

****Projeto de Formulário Web com Node.js, Express, SQLite e Estilização CSS****

****Objetivo:****

Desenvolver uma aplicação web simples que inclua um formulário de contato. A aplicação deve ser construída utilizando Node.js, o framework Express para o lado do servidor, SQLite para armazenamento de dados e estilização CSS para uma apresentação atraente.

****Requisitos Funcionais:****

1. **Servidor Node.js:**

- Crie um arquivo `app.js` para configurar um servidor Node.js.
- Utilize o framework Express para gerenciar rotas e requisições HTTP.
- Configure o servidor para escutar na porta 3055.

2. **Formulário de Contato:**

- Crie um arquivo HTML chamado `index.html` com um formulário contendo campos para nome, e-mail e mensagem.
- Configure o formulário para enviar dados para o servidor via método POST para a rota `/submit`.
- Adicione campos obrigatórios no formulário.
- Inclua uma página de confirmação (`confirmacao.html`) que seja exibida após o envio bem-sucedido do formulário.

3. **Estilização CSS:**

- Crie um arquivo CSS chamado `styles.css` para estilizar a aplicação.
- Adicione estilos globais, incluindo a definição de fontes e cores de fundo.
- Estilize o cabeçalho, conteúdo principal, rodapé e formulário de maneira atraente.
- Implemente um estilo de botão interativo.

4. **Banco de Dados SQLite:**

- Utilize o módulo `sqlite3` para interagir com um banco de dados SQLite.
- Configure a aplicação para se conectar ao banco de dados `database.db`.
- Crie uma tabela chamada `contatos` com campos para id, nome, e-mail e mensagem. Garanta que a tabela seja criada apenas se não existir.

5. **Lógica do Servidor:**

- Ao receber uma requisição POST na rota `/submit`, extraia os dados do formulário e armazene-os na tabela `contatos` do banco de dados SQLite.
- Exiba as informações recebidas no console para verificação.
- Redirecione o usuário para a página de confirmação após o envio bem-sucedido do formulário.

6. **Página de Confirmação:**

- Na página `confirmacao.html`, exiba uma mensagem indicando que o formulário foi enviado com sucesso.
- Inclua um link para a página inicial.

****Observações:****

- Considere utilizar o middleware `body-parser` para facilitar o processamento dos dados do formulário.

- Adicione estilos de sua escolha para criar uma apresentação visual atraente.
- Certifique-se de lidar adequadamente com validação no lado do servidor.
- Teste a aplicação para garantir que o formulário seja enviado corretamente, os dados sejam armazenados no banco de dados e a página de confirmação seja exibida.

****Entrega:****

Envie os arquivos do projeto organizados em uma estrutura de diretórios clara, incluindo todos os arquivos necessários para a execução da aplicação. Documente qualquer instrução adicional necessária para configurar e executar a aplicação.

Observações:

****1) app.js:****

Este arquivo é o ponto central da aplicação Node.js. Aqui estão algumas das principais funcionalidades:

- ****Express Configuration:**** Importa o framework Express, configura um servidor e define a porta (3055) para escutar as requisições.
- ****Middleware Configuration:**** Utiliza o middleware `body-parser` para analisar os corpos das requisições, facilitando o acesso aos dados enviados via formulário.
- ****SQLite Database Connection:**** Conecta-se a um banco de dados SQLite chamado 'database.db' e cria uma tabela chamada 'contatos' se ela não existir.
- ****Static Files:**** Configura o Express para servir arquivos estáticos do diretório atual (__dirname).
- ****Rotas:**** Define rotas para a página inicial, formulário de submissão, e página de confirmação. Quando o formulário é submetido, os dados são inseridos no banco de dados e o usuário é redirecionado para a página de confirmação.
- ****Servidor:**** Inicia o servidor na porta especificada.

****2) index.html:****

Este arquivo representa a estrutura HTML da página inicial. Algumas características incluem:

- ****Metatags e Título:**** Configura metatags para caracteres especiais e a viewport, além de definir o título da página.
- ****Links para Estilos:**** Referencia o arquivo 'styles.css' para a estilização da página.
- ****Cabeçalho, Conteúdo Principal e Rodapé:**** Organiza a estrutura da página com seções que destacam as informações sobre o curso de desenvolvimento frontend.
- ****Formulário de Contato:**** Inclui um formulário com campos para nome, e-mail e mensagem, configurado para ser enviado para a rota '/submit' via método POST.

****3) styles.css:****

Este arquivo contém estilos CSS para a aplicação. Algumas características notáveis são:

- ****Estilo Global:**** Define estilos para o corpo da página, contêineres e elementos gerais.
- ****Estilos do Cabeçalho, Conteúdo e Rodapé:**** Configura estilos específicos para as seções do cabeçalho, conteúdo principal e rodapé.
- ****Estilos do Formulário:**** Define estilos para o formulário, rótulos, campos de entrada e botão de envio.
- ****Estilos de Botão:**** Adiciona estilos específicos para botões, incluindo uma animação de transição ao passar o mouse sobre o botão.

****4) confirmacao.html:****

Este arquivo representa a página de confirmação exibida após o envio bem-sucedido do formulário. Algumas características são:

- ****Estilos embutidos:**** Os estilos são incorporados diretamente no cabeçalho da página usando a tag `<style>`.
- ****Mensagem de Confirmação:**** Exibe uma mensagem indicando que o formulário foi enviado com sucesso.
- ****Link para a Página Inicial:**** Inclui um link para a página inicial, permitindo que o usuário retorne facilmente.

Esses arquivos trabalham em conjunto para criar uma aplicação simples de formulário web usando Node.js, Express, SQLite e estilização CSS. O arquivo `app.js` lida com a lógica do servidor, enquanto os arquivos HTML fornecem a estrutura da página e os estilos CSS cuidam da apresentação visual. A página de confirmação é exibida após o envio bem-sucedido do formulário.