### 1. Conceito do Projeto

- Título do Projeto: Plataforma de experiências Turísticas

- Descrição: Criar um site que permite que diferentes empresas de turismo adicionem as suas tours, com detalhes como destino, descrição, preços e localização. Os clientes podem depois explorar os tours e reservar diretamente no site.

### 2. Estrutura do Site

- Página Inicial:

> Exibe os tours numa lista com filtros e opções de ordenação e poderá ter um mecanismo de procura para encontrar tours por destino, nome ou outros critérios.

- Página de Detalhes do Tour:

> Exibe informações detalhadas sobre um tour específico, incluindo fotos, descrição, itinerário, preço, localização, etc.

- Página de Submissão de Tour:

> Permite que várias empresas (ou gestores do site) de turismo adicionem tours, com validação de dados e autenticação.

### 3. Tecnologias a Usar

- Frontend:

> HTML, CSS, JavaScript em React.

- Backend:

> NodeJS com ExpressJS para o servidor.

- \*\*Base de Dados\*\*:

> MySQL

Ou ( qual usar?)

> MongoDB

- Autentificação:

> semelhante ao exercício da aula com login e registo de novo utilizador

### 4. Design

- Layout Responsivo

- Estética:

> Cores simples e consistentes em todo o site.

### 1. Arquitetura do Sistema

A arquitetura deve ser clara e permitir escalabilidade conforme o projeto cresce. Considere dividir a aplicação em camadas, como:

- \*\*Frontend\*\*:

- Framework de JavaScript para uma experiência de usuário rica (React,

- Comunicação com o backend por meio de APIs REST.

- Uso de CSS

- \*\*Backend\*\*:

- Servidor NodeJS com ExpressJS para roteamento e manipulação de pedidos HTTP.

- Interação com banco de dados e autenticação.

- \*\*Base de Dados\*\*:

- Estrutura relacional (MySQL) para dados como detalhes dos tours e informações de usuários.

- \*\*Autenticação e Autorização\*\*:

- JWT para autenticação segura.

- Controle de permissões para usuários, empresas e administradores.

### 2. Fluxos de Trabalho

Vamos delinear os principais fluxos de trabalho do sistema, para entender como os diferentes componentes interagem:

- \*\*Cadastro de Usuário\*\*:

- Usuário pode se registrar e criar uma conta.

- Empresas podem criar perfis para submeter tours.

- Autenticação usando JWT.

- \*\*Submissão de Tours\*\*:

- Empresas submetem detalhes do tour por meio de formulários.

- Validação de dados para garantir qualidade.

- Notificação para administradores para aprovação.

- \*\*Busca e Filtragem de Tours\*\*:

- Usuários buscam tours por destino, data, preço, etc.

- Filtros avançados para facilitar a descoberta de tours.

- \*\*Reserva de Tours\*\*:

- Usuários podem reservar tours.

- Integração de pagamento (PayPal, Stripe) para transações seguras.

- Confirmação por e-mail após a reserva.

- \*\*Administração\*\*:

- Administradores gerenciam tours e usuários.

- Aprovação ou rejeição de tours submetidos.

- Painel para visualizar estatísticas do sistema.

### 3. Detalhes Técnicos para Implementação

Aqui estão alguns aspectos técnicos que você pode considerar ao implementar o projeto:

- \*\*Roteamento no Frontend\*\*:

- Uso de React Router ou equivalente para gerenciar rotas no frontend.

- URLs amigáveis para SEO.

- \*\*Consumo de APIs\*\*:

- Fetch API ou Axios para comunicação entre frontend e backend.

- Gerenciamento de estados no frontend (Redux, Vuex, etc.).

- \*\*Lógica do Backend\*\*:

- Roteamento de pedidos HTTP com ExpressJS.

- Manipulação de lógica de negócios no Laravel.

- Implementação de APIs RESTful para comunicação entre frontend e backend.

- \*\*Segurança\*\*:

- Uso de JWT para autenticação segura.

- Proteção contra ataques comuns (ex: XSS, SQL Injection).

- \*\*Banco de Dados\*\*:

- Estrutura de tabelas para MySQL (ex: usuários, tours, reservas).

- Utilização do MongoDB para dados não-estruturados ou flexíveis.

### 4. Recursos Extras

- \*\*Documentação do Código\*\*:

- Certifique-se de documentar as partes críticas do código para facilitar a manutenção futura.

- Incluir instruções para instalação e configuração.

- \*\*Testes Automatizados\*\*:

- Implementar testes para garantir a estabilidade do sistema.

- Testes unitários para componentes individuais e testes de integração para fluxos completos.

Com essas diretrizes adicionais, você deve ter um plano mais detalhado para o seu projeto. Se precisar de mais informações ou dicas para aspectos específicos, estarei por aqui para ajudar!