**Documentação do Sistema**

**1. Introdução**

* **Nome do aluno(ª):** Ramon Santana
* **matrícula:** XXXXX
* **Nome do projeto:** Sistema de Checklist Operacional
* **Objetivo:** O sistema tem como objetivo digitalizar e gerenciar checklists de abertura e fechamento de estabelecimentos comerciais. Ele resolve o problema dos registros manuais em papel, fornecendo uma plataforma web para o preenchimento, armazenamento e consulta de dados operacionais de forma centralizada e auditável.
* **Público-alvo:** O público-alvo são os funcionários de uma empresa responsáveis por tarefas de abertura e fechamento, como líderes de turno e fiscais.

**2. Requisitos do Sistema**

**2.1 Funcionais**

* [x] O sistema deve permitir CRUD das entidades principais (registros de checklist).
* [x] O sistema deve armazenar dados em um banco de dados.
* [ ] O sistema deve permitir autenticação (login e senha).
* [ ] O sistema deve se conectar a uma API externa para obter informações.

**2.2 Não Funcionais**

* **Desenvolvido em:** PHP e JavaScript.
* **Banco de dados utilizado:** MySQL.
* **Frameworks/Bibliotecas:** Bootstrap 5.
* **O sistema deve ser simples, intuitivo e de fácil uso.**

**3. Funcionalidades do Sistema**

* **Gerenciamento de Checklists:** Permite realizar um CRUD completo (cadastrar, listar, editar e excluir) dos registros de abertura e fechamento.
* **Captura de Localização:** Utiliza a API de geolocalização do navegador para registrar as coordenadas GPS do local de preenchimento do checklist.
* **Preenchimento Dinâmico:** O formulário é pré-preenchido com a data atual e pode ser usado para criar novos registros ou editar existentes.

**4. Tecnologias Utilizadas**

* **Linguagem de programação:** PHP e JavaScript.
* **Banco de dados:** MySQL.
* **APIs utilizadas:** Geolocation API (nativa do navegador).
* **Outras ferramentas:** Bootstrap 5, Git, VS Code.

**5. Instruções de Execução**

1. **Fazer o download/clonar o repositório do projeto.**
2. **Importar o banco de dados:** Criar um banco de dados chamado checklist\_db e executar o script SQL de criação da tabela registros\_checklist.
3. **Configurar as variáveis de conexão:** No arquivo db\_connection.php, ajustar as variáveis $servidor, $usuario, $senha e $banco conforme as configurações do seu ambiente.
4. **Executar o projeto:** Iniciar os serviços Apache e MySQL no XAMPP.
5. **Acessar no navegador:** http://localhost/nome-da-pasta-do-projeto/index.php

**6. Conclusão**

* **Breve resumo do que foi aprendido:** O desenvolvimento deste sistema demonstrou a evolução de um projeto web, partindo de uma solução estática em HTML para uma aplicação dinâmica com back-end PHP, banco de dados MySQL e funcionalidades CRUD completas. Foi possível aplicar conceitos de boas práticas, como a separação de lógica e apresentação, o uso de prepared statements para segurança e a criação de componentes reutilizáveis.
* **Possíveis melhorias futuras que poderiam ser implementadas:**
  + Implementar um sistema de autenticação de usuários (login e senha).
  + Adicionar diferentes níveis de permissão (ex: administrador, funcionário).
  + Criar uma funcionalidade para gerar relatórios em PDF ou CSV.
  + Desenvolver um dashboard visual com gráficos e estatísticas sobre os checklists.