

Disciplina: Desenvolvimento Rápido de Aplicações em Python

Data: 22/11/2024

Equipe: Rick Steven Santos Andrade, Renato Apolinário da Silva, Guilherme de Souza Aguzzoli, Rui Romer Cupertino Sacramento Junior e Daniel Carmo dos Santos.

Professor: Heleno Filho

Introdução

Manter o controle financeiro é fundamental para atingir metas econômicas e evitar imprevistos no orçamento. Para atender a essa necessidade, foi desenvolvida uma aplicação simples, com interface intuitiva e recursos voltados para o usuário, que facilita a organização financeira. Essa ferramenta é útil tanto para iniciantes quanto para aqueles que buscam gerenciar suas finanças de forma mais detalhada e eficiente.

Objetivo do Projeto

O **Controle de Finanças** é uma aplicação web simples, desenvolvida em Python com Flask, para auxiliar no gerenciamento financeiro pessoal. A aplicação permite registrar receitas e despesas, categorizando-as e exibindo relatórios gráficos para visualização clara e objetiva.

Recursos Oferecidos

- **Cadastro e Autenticação de Usuários**
 - Registro de novos usuários.
 - Login seguro utilizando e-mail e senha.
 - Recuperação de senha com palavra-chave personalizada.
- **Gerenciamento de Receitas e Despesas**
 - Registro de receitas e despesas com descrição, valor, data e tipo (para despesas).
 - Possibilidade de excluir receitas e despesas diretamente na interface.
- **Gráficos**
 - **Gráfico de barras horizontal:** Exibe as despesas por valor.
 - **Gráfico de barras horizontal:** Exibe as despesas categorizadas por tipo.
 - **Gráfico de rosca:** Compara o total de despesas e receitas.
- **Interface Intuitiva**
 - Interface em tema dark.
 - Botões de ação para cadastrar receitas e despesas.
 - Visualização de tabelas responsivas para listar receitas e despesas.

Estrutura do Projeto

- **Backend**
 - Framework: Flask
 - Banco de Dados: SQLite
 - Gerenciamento de Migrações: Flask-Migrate

- Criptografia de Senhas: Flask-Bcrypt
 - Controle de Sessões: Flask-Login
- **Frontend**
 - Framework CSS: Bootstrap
 - Visualização de Dados: Chart.js
- **Scripts Auxiliares**
 - seed.py: Popula o banco de dados com dados iniciais, como tipos de despesas.

Como Utilizar

1. Clone o projeto do repositório no GitHub.
2. Configure o ambiente virtual e instale as dependências.
3. Execute as migrações para configurar o banco de dados.
4. Inicie o servidor Flask e acesse a aplicação no navegador.
5. Crie uma conta, registre receitas e despesas, e visualize os relatórios.

Pré-requisitos

1. **Python 3.8 ou superior** instalado.
2. Ferramenta de gerenciamento de pacotes pip.
3. Sistema operacional com suporte a sqlite3 (Windows, Linux ou macOS).

Passo a Passo (para execução local)

1. Clonar o Repositório

Abra o terminal e execute:

```
git clone https://github.com/rickstevenandrade/controle-despesas.git
cd controle-despesas
```

2. Criar e Ativar um Ambiente Virtual

Crie um ambiente virtual Python para isolar as dependências do projeto:

```
python -m venv venv
```

Ative o ambiente virtual:

- **Windows:**
venv\Scripts\activate
- **macOS/Linux:**
source venv/bin/activate

3. Instalar Dependências

Com o ambiente virtual ativo, instale as dependências necessárias:

```
pip install -r requirements.txt
```

4. Configurar o Banco de Dados

Este projeto utiliza **Flask-Migrate** para gerenciar o banco de dados.

4.1. Executar as Migrações

Crie as tabelas necessárias no banco de dados SQLite:

```
flask db upgrade
```

4.2. Popular o Banco de Dados (Opcional)

Se desejar popular o banco de dados com dados iniciais (como tipos de despesas padrão), execute o script seed.py:

```
python seed.py
```

5. Executar a Aplicação

Inicie o servidor Flask:

```
flask run
```

Por padrão, o servidor será executado no endereço: <http://127.0.0.1:5000>

6. Navegar na Aplicação

1. Criar uma Conta:

- Registre-se na aplicação pela tela inicial de cadastro.

2. Gerenciar Receitas e Despesas:

- Cadastre receitas e despesas com descrições, valores e datas.
- Visualize as tabelas de receitas e despesas, com a opção de excluir qualquer registro.

3. Visualizar Gráficos:

- Acompanhe gráficos interativos:
 - **Gráfico de barras (despesas por valor):** Exibe as despesas registradas em ordem decrescente.
 - **Gráfico de barras (despesas por tipo):** Exibe os valores totais de cada tipo de despesa.
 - **Gráfico de rosca:** Compara o total de despesas e receitas.

Observações Finais

- Caso o banco de dados já tenha sido configurado anteriormente, não será necessário executar novamente as migrações.
- Certifique-se de usar um ambiente virtual Python para evitar conflitos de dependências com outros projetos.