# Directions & Deliverable

In this exercise, you will build a Razor Pages .NET Core 2.1 DMV Change of Address Application.

Submit your zipped up Visual Studio project electronically to Blackboard by the due date specified.

#### Description

In the State of Oregon, you are required by law to submit a change of address notice for DMV records. The goal of this assignment is to develop an online version of the following form.

A screenshot of a cell phone

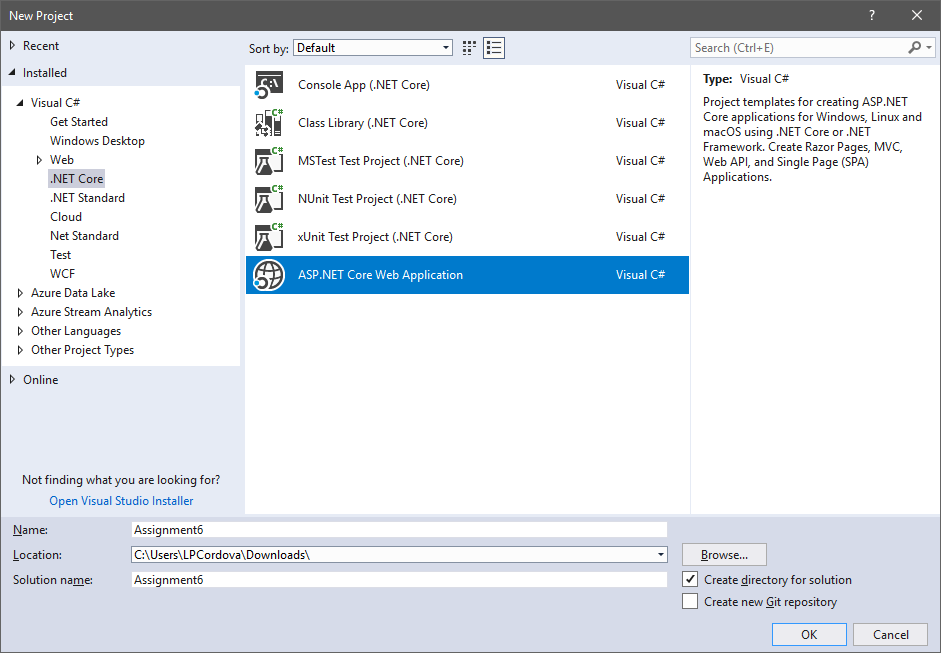
Description automatically generated

**Requirements Specification**

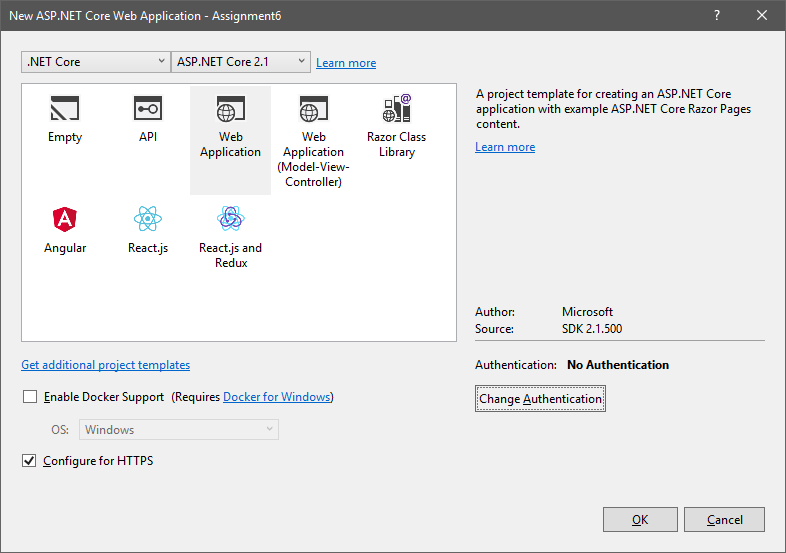
1. The landing page (Pages/Index.cshtml) can be simple (e.g. just some text that says “Welcome to the Oregon DMV Change of Address Application” along with instructions for the user to register for an account.
2. Create two additional pages:
   1. One page to view all current requests for change of address. The list should be represented by a table containing at a minimum, the name of the person requesting the change of address and the date submitted. The user should only see a list of their requests. If the logged-in user is an administrator, they should see all requests.
   2. Another to submit a new request for change of address.
3. Update the site navigation to reflect the two pages which should be visible for logged-in users only.
4. Include everything you see on the form except the signature and voter information checkbox. You should auto-generate and record the date when the form is submitted. Include a Boolean that will indicate whether the request has been processed by DMV staff.

**Project Setup Directions**

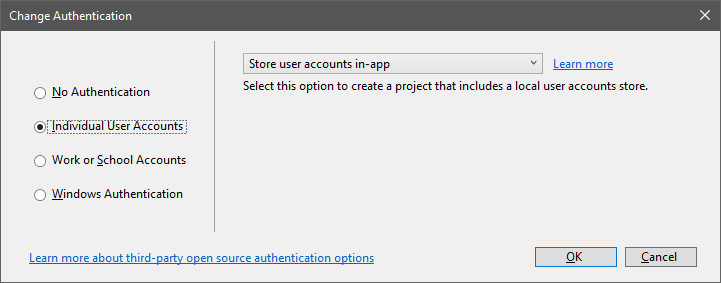
1. To start, create an ASP.NET Core Web Application in Visual Studio.



1. In the subsequent dialog, choose **Web Application** as the type of application, **ASP.NET Core 2.1** as the version, and click the **[Change Authentication]** button.



1. On the Change Authentication dialog, choose Individual User Accounts and click OK.



**Tips**

1. Start with the data model. The input form and data can be one table/class for simplicity. Extra credit (5 points) will be given for properly normalized tables. After every change to the model, remember to run a database migration.
2. Be sure to use annotations on the attributes/properties to distinguish the fields that are required as well as to provide more human readable field names.