgets. Below the navigation bar, it says "Logged in as Jeffery". On the left, there is a sidebar with a "Expense Management" logo and a "Add customer" button. The main content area displays a table of customers, each with an ID, name, email, and edit/delete links.

ID	Name	Email	Actions
1	kumar	kumar@gmail.com	Edit Delete
4	rahul	rahul@gmail.com	Edit Delete
5	sundar	sundar@gmail.com	Edit Delete
6	janu	dd@gmail.com	Edit Delete

Different Type Of Categories:

ID	Name	Type	Actions
1	Helping Finances	finance	Edit Delete
2	Iron export	Export	Edit Delete
3	Software As A Service	Online Service	Edit Delete
4	Little Toys	Toys	Edit Delete

Transaction History From Company to Cutomers:

ID	Account	Category	Customer	Amount	Date	Actions
1	Business account (debit card) - Balance: 1000000.0	soap Company (Soap)	kumar (kumar@gmail.com)	50000.00	Nov. 5, 2025	Edit Delete
2	Business account (credit) - Balance: 500000.0	Little Toys (Toys)	rahul (rahul@gmail.com)	60000.00	Nov. 5, 2025	Edit Delete
3	Business account (debit card) - Balance: 300000.0	Software As A Service (Online Service)	sundar (sundar@gmail.com)	80000.00	Nov. 5, 2025	Edit Delete
4	Business account (credit) - Balance: 100000.0	Iron export (Export)	janu (dd@gmail.com)	45000.00	Nov. 5, 2025	Edit Delete

Setting Budgets For Next Transaction:

The screenshot shows the 'Expense Management' application interface. At the top, there is a navigation bar with links for Home, Businesses, Accounts, Customers, Categories, Transactions, Budgets, Reports, Dashboard, and Logout. A user is logged in as 'Jeffery'. The main content area is titled 'Budgets' and contains a table with four rows of budget data. The columns are labeled ID, Business, Customer, Period, Limit, and Actions.

ID	Business	Customer	Period	Limit	Actions
1	Soap company (Owner: Tom)	kumar (kumar@gmail.com)	11/2025	25000.0	Edit Delete
2	Little Toys (Owner: Max)	rahul (rahul@gmail.com)	11/2025	45000.0	Edit Delete
3	Helping Finances (Owner: Sam)	sundar (sundar@gmail.com)	11/2025	56000.0	Edit Delete
4	Iron export (Owner: Nandy)	janu (dd@gmail.com)	11/2025	23000.0	Edit Delete

Monthly Reports Based On The Transaction:

The screenshot shows the 'Expense Management' application interface. At the top, there is a navigation bar with links for Home, Businesses, Accounts, Customers, Categories, Transactions, Budgets, Reports, Dashboard, and Logout. A user is logged in as 'Jeffery'. The main content area is titled 'Monthly Report — November 2025'.

Transactions this month: 4

- Total income: **0.00**
- Total expenses: **235000.00**
- Net profit: **-235000.00**
- Budget total: **149000.00**

Income change vs previous month: **N/A**

Expenses change vs previous month: **N/A**

3-month rolling average expenses: **78333.33**

Top expense categories

No expense transactions this month.

Category breakdown (by expense share)

Separate Credentials Login For Different Users:

The screenshot shows a web application interface for 'Expense Management'. At the top left is the application logo 'Expense Management'. To its right, the text 'Not logged in' is displayed. Next to it is a small icon consisting of three horizontal lines. Along the top navigation bar are links for 'Home', 'Businesses', 'Accounts', 'Customers', 'Categories', 'Transactions', and 'Budgets'. Below the navigation bar, there are two more links: 'Reports' and 'Dashboard'. The main content area is titled 'Login'. It contains two input fields: one for 'Username' and one for 'Password'. Below these fields is a 'Log in' button. Underneath the login form, the text 'Are you a customer or a vendor?' is displayed in bold. A note below it says, 'If you're a customer or vendor, use the links below to enter your information and then log in.' Two links are listed: 'Customer registration / enter info' and 'Vendor / Business registration'. At the bottom of the content area, there is a link 'Back to home'.

Expense Management

Not logged in

☰

Home Businesses Accounts Customers Categories Transactions Budgets

Reports Dashboard Login

Login

Username:

Password:

Log in

Are you a customer or a vendor?

If you're a customer or vendor, use the links below to enter your information and then log in.

- [Customer registration / enter info](#) (creates a customer record)
- [Vendor / Business registration](#) (create your business)

[Back to home](#)

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

The Project Expense Management System provides organisations with a comprehensive solution for tracking, managing, and analysing project expenses. By implementing this system, organisations can:

- Achieve better financial visibility and control
- Prevent budget overruns through real-time monitoring
- Reduce time spent on expense management and reporting
- Ensure compliance and maintain audit trails
- Make data-driven financial decisions

This system demonstrates the practical application of database design principles and provides a scalable foundation for enterprise expense management.

Future Enhancements:

- 1. Mobile Application:** Develop mobile apps for on-the-go expense tracking.
- 2. AI-Powered Categorisation:** Implement machine learning for automatic expense categorisation.
- 3. Integration with Banking Systems:** Direct bank feed integration for automated transaction import.
- 4. Advanced Analytics:** Implement predictive analytics for budget forecasting.
- 5. Multi-Currency Support:** Enable expense tracking in multiple currencies.
- 6. Integration with Accounting Software:** Connect with QuickBooks, SAP, or other ERP systems.
- 7. Real-time Notifications:** Email/SMS alerts for budget overruns and invoice due dates.
- 8. Customizable Dashboards:** Allow users to create personalised dashboard view.

REFERENCES

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Database System Concepts. McGraw-Hill.

<https://dev.mysql.com/doc/>

<https://www.postgresql.org/docs/>

<https://www.oracle.com/database/documentation/>

<https://docs.djangoproject.com/>

<https://spring.io/projects/spring-boot>

<https://www.w3schools.com/sql/>

https://en.wikipedia.org/wiki/Database_management_system