Projeto Final do 1º Semestre - PW II

o por Rafael SP



Critérios de Avaliação

Orientações Iniciais para o Projeto:

Objetivo do Projeto:

O objetivo deste projeto é desenvolver uma aplicação web com funcionalidades de **CRUD** (**Create, Read, Update, Delete**) utilizando **PHP** e **SQL**. A aplicação deve permitir a criação, leitura, atualização e exclusão de dados de um banco de dados, com uma interface amigável e moderna, priorizando a experiência do usuário.

Tecnologias a serem usadas:

- **Frontend:** HTML, CSS e JavaScript (opcional, mas recomendado para melhorar a interação).
- Backend: PHP (para lógica de servidor e integração com o banco de dados).
- o Banco de Dados: MySQL (ou outro banco de dados relacional).
- Design: O design deve ser limpo, moderno e responsivo. Pode-se utilizar frameworks como Bootstrap ou outros para facilitar a criação de uma interface agradável.

Dica → consulte referências de design nos seguintes sites:

CSS Winner CSS Awards Dribbble

Funcionalidades obrigatórias:

- o Cadastro de novos registros.
- Exibição dos registros cadastrados.
- o Edição de registros existentes.
- o Exclusão de registros.
- Validação de dados no lado do servidor (PHP).
- Conexão com o banco de dados (SQL) e execução das operações CRUD.
- Uso de formulários adequados para cada operação (inserção, edição, exclusão).

Design e Usabilidade:

- o A interface deve ser moderna, intuitiva e fácil de usar.
- o O design deve ser responsivo, ou seja, funcionar bem tanto em dispositivos móveis quanto em desktop.
- o Utilização de cores, tipografia e espaçamento adequados para criar uma experiência visual agradável.
- As interações de CRUD devem ser claras e simples. Mensagens de sucesso ou erro devem ser exibidas de forma visível e clara para o usuário.

Segurança:

- Utilização de práticas de segurança como validação de entradas do usuário (para prevenir SQL Injection).
- Senhas devem ser armazenadas de maneira segura no banco de dados (ex: uso de password_hash() no PHP).
- A aplicação deve garantir que apenas usuários autorizados possam editar ou excluir registros.

Critérios de Avaliação:

Funcionalidade (30%)

- o O sistema implementa corretamente as operações CRUD: criar, ler, editar e excluir registros.
- A aplicação está integrada corretamente com o banco de dados (MySQL ou outro escolhido).
- As validações no lado do servidor estão funcionando adequadamente.

2. Design e Usabilidade (25%)

- o O design é moderno, limpo e funcional.
- A aplicação é responsiva, funcionando bem tanto em dispositivos móveis quanto em desktop.
- A experiência do usuário (UX) é fluida e intuitiva.

3. **Segurança (20%)**

- O código implementa boas práticas de segurança (ex: prevenção de SQL Injection, uso de hash para senhas, validação de dados de entrada).
- A aplicação é segura contra vulnerabilidades comuns.

4. Organização e Qualidade do Código (25%)

- o O código está bem estruturado e organizado, facilitando a manutenção.
- Uso de boas práticas de codificação (nomes de variáveis e funções claros, código modular e reutilizável).
- o O código é comentado adequadamente para explicar sua lógica.

Dicas e Recomendações:

• **Organização do Projeto:** Organize o projeto em pastas separadas para o frontend, backend e banco de dados. Exemplo de estrutura:

```
/project
/assets
/css
/js
/includes
/db_connection.php
/views
/home.php
/edit.php
/controllers
/create.php
/update.php
/delete.php
/models
/User.php
```

- Validação de Formulários: Certifique-se de validar os dados tanto no frontend (JavaScript) quanto no backend (PHP). A
 validação do lado do servidor é essencial para evitar falhas de segurança.
- Boas Práticas de Design: Use o conceito de "Mobile First" ao criar o layout, garantindo que o design funcione bem em dispositivos móveis antes de adaptá-lo para telas maiores.
- Interação com o Usuário: Adicione feedback visual ao usuário após cada operação (como uma mensagem de confirmação ou erro ao adicionar, editar ou excluir dados).
- **Revisão de Código:** Encoraje os alunos a revisarem o código uns dos outros. Isso pode ajudar a identificar problemas e melhorar a qualidade do código final.