Manual de Comandos - Integração Flask, HTML, CSS e SQLite

* from flask import Flask

Importa a classe Flask para criar a aplicação web.

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

Cria a aplicação Flask usando o nome do módulo atual.

* app.config['SECRET\_KEY']

Define a chave secreta para gerenciar sessões do Flask.

* app.config['SQLALCHEMY\_DATABASE\_URI']

Configura o caminho do banco de dados SQLite.

* app.config['UPLOAD\_FOLDER']

Define a pasta onde os arquivos enviados serão salvos.

* app.config['MAX\_CONTENT\_LENGTH']

Define o tamanho máximo de arquivos que podem ser enviados (em bytes).

* from flask\_sqlalchemy import SQLAlchemy

Importa o SQLAlchemy para ORM com banco de dados.

* db = SQLAlchemy(app)

Inicializa o banco de dados vinculado à aplicação Flask.

* db.Column()

Define uma coluna dentro de um modelo de tabela (ORM).

* db.create\_all()

Cria todas as tabelas do banco de dados a partir dos modelos definidos.

* db.session.add()

Adiciona um novo objeto ao banco de dados.

* db.session.commit()

Confirma e salva as alterações feitas no banco de dados.

* db.session.delete()

Remove um objeto do banco de dados.

* User.query.all()

Busca todos os registros da tabela User.

* User.query.get(id)

Busca um usuário pelo ID.

* User.query.filter\_by(campo=valor).first()

Filtra registros com base em um campo e retorna o primeiro encontrado.

* from flask import request

Importa o objeto request para capturar dados de formulários.

* request.form['campo']

Pega o valor de um campo enviado via POST.

* request.files['arquivo']

Pega o arquivo enviado via formulário.

* from flask import render\_template

Permite renderizar arquivos HTML armazenados na pasta templates.

* render\_template('arquivo.html')

Carrega e exibe um template HTML.

* from flask import redirect

Permite redirecionar o usuário para outra rota.

* redirect(url\_for('rota'))

Redireciona o usuário para uma função Flask específica.

* from flask import url\_for

Gera a URL para uma rota Flask definida.

* from flask import session

Permite o uso de sessões (armazenar dados temporários do usuário).

* session['chave']

Armazena dados temporários identificados por uma chave.

* session.clear()

Remove todos os dados da sessão atual.

* if 'chave' in session

Verifica se a chave existe na sessão.

* from werkzeug.utils import secure\_filename

Garante que o nome do arquivo enviado não cause falhas de segurança.

* os.path.join()

Une diretórios e nomes de arquivos corretamente conforme o sistema operacional.

* os.makedirs(pasta, exist\_ok=True)

Cria uma pasta, se ela ainda não existir.

* @app.route('/rota')

Define uma rota (URL) e vincula a uma função Python que será executada.

* methods=['GET', 'POST']

Permite que a rota aceite requisições GET e POST.

* if request.method == 'POST'

Verifica se o formulário foi enviado com o método POST.