SENAI – SP

DOCUMENTAÇÃO DO PROJETO

THIMOTIO E. A. JERONIMO

Estrutura das Pastas:

**Pycache:** Armazena arquivos compilados pelo Python.

**Static:** Contém arquivos estáticos como imagens e o CSS

**Templates:** Guarda os templates HTML dinâmicos, como index.html, login.html, e páginas específicas como adm.html e erro404.html.

**App.Py:** Provavelmente o arquivo principal, definindo rotas e lógica da aplicação.

**Config.Py:** Contém configurações do projeto.

**Database.SQL:** Script SQL do banco de dados.

Bibliotecas:

As bibliotecas são:

* Flask
* Render\_template
* Request
* Redirect
* Session

Arquivos Principais:

Os arquivos principais são:

* O **app.py**, sendo considerado o coração da aplicação do flask
* O **config.py**, contendo as configurações do projeto
* O **Banco de Dados**, que contém o script SQL do projeto
* A pasta **templates**, possuindo os arquivos html, renderizados pelo flask
* A pasta **static**, obtendo os recursos estáticos (imagens, o css, ícones)
* A **chave secreta**, fundamental para gerenciar sessões seguras e proteger a aplicação contra ataques como falsificação de solicitação entre sites.

Rotas e o que faz cada uma:

* **Login:** Renderiza o template da página de login (login.html)
* **Logout:** Limpa a sessão do usuário e redireciona para a página inicial
* **/:** Exibe a página principal (index.html) com uma lista de posts. Os posts são carregados do banco de dados e formatados antes de serem enviados ao template. Também verifica se o usuário está logado.
* **Acesso:** Valida as credenciais do usuário ou do administrador. Se for o admin, ele é redirecionado para /adm. Se for um usuário normal, ele é redirecionado para a página principal (/). Se as credenciais estiverem erradas, retorna ao login com uma mensagem de erro.
* **Novopost:** Renderiza a página de criação de um novo post (novopost.html) para usuários logados. Se o usuário não estiver logado, ele é redirecionado para o login.
* **Cadastro\_post:** Recebe os dados do formulário de criação de um post, insere no banco de dados e redireciona para a página inicial.
* **Adm:** Verifica se o usuário é o administrador (com sessão válida). Se for, renderiza a página de administração (adm.html), listando todos os usuários e posts.
* **Novousuario:** Renderiza a página para cadastro de um novo usuário (novousuario.html). Somente o administrador tem acesso a esta página.
* **Cadastro\_usuario:** Recebe os dados do formulário de cadastro de usuário e insere no banco de dados. Se o email já existir, retorna um erro. Somente o administrador pode acessar esta rota.
* **Editar-user:** Carrega os dados do usuário com base no id e renderiza o template de edição de usuário (editarusuario.html). Somente o administrador pode acessar esta rota.
* **Atualizar\_usuario:** Atualiza os dados de um usuário no banco de dados com base no formulário submetido. Somente o administrador pode acessar esta rota.
* **Excluir-user:** Exclui um usuário e seus posts com base no id passado. Somente o administrador pode excluir usuários.
* **Excluir-post:** Exclui um post do banco de dados com base no id do post. O administrador pode excluir qualquer post. Um usuário comum só pode excluir seus próprios posts.
* **Erro-404:** Renderiza a página de erro 404 (erro404.html) quando uma rota não encontrada é acessada.

DOCUMENTO SOBRE O BLOG

PYTHON

* O app.py concentra todos os códigos python para o funcionamento
* O config.py traz as informações necessárias como endpoints, senhas e outras informações de configuração

TEMPLATE

* A pasta templates são armazenadas as páginas HTML, renderizadas de forma dinâmicas.
* A página modelo será o template padrão e as outras irão incorporá-la na renderização.
* As páginas contêm códigos (jinja2) para torná-las dinâmicas {% %} {{ }}

STATIC

* A pasta static armazena os arquivos estáticos, ou seja, que não são alterados
* São arquivos importantes para o funcionamento do blog (imagens, css, java, frameworks, bootstrap, etc)

MySQL

* O MySQL é uma biblioteca que permite conectar aplicativos python a banco de dados MySQL. Podendo executar operações como inserir, atualizar, deletar e consultar dados no BD a partir de scripts python.