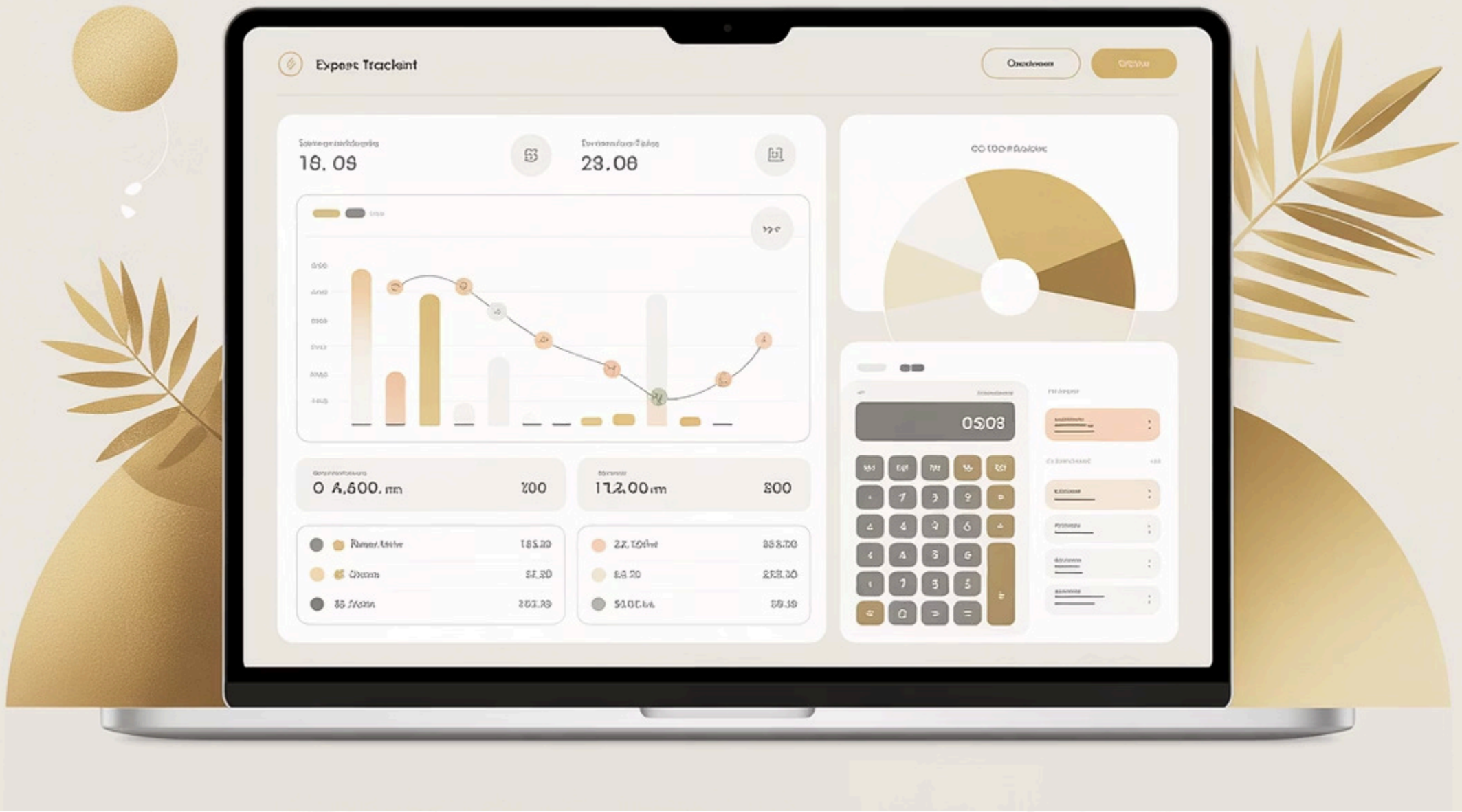


PROJECT REPORT

EXPENSE TRACKER WEB APPLICATION

1. INTRODUCTION

In today's fast-paced life, managing personal expenses efficiently is very important. Many individuals struggle to track their daily spending, which leads to poor financial planning. The Expense Tracker Web Application is developed to help users record, manage, and analyze their expenses in a simple and user-friendly way.

This project is developed using the Django web framework with a SQLite database. It allows users to add, view, edit, and delete their expenses while also providing monthly total expenses and category-wise analysis.

1. OBJECTIVES OF THE PROJECT

The main objectives of this project are:

To develop a secure web-based expense tracking system

To allow users to manage their daily expenses efficiently

To provide monthly expense summaries

To show category-wise expense distribution

To implement user authentication and data privacy

To gain practical experience with Django framework

1. SCOPE OF THE PROJECT

The scope of this project includes:

User registration and authentication

Expense management (CRUD operations)

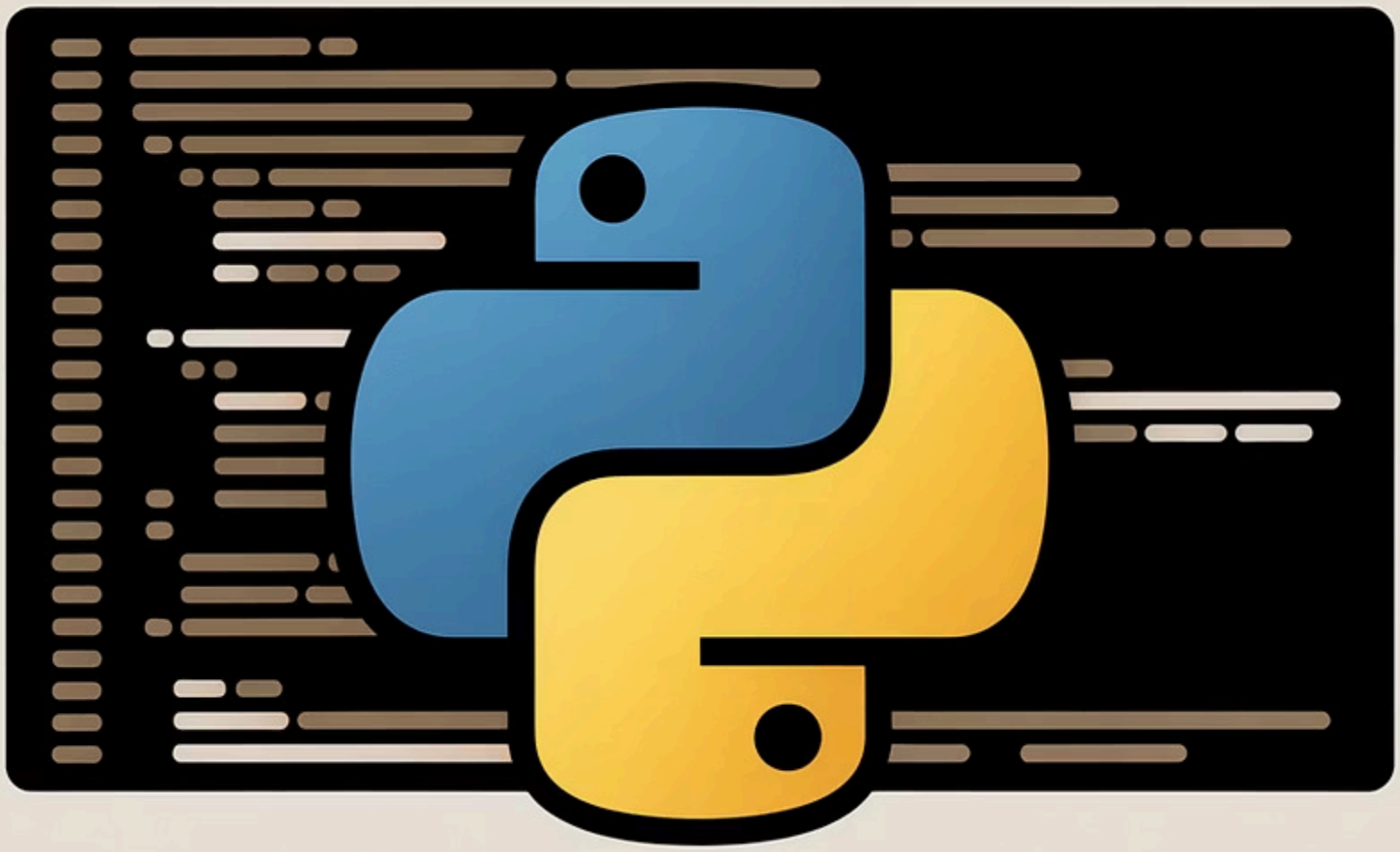
Monthly expense calculation

Category-wise expense analysis

Admin panel for data management

Responsive user interface using Bootstrap

Future enhancements can include charts, reports, and REST APIs.



1. TECHNOLOGY STACK

Frontend

- HTML5
- CSS3
- Bootstrap 5

Backend

- Django (Python Framework)
- Django ORM

Database

- SQLite

Tools & Environment

- Python 3
- Virtual Environment (venv)
- VS Code
- Django Admin Panel

Architecture

- MVT (Model–View–Template)

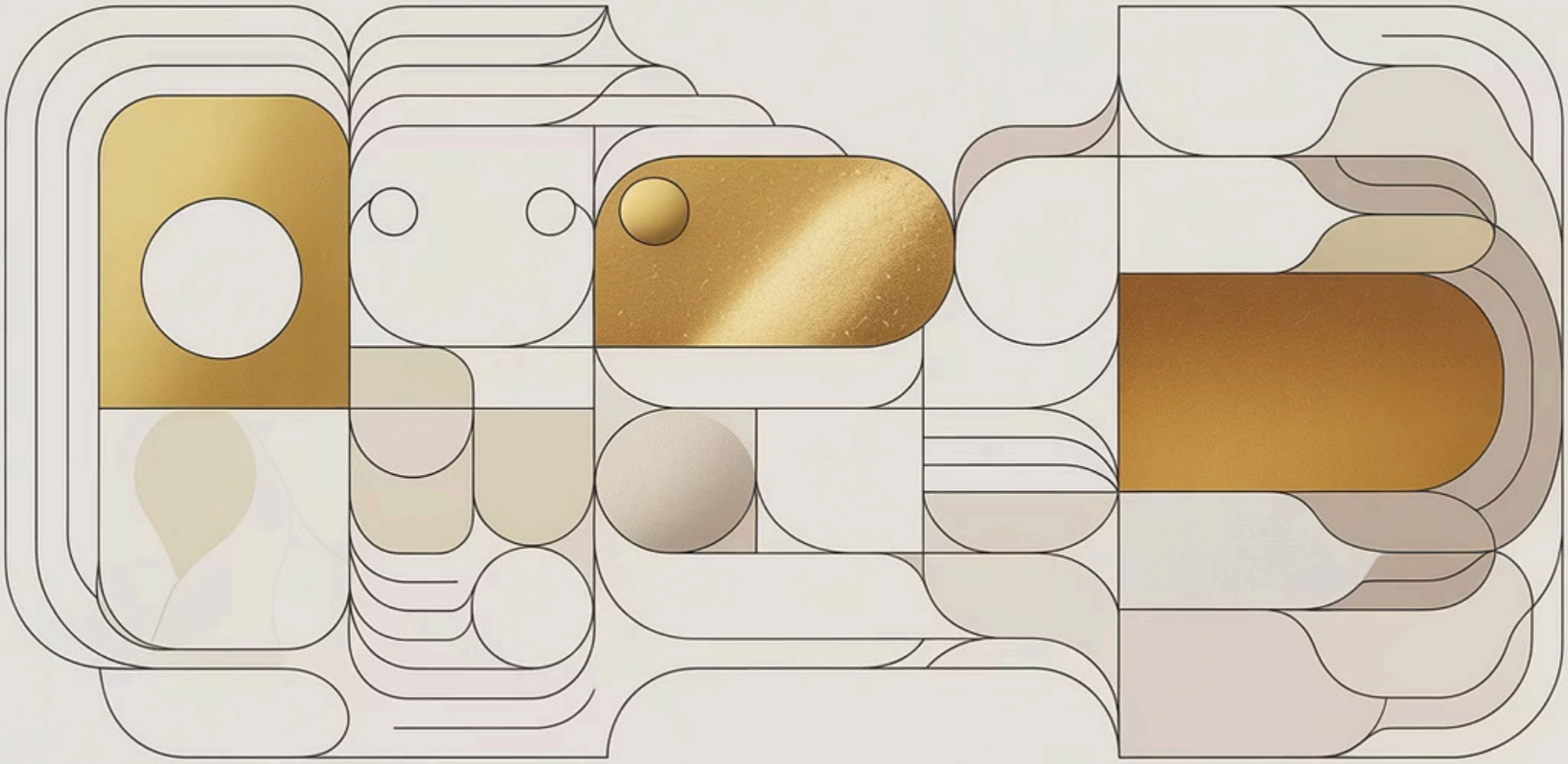
1. SYSTEM REQUIREMENTS

Hardware Requirements

- Minimum 4 GB RAM
- Intel i3 or higher processor

Software Requirements

- Python 3.x
- Django
- Web Browser (Chrome / Firefox)
- Operating System: Windows / Linux / macOS



1. SYSTEM ARCHITECTURE

The project follows Django's MVT architecture:

Model

Defines database structure (Expense model)

View

Handles business logic and data processing

Template

Displays data using HTML and Bootstrap

1. DATABASE DESIGN

Expense Table

Field Name	Data Type	Description
id	Integer	Primary Key
user	Foreign Key	Linked to User table
title	CharField	Expense title
amount	Float	Expense amount
category	CharField	Expense category
date	DateField	Expense date

2. FUNCTIONAL MODULES

8.1 User Authentication

- Login required to access expenses
- Each user can see only their own data

8.2 Expense Management

- Add new expense
- View expenses
- Edit existing expenses
- Delete expenses

8.3 Expense Analytics

- Monthly total expense calculation
- Category-wise expense summary

8.4 Admin Module

- Admin can view and manage all expenses
- Admin panel customization

1. IMPLEMENTATION DETAILS

Django ORM is used for database operations

ModelForms are used for handling forms

Login protection is implemented using login_required

Bootstrap is used for responsive UI

Static files (CSS) are managed using Django static configuration

1. OUTPUT SCREENS (Description)

Login Page

Expense Dashboard

Add/Edit Expense Form

Monthly Expense Summary

Category-wise Expense Table

Admin Panel

1. ADVANTAGES OF THE SYSTEM

- Easy to use and understand
- Secure user-based data access
- Time-saving expense management
- Accurate expense tracking
- Responsive and clean UI

1. LIMITATIONS

- Uses SQLite (not ideal for large-scale production)
- No graphical charts
- No export feature (PDF/CSV)
- Single-user focus per login

1. FUTURE ENHANCEMENTS

- Add graphical charts using Chart.js
- Month and year filter
- Export expenses to PDF or Excel
- REST API using Django REST Framework
- Mobile application integration

1. CONCLUSION

The Expense Tracker Web Application successfully fulfills its objective of managing and analyzing personal expenses. The project demonstrates practical implementation of Django concepts such as authentication, CRUD operations, ORM, and template inheritance. This project provides a strong foundation for future enhancements and real-world applications.

1. REFERENCES

- Django Official Documentation
- Python Documentation
- Bootstrap Documentation