AdmitKard FullStack Intern Exercise - User Profile

Problem Statement

Create 2 web-pages:
Web-Page 1: to take input from users. Data should be saved in backend DB.
Web-Page 2: to get the details of an existing user based on email id.

Web-Page 1: Page Structure:
Form Fields:

1. **Name** (Text Field) - Mandatory Field
2. **Email** (Email Field) - Mandatory Field
3. **Contact Number** (Number Field) - Mandatory Field
4. **Course Level** (Single Select Drop Down with options - [UG, PG]) - Mandatory Field
5. **Country Preferences** (Multi Select with dropdown options - [USA, Australia, New-Zealand, Canada, UK, Ireland, Germany]) - Mandatory Field
6. **Date of Birth** (Date Field) - Optional Field

Example:

Name of the field	DataType	Mandatory or Optional Field	Example
Name	Text Field	Mandatory	Sashank Kumar
Email	Text Field	Mandatory	shashank@abcd.com
Contact Number	Number Field	Mandatory	9876543210
Course Level	Single Select DropDown	Mandatory	ug
Country Preferences	Multi Select DropDown	Mandatory	Australia, UK
Date of Birth	Date Field	Optional	23/09/1994

Web-Page 2: Page Structure:
Form Fields:

1. **Email Id** (Text Field) - Mandatory Field

Expected output:

Details of users should be fetched from Database and should be displayed on this page.

Minimum Requirements

- We should be able to create user details from this page with all the fields.
- If we are submitting the details of already existing user. Means if we again submit the form with the same email id then new fields should be updated in db for that existing email id.
- Proper handling of multi select in web-page and backend db.
- Proper handling of Date of Birth Field.

Good to have:

- All kinds of validations to stop user from filling garbage values in the fields.
- Validations both on fornt-end and Backend
- Web-Page should be responsive (Both for desktop and mobile browser)

Few Pointers:

- You can use your own custom approach.
- Try to keep things as simple as possible. Do not complicate it for the system engineers for operations.
- You can use your own example for services: 1) API service 2) Database

Pointers for extra edge:

- You can use NodeJS / ExpressJS for backend APIs
- You can use MongoDB for Database
- You can use ReactJS for building web-page

How to submit the solution

- You have to send us a zipped version of your project (it will include your services for API and Throttling Service).
- Include a Readme on how to see the project in action.
- Include a video recording of the app in action.
- You can also host your project somewhere like heroku, github pages etc.