Universidade de Évora - Colégio Luís António Verney | Ano Letivo: 2023/2024
Tecnologias Web | Engenharia Informática
Alunos: Fidelis Silva | Nº 54761 e Enoque Masasu | Nº 53235

RELATÓRIO
Partilha de transporte / Boleias

INTRODUÇÃO:

Este documento aborda todos os passos para o desenvolvimento de uma interface web que permite visualizar as viagens
Relativamente aos objetivos do trabalho, são apontados como principais:

→ A listagem de todas as viagens já registradas;
→ Filtros aplicados com base em Local ou Data;
→ A listagem de todos os pedidos de viagem feitos;
→ Juntar-se ou deixar uma viagem a qual fez um pedido;
→ A procura de viagens com base na origem, destino ou data;
→ Fazer um pedido de viagem, adicionar uma viagem ou deletar os mesmos;
→ Obter o histórico de viagens de passageiros e motoristas, assim como sua avaliação.

HTML:
O projeto apresenta 3 ficheiros HTML, em que cada um tem um conjunto de funções diferentes para a interface.

→**main.html**: A home page onde são feitas as pesquisas de viagens. Além de ser possível pesquisar, também é possível fazer um pedido para a viagem em específico e então juntar-se/deixar a mesma. Tem como fonte o main.css e main.js

→**trips.html:** Página em que é possível ver as listas de viagens programadas e os pedidos de viagens. No caso da lista de viagens, também é possível aplicar um filtro por local e data, para facilitar a pesquisa. Tem como fonte o trips.css e trips.js

→**history.html:** Aqui é possível verificar o histórico de viagens de motoristas e condutores. Assim como sua classificação e o número total de viagens que já participou. Tem como fonte o history.css e history.js.

JAVASCRIPT:
Nesta seção falaremos das principais funções e suas descrições, entre elas estão:

- Busca de Dados:
  → fetchData: Faz uma requisição à API para obter informações de condutores ou passageiros e processa os resultados.
  → listarViagens: Requisita dados de viagens da API e renderiza uma tabela paginada.
  → listarPedidos: Faz o mesmo para os pedidos, exibindo-os em uma tabela paginada.

→ buscarViagens(origem, destino, data, callback): Busca viagens com base nos critérios fornecidos e retorna os resultados filtrados.
→ aplicarFiltros(viagens, origem, destino, data): Filtra os dados das viagens de acordo com os critérios de origem, destino e data.

- Exibição de Resultados:
  → displayData: Exibe os resultados da API (classificação, número de viagens, detalhes) em um contêiner.
  → renderizarTabelaViagens: Exibe uma tabela paginada de viagens, com filtros e navegação entre páginas.
  → renderizarTabela: Exibe uma tabela de pedidos paginada e com navegação.
  → mostrarViagens(viagens): Exibe viagens filtradas ou uma mensagem caso nenhuma viagem corresponda aos critérios.
  → criarElementoViagem(viagem): Cria elementos HTML para exibir detalhes de viagens.

- Manipulação de Associações
  → window.associarViagem: Associa um usuário à viagem indicada.
  → window.cancelarViagem: Cancela a associação do usuário à viagem especificada.
  → updateButtonToJoin(viagemId): Atualiza o botão para permitir associação à viagem.
  → updateButtonToLeave(viagemId): Atualiza o botão para permitir cancelamento da associação.

- Registro e Remoção
  → registarPedido: Registra um pedido associado a uma viagem específica.
  → registarViagem(): Registra uma nova viagem na API com as informações fornecidas pelo usuário.
  → window.removerViagem: Remove uma viagem específica com base no ID fornecido.
  → removerPedido(): Remove um pedido específico associado a um passageiro.

- Manipulação de Eventos
  → DOMContentLoaded: Garante que o DOM esteja carregado antes de executar o script.
  → Eventos de Botões: Liga botões como "Remover Pedido", "Registrar Viagem" e "Remover Viagem" às suas respectivas funções.
  → searchForm.addEventListener('submit', ...): Captura o envio do formulário de busca de viagens.

- Utilitários
  → exibirMensagemErro: Exibe mensagens de erro em um contêiner designado.
  → formatDate(dateString): Formata datas para facilitar comparações.