

Code Challenge: Full Stack Development with Elixir and React

Objective

Welcome! This challenge is designed to see how you integrate backend and frontend components into a cohesive application. You'll be setting up a backend API, building a frontend interface, managing user authentication, and ensuring everything works smoothly within Docker containers.

Requirements

1. Backend

- **Technology:** Develop the backend using Elixir. Using the Phoenix framework is recommended but not mandatory.
- **Functionality:** Your API should handle CRUD operations. This can be something simple like todos or blog posts.
- **Authentication:** Implement basic user authentication. Assume that the users are already present in the database; no sign-up functionality is needed.
- **Validation:** Include basic validation checks for CRUD operations.

2. Frontend

- **Technology:** Use React for the frontend.
- **Functionality:** Users should be able to log in with pre-existing credentials and perform CRUD operations.
- **Design:** Focus purely on functionality—no need to apply any styling or CSS. The simpler, the better.

3. Integration and Deployment

- **Monorepo:** Both the frontend and backend should be in the same repository.
- **Containerization:** Ensure that all components are dockerized and can be run within Docker containers.
- **Startup Script:** Provide a script to launch the entire application stack with one command.

Evaluation Criteria

- **Learning and Adaptability:** If you're new to Elixir or Phoenix, that's fine. We're interested in your ability to learn new technologies.
- **Code Organization:** We value clean, organized code that is easy to read and maintain.
- **Functionality:** The application should work as expected. We're looking for basic, solid functionality rather than a polished final product.
- **Simplicity:** Avoid over-engineering the solution. Aim to spend no more than 4 hours on this challenge and keep it straightforward.

Submission Guidelines

- **Repository:** Please push your code to a Git repository (e.g., GitHub, GitLab) and ensure it is accessible to our team.

- **Documentation:** Include a README.md in your repository detailing how to set up and run the application.

Note

This challenge is about how you approach integrating different technologies into a functional application. We're not looking for expertise in Elixir, Phoenix, or React, nor are we concerned with the aesthetics of the frontend at all. It should be a raw/unstyled frontend with minimal or no CSS. Please don't waste your time making it look good.