

Lista 6

Gerado com IA

Exercício: Criando um Sistema de Login com Flask

Objetivo: Desenvolver um mecanismo de login que utilize sessões e cookies para gerenciar o acesso do usuário, com uma rota protegida (**dashboard**) e funcionalidade de logout.

Passo 1: Preparando o Ambiente

1. Instalação do Flask (se necessário):

- Crie e ative um ambiente virtual:

```
python -m venv venv
```

- Ative no Windows:

```
venv\Scripts\activate
```

- Ative no Linux/Mac:

```
source venv/bin/activate
```

- Instale o Flask:

```
pip install Flask
```

- 2. **Objetivo deste passo:** Garantir que o ambiente esteja pronto para o desenvolvimento com Flask.
-

Passo 2: Configurando o Arquivo da Aplicação

- 1. Crie um arquivo chamado **app.py** com a seguinte configuração inicial:

```
from flask import Flask, request, render_template, session, redirect, url_for, make_response

app = Flask(__name__)
app.secret_key = 'chave-secreta' # Necessário para gerenciar sessões
```

2. **Objetivo deste passo:** Configurar o projeto e preparar o Flask para lidar com sessões.
-

Passo 3: Criando os Templates

1. Template de Login (`templates/login.html`):

- Crie a pasta `templates` no mesmo diretório do `app.py`.
- Adicione o arquivo `login.html` com o seguinte conteúdo:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
</head>
<body>
  <h1>Faça Login</h1>
  <form action="{{ url_for('login') }}" method="post">
    <label for="username">Usuário:</label>
    <input type="text" id="username" name="username" required><br>
    <br>
    <label for="password">Senha:</label>
    <input type="password" id="password" name="password" required>
    <br><br>
    <button type="submit">Entrar</button>
  </form>
  {% if erro %}
    <p style="color:red;">{{ erro }}</p>
  {% endif %}
</body>
</html>
```

2. Template do Dashboard (`templates/dashboard.html`):

- Adicione o arquivo `dashboard.html`:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Dashboard</title>
</head>
<body>
  <h1>Bem-vindo, {{ username }}!</h1>
  <form action="{{ url_for('logout') }}" method="post">
    <button type="submit">Sair</button>
  </form>
```

```
</body>
</html>
```

3. **Objetivo deste passo:** Criar interfaces para login e o painel protegido.

Passo 4: Definindo as Rotas

1. No arquivo `app.py`, adicione as rotas para login, logout e o dashboard:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        # Usuário e senha pré-definidos
        if username == 'admin' and password == 'senha123':
            # Salvar usuário na sessão
            session['username'] = username
            # Criar um cookie para lembrar o usuário
            resposta = make_response(redirect(url_for('dashboard')))
            resposta.set_cookie('username', username, max_age=60*60*24) #
            Cookie válido por um dia
            return resposta
        else:
            return render_template('login.html', erro='Usuário ou senha
            inválidos.')

    # Verificar se o usuário já está logado
    if 'username' in session:
        return redirect(url_for('dashboard'))

    return render_template('login.html')

@app.route('/dashboard')
def dashboard():
    # Verificar se o usuário está na sessão
    username = session.get('username')
    if not username:
        return redirect(url_for('login')) # Redirecionar para o login se
    não estiver logado
    return render_template('dashboard.html', username=username)

@app.route('/logout', methods=['POST'])
def logout():
    # Remover usuário da sessão
    session.pop('username', None)
    # Remover o cookie
    resposta = make_response(redirect(url_for('login')))
    resposta.set_cookie('username', '', max_age=0) # Excluir o cookie
    return resposta
```

-
2. **Objetivo deste passo:** Implementar as funcionalidades de login, logout e proteger o acesso ao dashboard.
-

Passo 5: Testando a Aplicação

1. No terminal, inicie o servidor:

```
python app.py
```

2. Abra o navegador e siga os passos:

- Acesse <http://127.0.0.1:5000/login>.
- Realize o login com as credenciais:
 - Usuário: `admin`
 - Senha: `senha123`
- Após o login bem-sucedido, o usuário será redirecionado ao `dashboard`.
- Clique no botão "Sair" para realizar o logout e observe que a sessão e o cookie são removidos.

3. Tente acessar <http://127.0.0.1:5000/dashboard> após o logout e confirme que você é redirecionado para a página de login.
-

Conclusão do Exercício

Este exercício abordou:

1. Uso de sessões e cookies para gerenciar o login.
2. Redirecionamentos condicionais com `url_for` e `redirect`.
3. Proteção de rotas utilizando sessões.
4. Implementação de logout para limpar dados de sessão e cookies.

Desafio Extra: Modifique o código para suportar múltiplos usuários com suas respectivas senhas. Use um dicionário ou um banco de dados para validar as credenciais.