take-home-assignment-B

Getting Started

- copy the .env.example file into a .env file
- `docker compose build`
- `docker compose up`
- `npm run migrate`
- `npm run seed`
- if you view your database, you should be able to see a populated form data table
- running the following in your terminal will perform a GET request to fetch the form data

```
curl --location '<http://127.0.0.1:8080/form/>{insert your generated form id here}' --header 'Content-Type: application/json'
```

Introduction

The purpose of this project is to evaluate your full stack web development skills with an example project that closely resembles your day to day tasks at Vial.

You will build a simple **Form App** where a user can build their own customized form and save data from that generated form.

NOTE: Any images provided in the assignment are just examples, and the frontend does not need to look the same as the provided mock. Be creative with your design!

Preferred workflow

- Fork the repository

- Create as many commits as needed, with the corresponding descriptive message/description

Tech stack

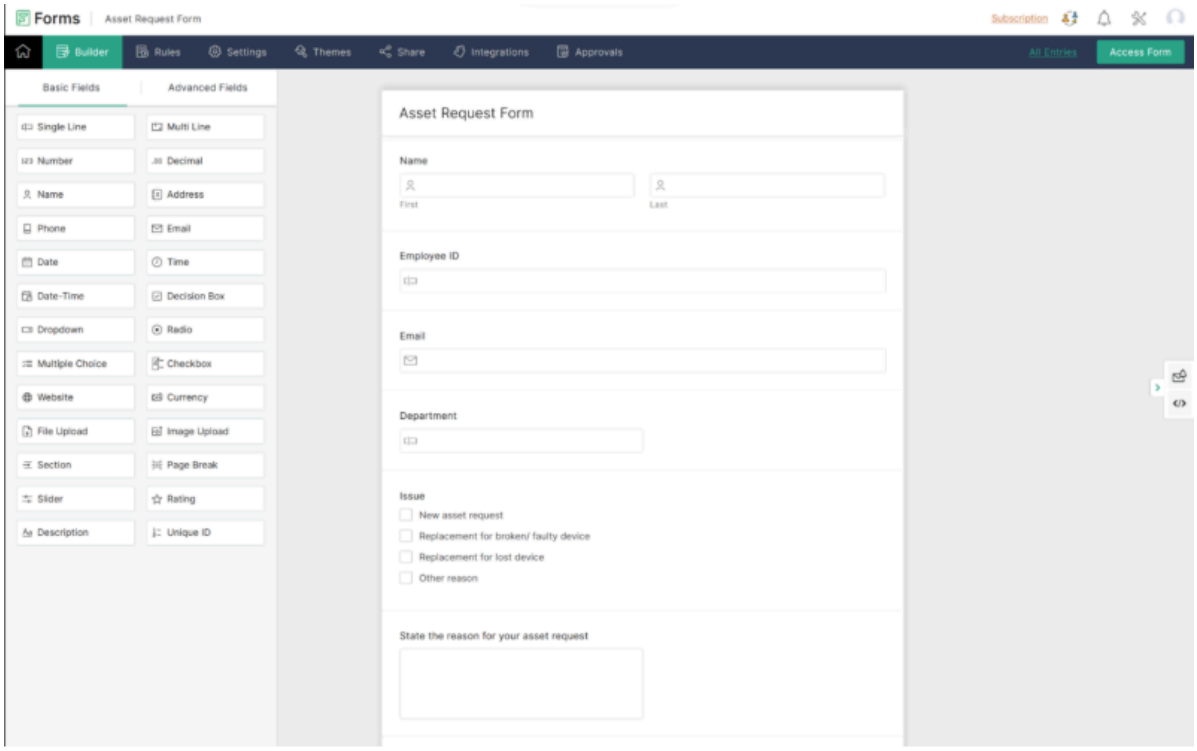
- Node
- Typescript
- Fastify
- Prisma ORM
- PostgreSQL
- Docker and Compose

Requirements

1. **Frontend (UI)**:
    - Use TypeScript / React or other framework to build a single-page web application.
    - Build a frontend that can save a user's desired form
        - name of the form
        - fields (saved as JSON)
            - field type
            - question
            - required
            - any other variables as you see fit
        - the user should be able to customize different field types (text, textfield, datetime, boolean, etc), at minimum 2-3 different types be creative on this one!
    - Build a frontend that can display a user's form
        - should clearly use the saved form's information (field type, question, whether the question is required)

    - displays the field as an input for the user to type into
- User should be able to save this form data and submit the data as a source record with source data entries (the seed data shows some examples)

(OPTIONAL) Display saved source data from database - not required



2. **Backend (API)**:
    - Build a RESTful API for the form, source record and source data, a simple application skeleton is provided for the BE with example seed data for all three tables
    - The API should have the following endpoints:
        - FORM
            - ENDPOINT 1: Create a form
            - ENDPOINT 2: Fetch a form by ID
        - SOURCE RECORD
            - ENDPOINT 3: Create a source record (related to a form id) with source data

3. **Database**:
    - (Given) The **form model** should have the following fields:
        - `id`: Unique identifier (auto-increment or UUID).
        - `name`: String, required
        - `fields`: JSON, required
    - The **source record model** should have the following fields:
        - `id`: Unique identifier (auto-increment or UUID).
        - `formId`: Foreign Key Unique identifier
    - The **source data model** should have the following fields:
        - `id`: Unique identifier (auto-increment or UUID).
        - `sourceRecordId`: Foreign Key Unique identifier
        - `question`: String, required
        - `answer`: String, required

Guidelines

- **Tech Stack**: You are free to choose any technologies or libraries you prefer, but please stick to modern web development practices.
- **Code Quality**: Please ensure your code is clean, well-organized, and well-documented. Add comments where necessary to explain key decisions.
- **Time Management**: This is intended to be a 4+ hour assignment. Focus on getting the basic functionality working first, and add optional features if you have time.
- **(OPTIONAL) API Documentation**: Provide basic API documentation (e.g., using Swagger or in README.md).
- **(OPTIONAL) Deployment**: If possible, deploy your application to a service like Heroku, Vercel, or Netlify, and share the live URL with us.

Submission Instructions

- Share a GitHub repository with your code and provide instructions for how to run the project locally.
- (OPTIONAL) If you deploy the application, include the live link in the repository's README.
- Ensure that your submission includes clear documentation on how to set up and run the backend and frontend.

We hope you have fun with the assignment and we look forward to hearing from you!

Take Home Zip File:

https://storage.googleapis.com/take-home-assigment/take-home-assignment-B-main.zip