**Project Title: Full Stack Expense Tracker**

1. Setting Up the Flask Application

from flask import Flask, render\_template, request, redirect, url\_for, session, flash

from flask\_sqlalchemy import SQLAlchemy

from datetime import datetime

from werkzeug.security import generate\_password\_hash, check\_password\_hash

import jwt

import os

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = 'your\_secret\_key'

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///expenses.db'

db = SQLAlchemy(app)

# Database Model

class User(db.Model):

id = db.Column(db.Integer, primary\_key=True)

username = db.Column(db.String(150), unique=True, nullable=False)

password = db.Column(db.String(150), nullable=False)

class Expense(db.Model):

id = db.Column(db.Integer, primary\_key=True)

category = db.Column(db.String(100), nullable=False)

amount = db.Column(db.Float, nullable=False)

comments = db.Column(db.String(200), nullable=True)

created\_at = db.Column(db.DateTime, default=datetime.utcnow)

# Routes

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/signup', methods=['GET', 'POST'])

def signup():

if request.method == 'POST':

username = request.form['username']

password = generate\_password\_hash(request.form['password'], method='sha256')

new\_user = User(username=username, password=password)

db.session.add(new\_user)

db.session.commit()

flash('Sign up successful! Please log in.', 'success')

return redirect(url\_for('login'))

return render\_template('signup.html')

@app.route('/login', methods=['GET', 'POST'])

def login():

if request.method == 'POST':

user = User.query.filter\_by(username=request.form['username']).first()

if user and check\_password\_hash(user.password, request.form['password']):

session['user\_id'] = user.id

flash('Login successful!', 'success')

return redirect(url\_for('dashboard'))

flash('Invalid credentials. Please try again.', 'danger')

return render\_template('login.html')

@app.route('/dashboard')

def dashboard():

if 'user\_id' not in session:

return redirect(url\_for('login'))

expenses = Expense.query.all()

return render\_template('dashboard.html', expenses=expenses)

@app.route('/add\_expense', methods=['POST'])

def add\_expense():

if 'user\_id' in session:

category = request.form['category']

amount = request.form['amount']

comments = request.form['comments']

new\_expense = Expense(category=category, amount=amount, comments=comments)

db.session.add(new\_expense)

db.session.commit()

flash('Expense added successfully!', 'success')

return redirect(url\_for('dashboard'))

@app.route('/edit\_expense/<int:expense\_id>', methods=['GET', 'POST'])

def edit\_expense(expense\_id):

expense = Expense.query.get(expense\_id)

if request.method == 'POST':

expense.category = request.form['category']

expense.amount = request.form['amount']

expense.comments = request.form['comments']

db.session.commit()

flash('Expense updated successfully!', 'success')

return redirect(url\_for('dashboard'))

return render\_template('edit\_expense.html', expense=expense)

@app.route('/delete\_expense/<int:expense\_id>', methods=['POST'])

def delete\_expense(expense\_id):

expense = Expense.query.get(expense\_id)

db.session.delete(expense)

db.session.commit()

flash('Expense deleted successfully!', 'success')

return redirect(url\_for('dashboard'))

@app.route('/logout')

def logout():

session.pop('user\_id', None)

flash('You have been logged out.', 'info')

return redirect(url\_for('index'))

if \_\_name\_\_ == '\_\_main\_\_':

if not os.path.exists('expenses.db'):

db.create\_all()

app.run(debug=True)

1. Database Models

from flask\_sqlalchemy import SQLAlchemy

from datetime import datetime

db = SQLAlchemy()

class User(db.Model):

id = db.Column(db.Integer, primary\_key=True)

username = db.Column(db.String(150), unique=True, nullable=False)

password = db.Column(db.String(150), nullable=False)

class Expense(db.Model):

id = db.Column(db.Integer, primary\_key=True)

category = db.Column(db.String(100), nullable=False)

amount = db.Column(db.Float, nullable=False)

comments = db.Column(db.String(200), nullable=True)

created\_at = db.Column(db.DateTime, default=datetime.utcnow)

1. Frontend HTML Templates

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

<title>Expense Tracker</title>

</head>

<body>

<h1>Welcome to Expense Tracker</h1>

<a href="/signup">Sign Up</a>

<a href="/login">Login</a>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

<title>Sign Up</title>

</head>

<body>

<h1>Sign Up</h1>

<form method="POST">

<input type="text" name="username" placeholder="Username" required>

<input type="password" name="password" placeholder="Password" required>

<button type="submit">Sign Up</button>

</form>

<a href="/login">Already have an account? Login</a>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

<title>Login</title>

</head>

<body>

<h1>Login</h1>

<form method="POST">

<input type="text" name="username" placeholder="Username" required>

<input type="password" name="password" placeholder="Password" required>

<button type="submit">Login</button>

</form>

<a href="/signup">Don't have an account? Sign Up</a>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<title>Dashboard</title>

</head>

<body>

<h1>Dashboard</h1>

<form action="/add\_expense" method="POST">

<input type="text" name="category" placeholder="Category" required>

<input type="number" name="amount" placeholder="Amount" required>

<input type="text" name="comments" placeholder="Comments (optional)">

<button type="submit">Add Expense</button>

</form>

<h2>Expenses</h2>

<table>

<tr>

<th>Category</th>

<th>Amount</th>

<th>Created At</th>

<th>Comments</th>

<th>Actions</th>

</tr>

{% for expense in expenses %}

<tr>

<td>{{ expense.category }}</td>

<td>{{ expense.amount }}</td>

<td>{{ expense.created\_at }}</td>

<td>{{ expense.comments }}</td>

<td>

<a href="/edit\_expense/{{ expense.id }}">Edit</a>

<form action="/delete\_expense/{{ expense.id }}" method="POST" style="display:inline;">

<button type="submit">Delete</button>

</form>

</td>

</tr>

{% endfor %}

</table>

<h2>Category-wise Expense Distribution</h2>

<canvas id="expenseChart"></canvas>

<script src="{{ url\_for('static', filename='chart.js') }}"></script>

<script>

const ctx = document.getElementById('expenseChart').getContext('2d');

const data = {

labels: [{% for expense in expenses %}'{{ expense.category }}',{% endfor %}],

datasets: [{

label: 'Expenses',

data: [{% for expense in expenses %}{{ expense.amount }},{% endfor %}],

backgroundColor: 'rgba(75, 192, 192, 0.6)',

borderColor: 'rgba(75, 192, 192, 1)',

borderWidth: 1

}]

};

4.Add Expense Component (AddExpense.js):

// frontend/src/components/AddExpense.js

import React, { useState } from 'react';

import axios from 'axios';

const AddExpense = ({ token }) => {

const [category, setCategory] = useState('');

const [amount, setAmount] = useState('');

const [comments, setComments] = useState('');

const handleSubmit = async (e) => {

e.preventDefault();

await axios.post('http://localhost:5000/expenses', { category, amount, comments }, {

headers: { Authorization: `Bearer ${token}` }

});

};

return (

<form onSubmit={handleSubmit}>

<input type="text" placeholder="Category" onChange={(e) => setCategory(e.target.value)} required />

<input type="number" placeholder="Amount" onChange={(e) => setAmount(e.target.value)} required />

<input type="text" placeholder="Comments" onChange={(e) => setComments(e.target.value)} />

<button type="submit">Add Expense</button>

</form>

);

};

export default AddExpense;

// frontend/src/components/ExpenseList.js

import React, { useEffect, useState } from 'react';

import axios from 'axios';

const ExpenseList = ({ token }) => {

const [expenses, setExpenses] = useState([]);

useEffect(() => {

const fetchExpenses = async () => {

const response = await axios.get('http://localhost:5000/expenses', {

headers: { Authorization: `Bearer ${token}` }

});

setExpenses(response.data);

};

fetchExpenses();

}, [token]);

return (

<table>

<thead>

<tr>

<th>Category</th>

<th>Amount</th>

<th>Comments</th>

<th>Created At</th>

</tr>

</thead>

<tbody>

{expenses.map((expense) => (

<tr key={expense.id}>

<td>{expense.category}</td>

<td>{expense.amount}</td>

<td>{expense.comments}</td>

<td>{expense.created\_at}</td>

</tr>

))}

</tbody>

</table>

);

};

export default ExpenseList;