Full Stack Engineer Assignment

Welcome to the Full Stack Engineer assignment! This exercise is designed to evaluate your skills in Java, React, Docker, and your approach to building and structuring an application.

Project Overview

The goal of this assignment is to create a simple web application that allows users to **list and view details of TV Shows**. You are required to build both the backend (Java 8+) and frontend (React) parts of the application and dockerize the entire setup.

Requirements

1. **Backend (Java 8+)**
   - Develop a REST API in Java (Java 8 or above) with endpoints for:
     - Consume provided tvtitles.txt file
       - process titles and get tv show details including summary from tvmaze api (http://api.tvmaze.com/singlesearch/shows?q={TITLENAMEHERE})
       - Persist **required** details in memory or in db
     - Fetching a list of TV Shows.
       - Fetch tv show list
     - Display details of TV Show.
   - Use any framework you're comfortable with.
   - Data persistence is not necessary, but feel free to use an in-memory database (e.g., H2) if you'd like.
   - The API should be well-documented.
2. **Frontend (React)**
   - Build a frontend using React for interacting with the backend API.
   - The UI should allow users to:
     - List all TV shows (similar to what you see https://www.tvmaze.com/shows, please ignore filters)
     - Display TV show details (similar to what you see https://www.tvmaze.com/shows/60/ncis, please only show Main tab details/summary of TV Show with show info which has network,schedule,status etc)
   - Use any CSS library or framework to style the application (e.g., Bootstrap, Material-UI).
3. **Docker**
   - Dockerize both the backend and frontend applications.
   - Provide a `docker-compose.yml` file to easily set up and run the application.
4. **TV Maze API details**

- API Endpoint: http://www.tvmaze.com/api
- Get TV-Maze ID for the Show
  - http://api.tvmaze.com/singlesearch/shows?q=Game%20Of%20Thrones
- Get Show Details
  -

http://api.tvmaze.com/shows/183/akas

Deliverables

1. **Codebase**: Submit your codebase via a GitHub repository link or a zip file.
2. **README.md**: This file should contain:
   - Instructions on setting up and running the application.
   - Brief documentation of the endpoints.
   - Any assumptions or decisions made in the project.
3. **Docker**: Ensure both backend and frontend are dockerized and can be started with `docker-compose up`.

Evaluation Criteria

1. **Code Quality**: Clean, readable, and modular code.
2. **Functionality**: Completeness of the required features.
3. **Documentation**: Clarity of instructions and API documentation.
4. **Best Practices**: Adherence to best practices for Java, React, and Docker.
5. **Creativity**: Any extra features or improvements are welcome!

Notes

- Ensure your code is production-ready.
- Feel free to reach out if you have any questions or require clarification on any of the requirements.

Good luck, and happy coding!