

RESPONSIVE WEB DEVELOPMENT

A atividade proposta envolve a criação de uma aplicação web, utilizando o conceito de SPA (Single Page Application), com base no problema acima exposto, utilizando REACT + TYPESCRIPT + NEXTJS + API JAVAWEB como parte da solução do problema proposto.

OBJETIVO

Os alunos devem criar seus projetos dentro das especificações já apresentadas em aula, ou seja, utilizando como padrão o boiler-plate do NEXT.js, em sua última versão, sem nenhum arquivo e ou traço do projeto padrão.

RESPONSIVE WEB DEVELOPMENT

INSTRUÇÕES

Você deve iniciar um novo projeto. Não serão aceitos projetos construídos a partir dos exemplos utilizados em aula, modelos diferentes dos apresentados, templates existentes e claramente criados a partir de plataformas de terceiros. No máximo dependências como bibliotecas de ícones e ou fontes, outras bibliotecas como accordions, carroussel e outras desta natureza, não serão aceitas. Todo o desenvolvimento deve se basear no conhecimento adquirido pelos alunos.

Neste novo projeto obrigatoriamente deve conter os componentes (Server-Componentes/Clients-Components e Route Handlers-endpoints - APIs locais) em NEXT.js, para a exibição dos dados gerados pela API JAWAWE (backend em JAVA).

RESPONSIVE WEB DEVELOPMENT

INSTRUÇÕES

Para a entrega considerar que você terá uma estrutura para cadastro (por exemplo) e uma para login de usuários (por exemplo) e no mínimo mais duas estruturas que consumirão os dados cadastrados, esses dados serão retornados ao endpoint em formato JSON, essas estruturas deverão ser criadas na disciplinas **Domain Driven Design** e deve ter seu resultado exibido na tela do navegador através dos endpoints consumidos na disciplina de *Responsive Web Development*.

Para garantir a clareza e a transparência no acesso ao projeto e na documentação, é fundamental incluir todas as informações necessárias no README.MD do repositório do projeto no GitHub. Isso inclui fornecer todos os usuários e senhas necessários para acessar o projeto, links para o repositório do projeto no GitHub e um vídeo demonstrativo hospedado no YouTube, com o link fornecido no README.MD. A falta dessas informações será considerado um erro e resultará em uma penalidade de 2 pontos do total. Além disso, a parte do projeto que requer acesso a esses dados para análise não será considerada na avaliação.

RESPONSIVE WEB DEVELOPMENT

INSTRUÇÕES

A utilização correta de componentes do lado cliente e do lado servidor, como `useState`, `useEffect`, `redirects`, etc., é crucial para a eficiência e a performance da aplicação. Isso inclui a criação de APIs locais com rotas dinâmicas para otimização do tráfego de dados e gerenciamento. É essencial fazer com que o tráfego de dados passe pelas APIs locais do `NEXT.js`, juntamente com as Rotas Dinâmicas, de modo a permitir o controle completo do fluxo de dados e, conseqüentemente, otimizar a performance da aplicação.

Aplice estilização de forma, Global e Local, de acordo com a necessidade. É terminantemente vedada a utilização de CSS puro, isso quer dizer que somente será considerado para análise o CSS que for aplicado como: `SASS` e/ou `CSS-MODULE`

No rodapé do layout-root deve conter o nome dos integrantes da equipe com os seguintes dados: Nome completo, RM, Turma.

RESPONSIVE WEB DEVELOPMENT

ENTREGAS E PONTUAÇÕES

Atratividade do Design (20 pontos)

Originalidade e criatividade na disposição de elementos visuais. (2,5 ponto)

Coerência e harmonia das cores utilizadas. (2,5 ponto)

Adequação das cores ao tema e propósito do projeto. (2,5 ponto)

Legibilidade e estética das fontes escolhidas. (2,5 ponto)

Consistência no uso das fontes em todo o projeto. (2,5 ponto)

Apelo visual para o público-alvo pretendido. (2,5 ponto)

Impacto emocional e estético do design. (2,5 ponto)

Clareza na comunicação visual e facilidade de navegação. (2,5 ponto)

RESPONSIVE WEB DEVELOPMENT

ENTREGAS E PONTUAÇÕES

Componentização das páginas e itens reaproveitáveis (20 pontos)

- Identificação e separação clara de componentes. (3,5 ponto)
- Facilidade de reutilização de elementos em diferentes partes do projeto. (3,5 ponto)
- Eficiência na manutenção e atualização de componentes. (3,3 ponto)
- Coerência na estruturação e organização dos elementos. (3,3 ponto)
- Consistência no uso de estilos, classes e IDs. (3,3 ponto)
- Documentação adequada dos componentes para facilitar o entendimento e reutilização. (3,3 ponto)

RESPONSIVE WEB DEVELOPMENT

ENTREGAS E PONTUAÇÕES

Consumo das APIs (20 pontos)

- Correto uso das chamadas de API, criada na disciplina de Domain Driven Design. (5 ponto)
- Tratamento de erros e respostas inesperadas. (5 ponto)
- Integração bem-sucedida das funcionalidades que dependem das APIs. (5 ponto)
- Manipulação correta dos dados obtidos das requisições. (5 ponto)

RESPONSIVE WEB DEVELOPMENT

ENTREGAS E PONTUAÇÕES

Criação de Rotas e Navegação com NEXT.JS (20 pontos)

- Definição adequada e funcional de rotas utilizando REACT + TYPESCRIPT + NEXT.JS. (3,5 ponto)
- Navegação fluida entre páginas. (3,3 ponto)
- Utilização eficiente do estado da aplicação para a troca de informações entre as páginas. (3,3 ponto)
- Manuseio correto de parâmetros de rota, quando aplicável. (3,3 ponto)
- Desempenho satisfatório na transição entre rotas. (3,3 ponto)
- Utilização de técnicas para otimização do carregamento das páginas. (3,3 ponto)

RESPONSIVE WEB DEVELOPMENT

ENTREGAS E PONTUAÇÕES

Utilização do Git/Github, Projeto com no Mínimo 10 Commits e Participação de Todos os Integrantes (10 ponto)

- Evidências de colaboração equitativa de todos os membros do grupo. (5 ponto)
- Mensagens descritivas nos commits indicando claramente o que foi realizado. (5 ponto)
- Organização adequada dos arquivos no repositório. (5 ponto)
- Demonstração de uso efetivo de branches, merges e resolução de conflitos, se aplicável. (5 ponto)

RESPONSIVE WEB DEVELOPMENT

ENTREGAS E PONTUAÇÕES

Deploy do Projeto para a Plataforma Vercel (20 ponto)

- Efetivação bem-sucedida do deploy na plataforma Vercel. (5 ponto)
- Disponibilização de uma URL funcional para acesso ao projeto. (5 ponto)
- Configuração adequada das variáveis de ambiente, se necessárias. (5 ponto)
- Otimização para um carregamento rápido e eficiente do projeto. (5 ponto)

RESPONSIVE WEB DEVELOPMENT

ENTREGAS E PONTUAÇÕES

PENALIZAÇÕES

- Grupo deverá gravar um vídeo de no máximo 3 minutos apresentando os recursos do projeto, telas, layout, API funcionando, o mesmo poderá ser disponibilizado via link e hospedado por exemplo Youtube. O não envio do link, acarretará a PERDA de CONCO PONTOS.(-5,0 pontos)
- O grupo deve compactar e entregar o repositório do projeto em formato .ZIP, seguindo as diretrizes do .gitignore para garantir a exclusão das pastas 'node-modules' e '.next'. O não cumprimento desta norma resultará em um arquivo excedendo 50MB, ultrapassando o limite do portal da FIAP, acarretando a PERDA de CINCO PONTOS para todos os integrantes do grupo (-5,0 pontos).
- Caso no rodapé do layout-root NÃO contenha o nome dos integrantes da equipe com os seguintes dados: Nome completo, RM e turma, acarretará a PERDA de CINCO PONTOS (-5,0 pontos).

RESPONSIVE WEB DEVELOPMENT

DISCIPLINAS PARA INTEGRAÇÃO

BUILDING RELATIONAL DATABASE

COMPUTATIONAL THINKING USING PYTHON (OPCIONAL)

DOMAIN DRIVE DESIGN

RESPONSIVE WEB DEVELOPMENT

O QUE DEVE SER ENTREGUE

Colocar um arquivo.txt dentro do zip contendo o link do repositório no GitHub.

Se o repositório for privado, dar acesso ao usuário do professor.

Projeto completo compactado (ZIP) no Plataforma da FIAP.

ONDE DEVE SER ENTREGUE

Portal do Auno: No portal do aluno vá em > Aulas, Clique na opção “Entrega de Trabalhos”; Clique em um trabalho referente a Global Solution; Anexe o arquivo do seu projeto referente a entrega escolhida.

Lembre-se que somente o representante deve enviar o trabalho!!

**NÃO SERÃO ACEITAS ENTREGAS PELO TEAMS OU OUTRO MEIO
DE COMUNICAÇÃO!!**