

## Capabilities Test

This test is designed to understand your coding style, your approach to solving problems and your knowledge of ASP.NET Core with React and Redux. You are supposed to write a self-containing application that solves the problem below. APIs to fetch data are provided in the sample below along with the instructions. You need to develop frontend as well as backend of the application. You will only be focusing on ASP.NET Core, React and Redux.

## Deliverables

You are required to deliver a .zip folder with pre-compiled DLL/Scripts. Also, in order to view your solution provide the source code for frontend and backend with compiled frontend code.

Follow below to provide source code of working application

- ASP.NET Core
  - Stable runnable application written in ASP.NET Core (preferable  $\geq 3.1$ )
- React
  - Pre-compiled react scripts combined with ASP.NET views

**NOTE:** Preference will be given if you write SQL Queries/Code First to prepare your structure and data

## Scenario

The hypothetical application that you are going to develop is meant for Admin Staff to enter new Students along with their basic information (Name, Nationality, and Date of Birth) as well family information (like parents and siblings). Once an Admin Staff submits the application, he/she cannot edit the information and all fields should appear as disabled for him/her.

This information then needs to be approved by the Registrar. The Registrar can edit any information of a student along with adding/editing or even deleting family members.

Both Admins and Registrars see a table where they can see registered students and can click to view their details in a Modal.

There is no login into the system, the role of the user can be changed via a drop down from the top menu on the landing page.

## Screens

### Landing Page

The landing page contains a table that lists all students. The rows can be clicked and opens the Modal with Student Details (see below). This screen also contains an Add New Student Button that will be used to create a New Student. It also contains a dropdown on the top of the page that allows the user to switch roles

### Student Modal

This is a Modal that will contain 2 Sections.

**Section 1** is Basic Information that contains the First Name, Last Name, Date of Birth (Date Picker) and Nationality (drop down).

**Section 2** is Family Information that will contain a section for each family member. Each Family Member Section contains the following fields: Name, Relationship (Dropdown with 3 choices Parent, Sibling, Spouse), Nationality (Dropdown). This section also contains the Add Family Member button which is used to add new Family Members (the user can add as many family members he wishes), as well as a Delete Family Member used to delete each Family Member. Also, all fields of the Family Member can be edited by the Registrar.

All Fields in the Modal should be disabled, when the Admin opens the students that are already submitted.

All Changes should be submitted only when the user taps submit and should be discarded when the user taps cancel.

## Endpoints

The following endpoints need to be created in order to fetch and store data.

### Students

#### Get all Students

**[GET]** /api/Students

#### Response Sample:

```
[
  {
    "ID": 1,
    "firstName": "John",
    "lastName": "Doe",
    "dateOfBirth": "2023-07-31T12:44:55.403Z"
  }
]
```

#### Add a new Student with Basic Details Only

**[POST]** /api/Students

#### Request Sample:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
}
```

#### Response Sample:

```
{
  "ID": 1,
  "firstName": "John",
  "lastName": "Doe",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
}
```

Updates a Student's Basic Details only

**[PUT] /api/Students/{id}**

**Request Sample:**

```
{
  "ID": 1,
  "firstName": "John Update",
  "lastName": "Doe",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
}
```

**Response Sample:**

```
{
  "ID": 1,
  "firstName": "John Update",
  "lastName": "Doe",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
}
```

Gets the Nationality of a particular student

**[GET] /api/Students/{id}/Nationality**

**Response Sample:**

```
{
  "ID": 1,
  "firstName": "John",
  "lastName": "Doe",
  "nationalityId": 1,
}
```

Updates a Student's Nationality

**[PUT] /api/Students/{id}/Nationality/{id}**

**Response Sample:**

```
{
  "ID": 1,
  "firstName": "John",
  "lastName": "Doe",
  "nationalityId": 2
}
```

Gets Family Members for a particular Student

**[GET] /api/Students/{id}/FamilyMembers/**

**Response Sample:**

```
[
  {
    "ID": 2,
    "firstName": "John Son",
    "lastName": "John Family",
    "dateOfBirth": "2023-07-31T12:44:55.403Z",
    "relationshipId": 3
  }
]
```

Creates a new Family Member for a particular Student (without the nationality)

**[POST]** /api/Students/{id}/FamilyMembers/

**Request Sample:**

```
{
  "firstName": "John Son",
  "lastName": "John Family",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
  "relationshipId": 3
}
```

**Response Sample:**

```
{
  "ID": 2,
  "firstName": "John Son",
  "lastName": "John Family",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
  "relationshipId": 3
}
```

## Family Members

Updates a particular Family Member

**[PUT]** /api/FamilyMembers/{id}

**Request Sample:**

```
{
  "firstName": "John Son",
  "lastName": "John Family",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
  "relationshipId": 3
}
```

**Response Sample:**

```
{
  "ID": 2,
  "firstName": "John Son",
  "lastName": "John Family",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
  "relationshipId": 3
}
```

Deletes a family member for a particular Student

**[DELETE]** /api/FamilyMembers/{id}

Gets a nationality associated with a family member

**[GET]** /api/FamilyMembers/{id}/Nationality/{id}

**Response Sample:**

```
{
  "ID": 2,
  "firstName": "John Son",
  "lastName": "John Family",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
  "relationshipId": 3,
  "nationalityId": 2
}
```

Updates a particular Family Member's Nationality

**[PUT]** /api/FamilyMembers/{id}/Nationality/{id}

**Response Sample:**

```
{
  "ID": 2,
  "firstName": "John Son",
  "lastName": "John Family",
  "dateOfBirth": "2023-07-31T12:44:55.403Z"
  "relationshipId": 3,
  "nationalityId": 1
}
```

Nationality

Gets all nationalities in the system

**[GET]** /api/Nationalities