Division: Moments of Life (MOL)
Team: Virtual Intelligent Chat Assistant (VICA)

Team Overview

At VICA, we are developing a new platform for agencies to onboard a wave of new virtual assistants for government services. This platform will eventually replace all "Ask Jamie" chatbots and enable the agencies to onboard and collaborate effectively on it, so that they can keep up with the questions by the citizens and businesses in the long term.

Assessment Overview

Depending on the role you are signing up for, you will be assessed on part or whole of the tech stack that we are using at VICA. Please select the assessment that is relevant to the role you are applying for.

1. React frontend development
2. Node.js/Python backend development
3. Full stack development (React + Node.js/Python)
4. Data science challenge

Submission Guidelines

Start a new repository in GitHub and send the link in an email to the assessors. In your repository, make sure that you make small and frequent commits. Each commit should have a descriptive message and clean code. If you have assumptions made about the requirements of the assessment, please write that in a section in a README file.

To ensure the integrity of this assessment, do not discuss and share the work with current candidates or anyone else. Please do not host this application on AWS, GCP, Azure, DigitalOcean, Heroku or other cloud infrastructure.

Software Engineering Assessment

Background

A book fan club, ABC Book, allows its members to borrow books from the club. Due to increasing membership, the club management needs a web console for easier management. It should be used by 3 types of users: **Admin**, **Editor** and **Member**.

This console should support the following functions:

1. **User Management**
    - Only accessible to Admin and Editor
    - Each user should have **Name**, **Role**, and **Date Joined** attributes
    - Only an Admin can add, remove and update users
2. **Book Management**
    - Accessible to all types of users
    - Each book should have **Title**, **Description**, **Genre**, **Author**, **Year Published**, **Borrowing Availability Status**, **Last Borrower**
    - Only Admin and Editor can add, remove and update books
    - All users can borrow and return books
    - Each book may have multiple copies

Assessment Requirements for Backend Development

Build a web service that fulfils the following requirements (50%):

1. For Node.js, use Express.js.
   For Python, use Flask or FastAPI.
2. Use MongoDB. Any database driver is accepted.
3. Routes
    - CRUD of user objects
    - CRUD of book objects
    - Borrowing and returning of books
4. Code formatting and linting
5. README file for setup instructions

Bonus points (10%) will be awarded for implementing a maker checker rule for adding, removing and updating users (i.e. an Admin changes to add/remove/update users will only be finalised after approved by another Admin).

Finally, also in your README file, explain and justify (50%):

1. How you structure your project
2. How you designed the APIs
3. The code implementation
4. Scalability of your project

Assessment Requirements for Frontend Development

Build a web console that fulfils the following requirements (50%):

1. Start an app using Create React App (CRA)
2. Write the app in TypeScript
3. Use React and Redux
4. Initialise your app with hardcoded data
5. Pages
    - A User Management page that:
        - List all the users
        - Allow Admin to add/remove/update users
        - Support sorting, pagination and filtering
        - You can use any React UI frameworks for the table
    - A Book Management page that:
        - List all the books
        - Allow Admin and Editor to add/remove/update books
        - Allow all users to borrow/return books
        - Support sorting, pagination and filtering
        - You can use any React UI frameworks for the table
    - An Analytics page that:
        - Use a suitable chart to demonstrate the breakdown of books by genre
        - Use a suitable chart to demonstrate the breakdown of books by year published
        - You can use any React chart libraries
6. Code formatting and linting
7. README file for setup instructions

Bonus points (20%) will be awarded for using React routing, custom hooks from React v16+ and mobile responsiveness.

Finally, also in your README file, explain and justify (50%):

1. How you set up and design the Redux store
2. The project file structure
3. The code implementation
4. Scalability of your project

Assessment Requirements for Full Stack Development

Implement both the backend and frontend requirements. Ensure that the web application is using data from the web service instead of the hardcoded data.