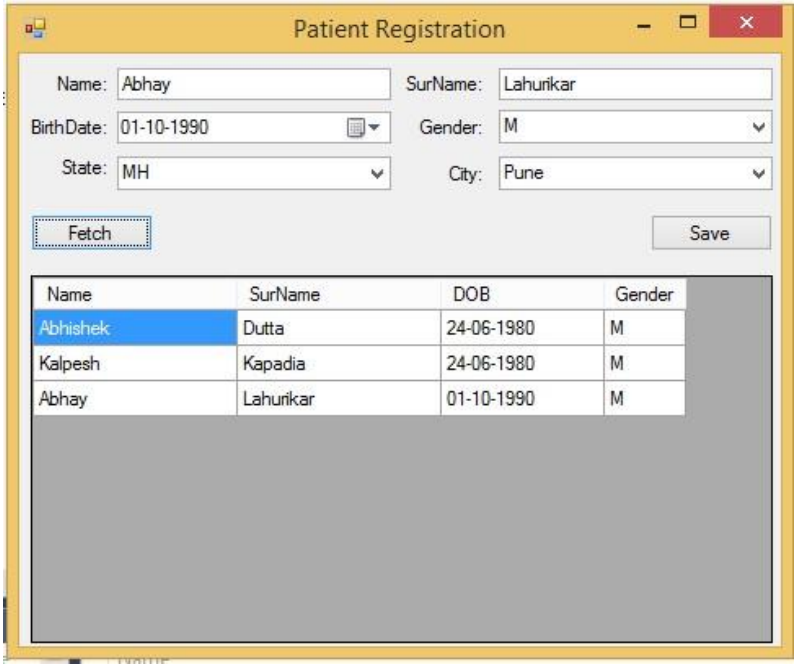
Assignment is to be done using:

- Angular 2 /4 /5 ASP.Net or Asp.net core, webAPI.
- Code should be unit testable.
- Writing some unit test cases is mandatory.
- Duration – 2 hours

Requirement:

- Register a new Patient- Create a page with 6 fields
  - Name, Surname, Gender, Date of Birth, State, City
- Allow Filtration of 2 fields (e.g. If user selects a state Maharashtra,
- all the cities in Maharashtra should be populated in <City> combo)
  - State, City
    - Validation:
    - Birthdate
      - Should not be greater than today's date
      - Should not be less than 100 years
    - Name and Surname
      - Only alphabets are allowed
      - No special characters are allowed.
    - Patient must be unique base on all four fields

UI Design:



To Save and fetch the data, please refer iMedOneDB.DLL (present on desktop) in your application/project-

This <dll> contains, following models/classes:

- iMedOneDB.Models.TBLPATIENT
- iMedOneDB.Models.Tblstate
- iMedOneDB.Models.Tblcity

Saving and fetching the data:

In this <dll>, there is a class “DBContext” which is responsible for saving and fetching the data.

```
bool DBContext.SaveAll<iMedOneDB.Models.TBLPATIENT>(IEnumerable<iMedOneDB.Models.TBLPATIENT> objectsToSave)
```

e.g.

```
DBContext.SaveAll<iMedOneDB.Models.TBLPATIENT> (patientList);
```

```
IEnumerable<iMedOneDB.Models.Tblcity> DBContext.GetData<iMedOneDB.Models.Tblcity>()
```

e.g. `var cities = DBContext.GetData<iMedOneDB.Models.Tblcity> ();`

```
IEnumerable<iMedOneDB.Models.Tblcity> DBContext.GetData<iMedOneDB.Models.Tblcity>(int id)
```

e.g.

```
var city = DBContext.GetData<iMedOneDB.Models.Tblcity> (cityId);
```