

Assignment Submission Guidelines

- There are 2 assignments, 1st assignment needs to be submitted by 30th April 2 pm, and 2nd assignment needs to be submitted by 1st May 2 pm.
- After completing it, please push it to GitHub public repo and share the GitHub link with you name at tech@attainu.com
- For any queries, please contact at 9474669944 (Ruhan)

Assignment 1 - Front End Assignment

Submit Deadline - Submit by 30th April 2 pm

Task Details:

- Please create a **User Directory Application** React App, where the followings have to be implemented
 1. Run the **JSON server** and fetch the data using pagination.
 2. **Name, Country, Date of birth**, and **Email** should be shown in the UI with a Minimalist design.
 3. **Date of birth** format should be - 1st Jan 1990
 4. Users should be able to search users by entering text in the search text box.
 5. The filter should be implemented on country and year of birth.
 6. Users should be able to **create, update** and **delete** a user.
 7. State Management should be designed well.
 8. UI Should be responsive
 9. Given that the data set is large, the fetch, filter, and search logic must be done efficiently.
- MOST IMPORTANTLY - code quality matters.

Other Details :

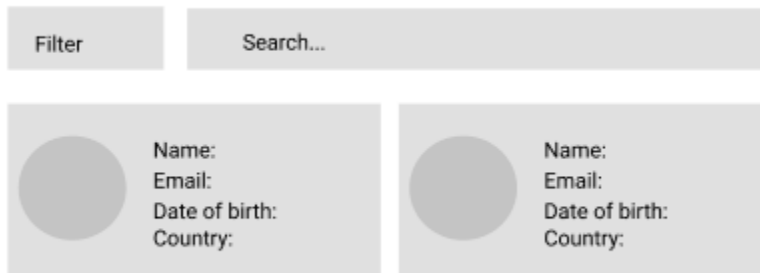
- You can use any UI Library of your choice.
- The app should run smoothly.

Bonus :

- Autocomplete search.
- Filter - select multiple values.

You can find the JSON server [here](#)

For UI, you can refer to the below image.



Assignment 2 - Back End Assignment

Submit Deadline - Submit by 1st May 2 pm

Your task is to build a simple stateless microservice in Node Js, with four major functionalities -

- Authentication
- JSON patching
- Image Thumbnail Generation
- Adding User address in database

We have no requirements for which frameworks/libraries/database to use, choose whichever seems best suited for the task!

Required Endpoints

The API should feature the following endpoint functionality -

Public Endpoints

- Login Request body should contain an arbitrary username/password pair. Treat it as a mock authentication service and accept any username/password. Return a signed Json Web Token(JWT, <https://jwt.io/>) which can be used to validate future requests.

Protected Endpoints

The following two endpoints should be protected. The JWT obtained in the “Login” endpoint must be attached to each request. If the JWT is missing or invalid, these endpoints should reject the request.

- Apply Json Patch Request body should contain a JSON object and a JSON patch object (<http://jsonpatch.com/>). Apply the json patch to the json object, and return the resulting json object.
- Create Thumbnail Request should contain a public image URL. Download the image, resize to 50x50 pixels, and return the resulting thumbnail.
- Create a Request to add a user address and save into the database.

General Requirements

Code Requirements

- API should reject invalid request inputs. Test the edge cases!
- Use modern javascript ES6 syntax.

Other Requirements

- Use Git for version control, and host the project in a Github repository.
- Project should contain documentation with setup and usage instructions.
- Project should install all dependencies with “npm install”, should start the server with “npm start”.
- MOST IMPORTANTLY - code quality matters.

