**Project Report**

**on**

**SpendWise**

Smart spending made simple

# Purpose & Objective

**Purpose:**

To develop a web-based expense tracking system using python - django framework that helps users monitor analyze and optimize their spending patterns.

**Objectives:**

* Provide secure user authentication
* Enable CRUD operations for expenses
* Categorize expenses for better analysis
* Visualize spending through charts and reports
* Offer mobile-responsive interface

# Features & Functionality

|  |  |
| --- | --- |
| Feature | Description |
| User Authentication | Secure login/logout with session management |
| Expense Management | Add/edit/delete expenses with categories |
| Dashboard | Summary of total spending and category breakdown |
| Data Visualization | Interactive charts using Chart.js |
| Responsive Design | Works on desktop, tablet, and mobile |
| Filtering | Filter expenses by date range and categories |

# Hardware & Software Specifications

**Hardware Requirements**

OS Windows 10

Processor Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz 2.30 GHz

Installed RAM 4.00 GB (Minimum 2.00 GB RAM)

System type 64-bit operating system, x64-based processor

**Software Requirements:**

Programming Language Python 3.8+

Web Framework Django 4.2

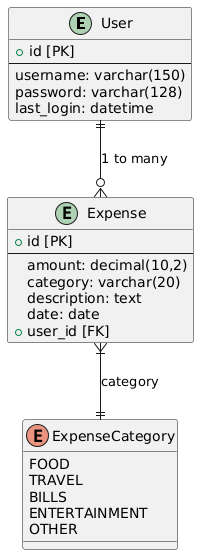
Database PostgreSQL/SQLite

Front End Bootstrap 5

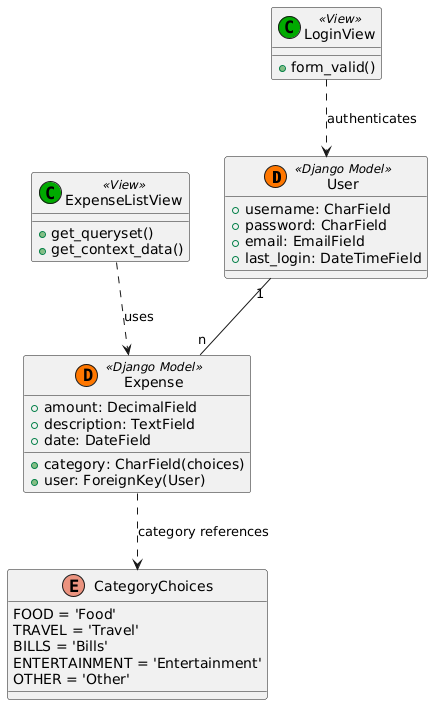
# Project Architecture

**Three-Tier Architecture:**

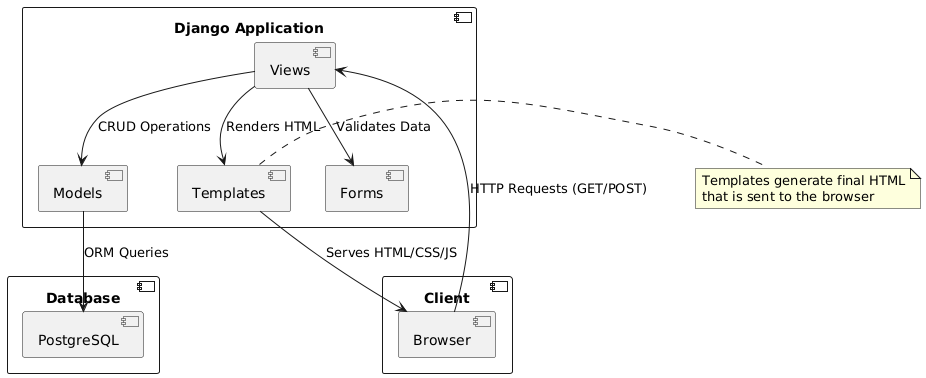
**E-R Diagram**



**Class Diagram**



**Component Diagram**



# Database Design

**User Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Parameters** | **Description** |
| Username | CharField | Max\_length=150,unique=True | User’s login identifier |
| Email | EmailField | Max\_length=254,unique=True | User’s contact email |
| Password | CharField | Max\_length =128 | Hashed password |
| Last\_loign | DateTimeField | Null=True,auto\_now=True | Session Tracking |

**Expense Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Parameters** | **Description** |
| User | ForeignKey | to=user, on\_delete=CASCADE | Owner of the expense |
| Amount | DecimalField | Max\_digits=10,decimal\_places=2 | Expense value |
| Category | CharField | Max\_length=20,choices=CATEGORIES | Spending category |
| Description | TextField | Blank=True,null=True | Optional notes |
| Date | DateField | Default=timezone.now | When expense occured |

**Category Choices**

|  |
| --- |
| **Categories** |
| Food |
| Travel |
| Bills |
| Entertainement |
| Other |

**Key Relationships**

|  |  |  |
| --- | --- | --- |
| **Relationship** | **Type** | **Description** |
| User ⇨ Expense | One to Many | One user can have many expenses |
| Expense ⇨ Category | Many to One | Each expense has one category |

# UI/UX Design

**Design Principles:**

The Expense Tracker app features a clean, intuitive UI/UX design built with Bootstrap 5 for responsiveness and Chart.js for data visualization. Key design principles include:

**Minimalist Interface:** Streamlined layout with clear typography and ample white space to reduce cognitive load.

**Mobile-First Approach:** Fully responsive design ensuring seamless use on all devices.

**Visual Hierarchy:** Color-coded expense categories and interactive charts for quick insights.

**User-Centric Workflow:** One-click actions (add/edit expenses), real-time feedback, and intuitive navigation.

**Accessibility**: High-contrast colors and ARIA labels for inclusive usability.

The design prioritizes speed and clarity, enabling users to log expenses in under 10 seconds while providing actionable financial insights through interactive dashboards.

# Deployment

**PythonAnywhere**

PythonAnywhere is a cloud-based platform specializing in Python web app hosting. Key features:

* Pre-configured Django environment
* Built-in database (MySQL/PostgreSQL) support
* Web-based console for management
* Automatic HTTPS with Let's Encrypt

**Git**

Git is a **distributed version control system** that:

* Tracks code changes with commit history
* Enables team collaboration
* Allows branching for feature development
* Integrates with platforms like GitHub/GitLab

**Why This Stack?**

**PythonAnywhere:** Simplifies Django deployment without server management

**Git:** Ensures code safety and version control

**PostgreSQL:** Robust database for financial data

**Deployment Steps:**

1. Push code to GitHub/GitLab

2. Create PythonAnywhere account

3. Configure virtual environment

4. Set up WSGI file

5. Migrate database

6. Collect static files

**Source Code:**

**Login.html**

{% extends 'base.html'%}

{% load static %}

{% block content %}

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-6 col-lg-4">

            <div class="card shadow-lg mt-5">

                <!-- Logo Section -->

                <div class="text-center mt-4">

                    <img src="{% static 'expenses/images/Logo.jpg' alt="Logo"%}"

                         alt="Expense Tracker Logo"

                         class="img-fluid"

                         style="max-height: 100px;">

                    <h2 class="mt-3">Expense Tracker</h2>

                </div>

                <!-- Login Form -->

                <div class="card-body p-4">

                    <form method="post">

                        {% csrf\_token %}

                        <div class="mb-3">

                            <label for="username" class="form-label">Username</label>

                            <input type="text"

                                   class="form-control"

                                   id="username"

                                   name="username"

                                   required>

                        </div>

                        <div class="mb-3">

                            <label for="password" class="form-label">Password</label>

                            <input type="password"

                                   class="form-control"

                                   id="password"

                                   name="password"

                                   required>

                        </div>

                        <button type="submit" class="btn btn-primary w-100">

                            <i class="fas fa-sign-in-alt me-2"></i> Login

                        </button>

                    </form>

                    <div class="text-center mt-3">

                        <p class="mb-0">Don't have an account?

                            <a href="{% url 'signup' %}">Sign Up</a>

                        </p>

                    </div>

                </div>

            </div>

        </div>

    </div>

</div>

{% endblock %}

**Add\_expense.html**

{% extends 'base.html' %}

{% block content %}

<div class="card shadow col-md-6 mx-auto">

    <div class="card-header bg-white">

        <h4 class="mb-0">{% if form.instance.id %}Edit{% else %}Add{% endif %} Expense</h4>

    </div>

    <div class="card-body">

        <form method="post">

            {% csrf\_token %}

            <div class="mb-3">

                {{ form.amount.label\_tag }}

                <div class="input-group">

                    <span class="input-group-text">$</span>

                    {{ form.amount }}

                </div>

            </div>

            <div class="mb-3">

                {{ form.category.label\_tag }}

                {{ form.category }}

            </div>

            <div class="mb-3">

                {{ form.description.label\_tag }}

                {{ form.description }}

            </div>

            <button type="submit" class="btn btn-primary w-100">

                <i class="fas fa-save me-2"></i>Save

            </button>

        </form>

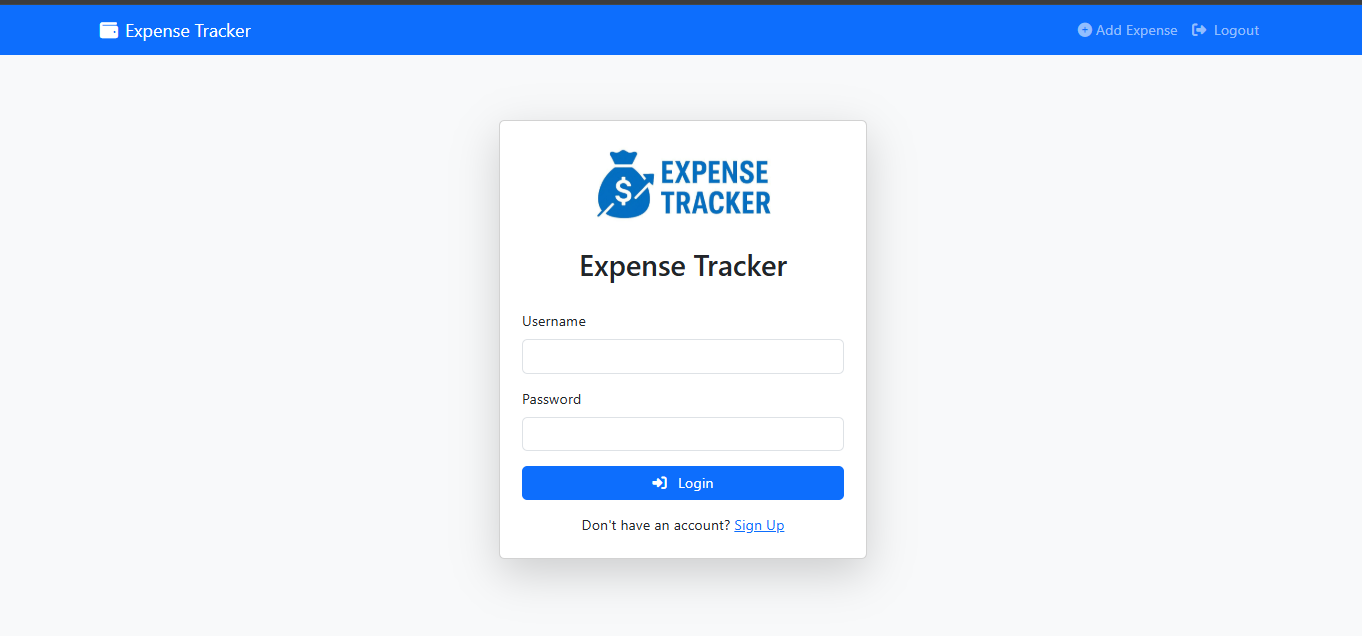
    </div>

</div>

{% endblock %}

# Output Screenshots:

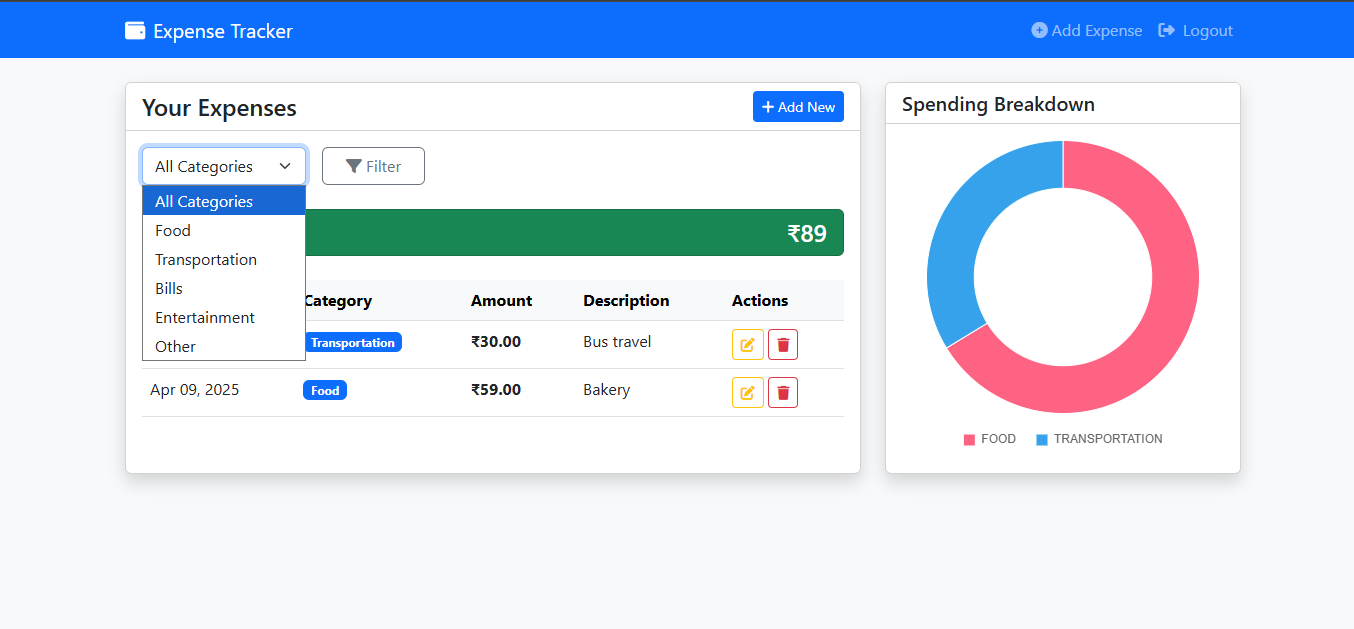
**1. Login/Registration**



**2. Dashboard with summary cards**A screenshot of a computer

AI-generated content may be incorrect.

3**. Expense list with filters**



**4. Add/Edit expense form**

A screenshot of a computer

AI-generated content may be incorrect.