**Student Information System (SIS):**

**Approach 1:**

*Technologies:* HTML, JQuery, CSS

**HTML:** for display

**JQuery:** StudentApi AJAX Calls

**CSS:** for styling

**Pros:**

1. Lightweight
2. No hosting required

**Cons:**

1. Underlying API credentials exposed in the page source
2. Custom changes becomes additional overhead

**Approach 2:**

*Technologies:* HTML, JQuery, CSS, ASP.NET Web API

**ASP.NET Web API:** Wrapper around student API

**HTML:** for display (We can use aspx also. I just went with HTML again)

**JQuery:** StudentApi AJAX Calls (Web Api’s)

**CSS:** for styling

**Pros:**

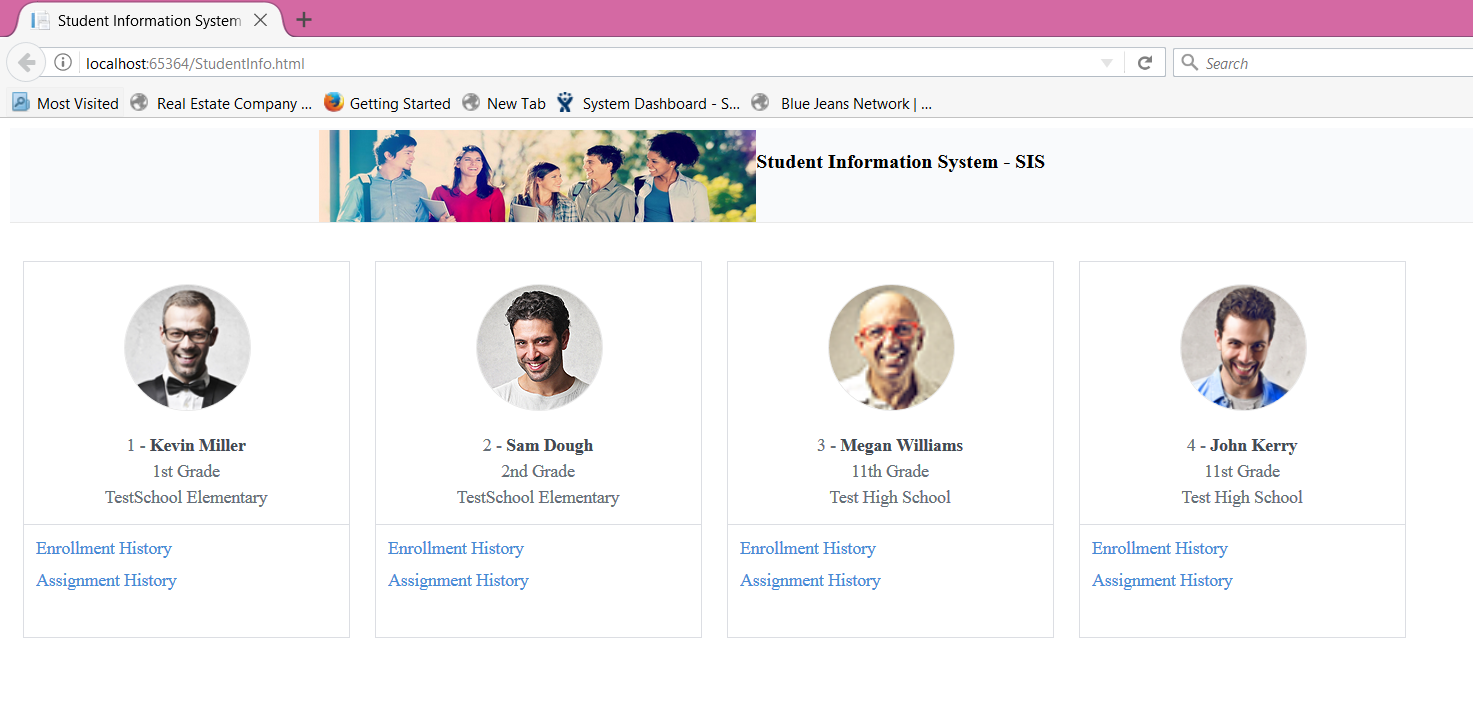
1. Underlying API details not exposed to client
2. Customization can be done easily. (ex: additional properties can be added on top of student api result set)

**Cons:**

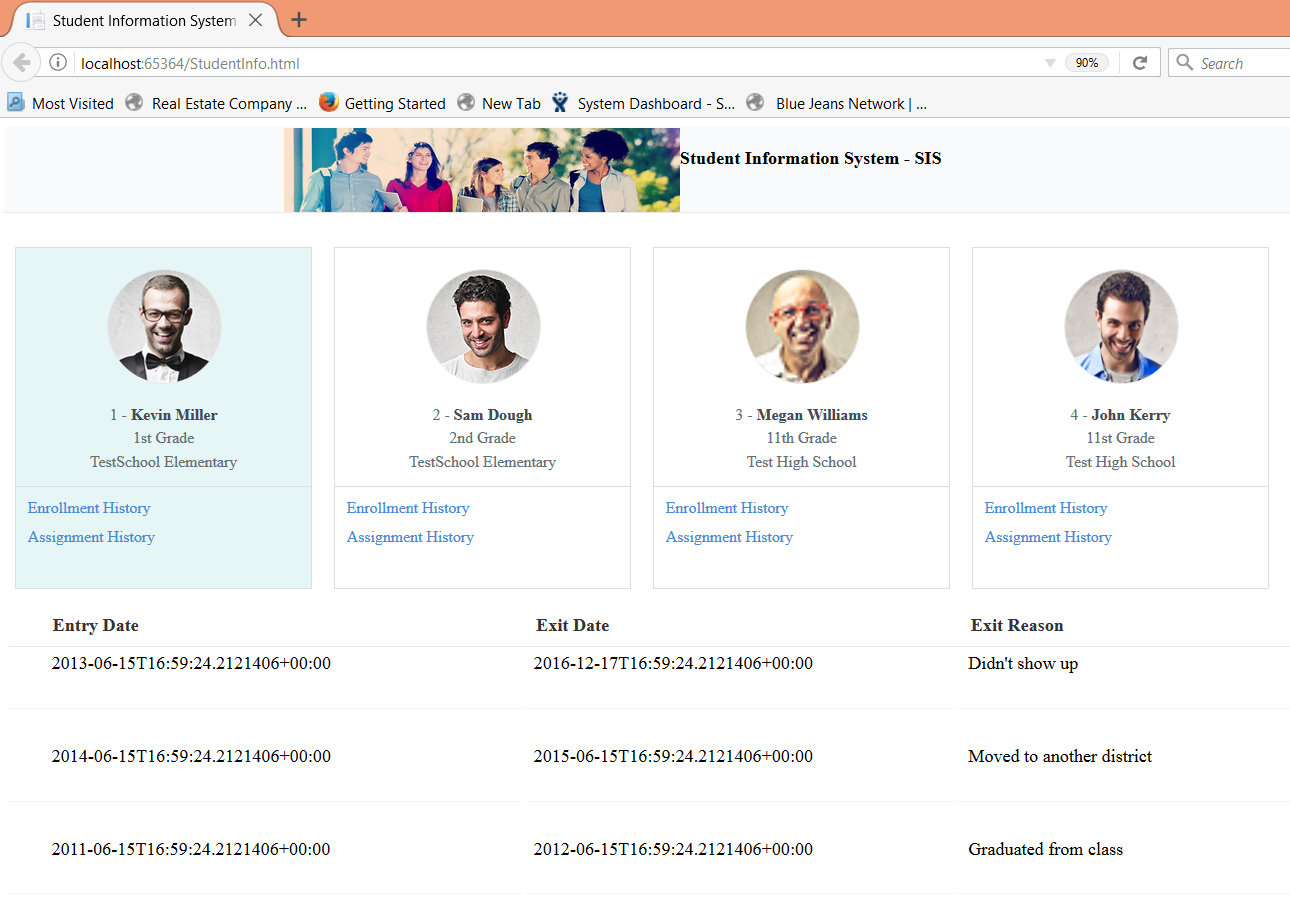
1. Still client can hit the API and get whatever details (ex: <http://www.sis.com/api/student> provides all the student information). Probably to overcome this we can authenticate the users during API access.

**Snapshots:**

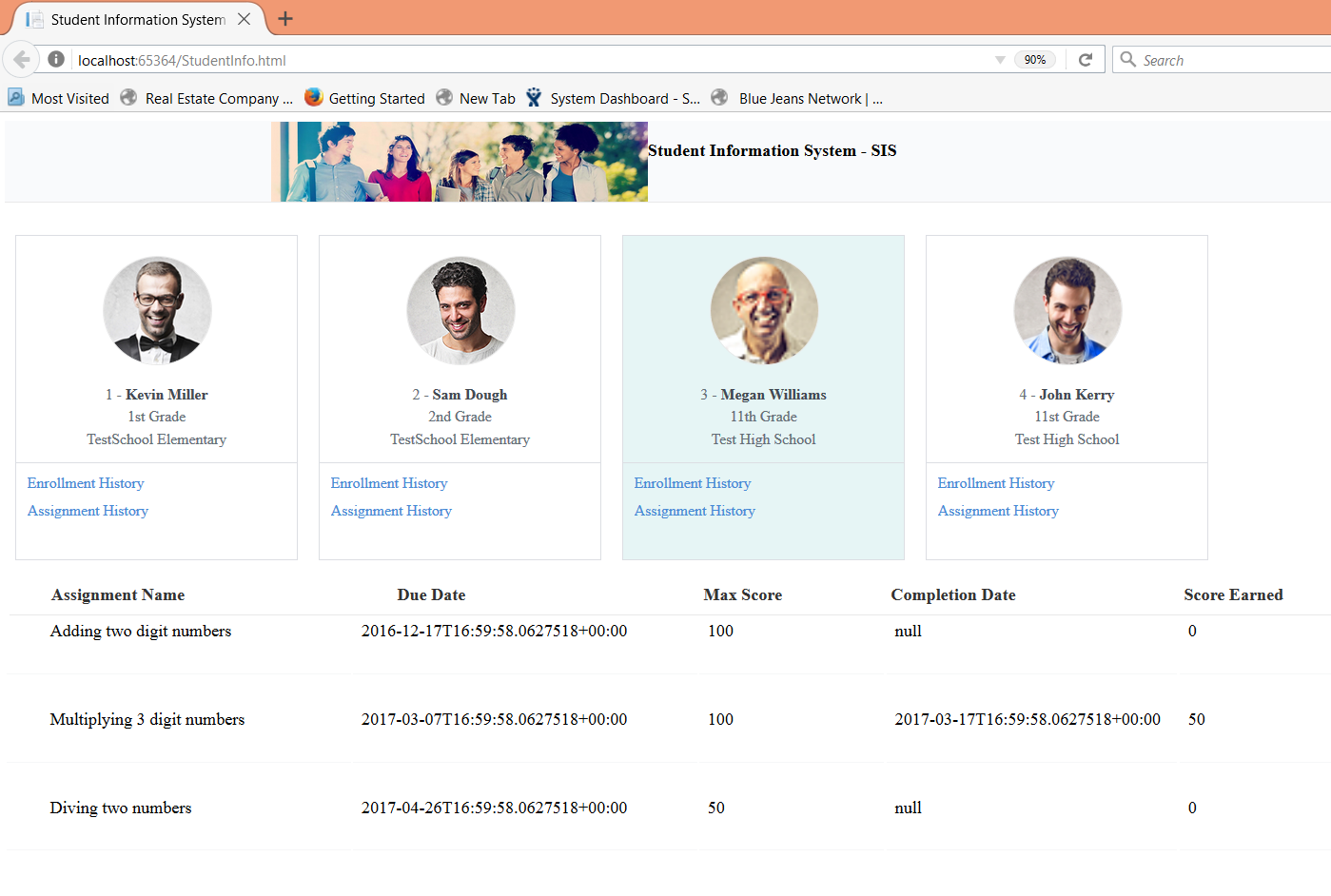
**Home Page :**



**Enrollment History:**

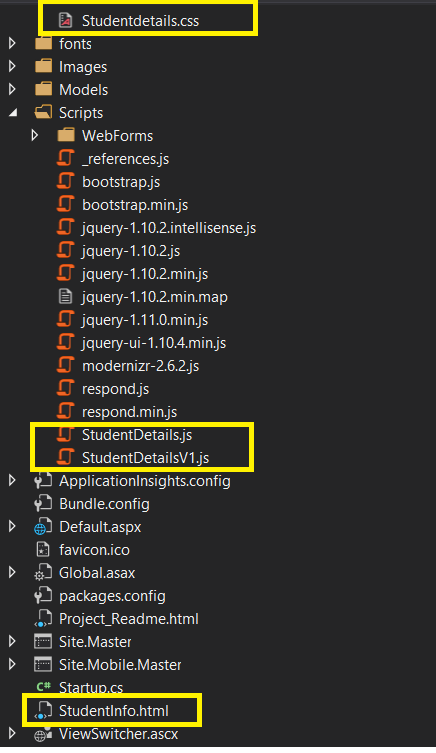


**Assignment History:**



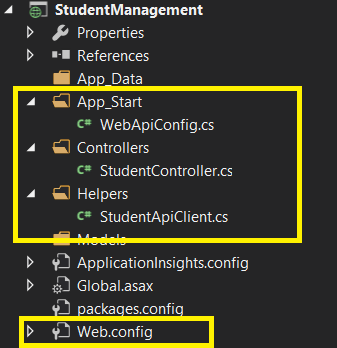
**Main files:**

**SIS Project (Web)**



1. StudentDetails.js (Approach 1)
2. StudentDetailsV1.js (Approach2)

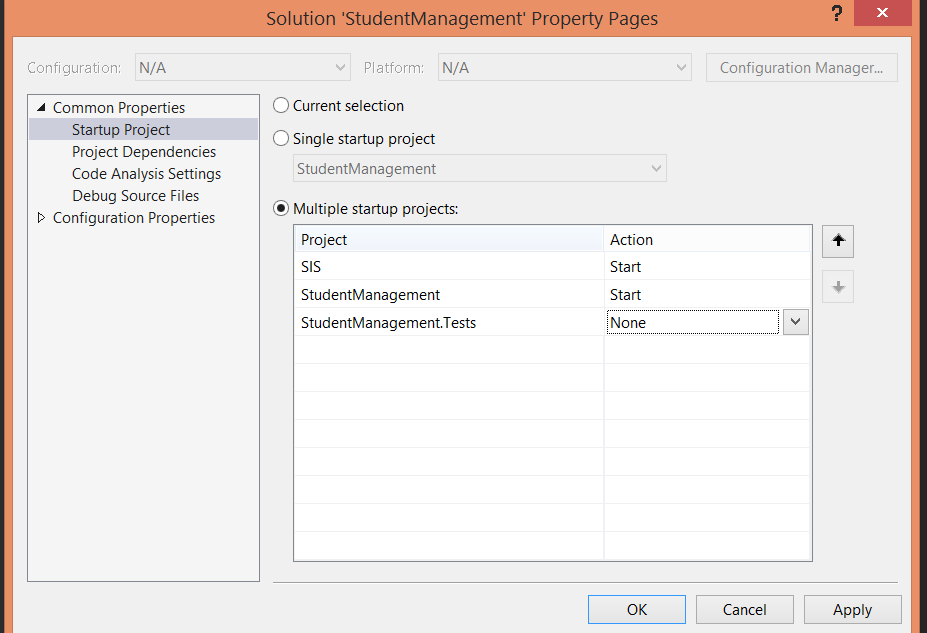
**Student Management (API)**

****

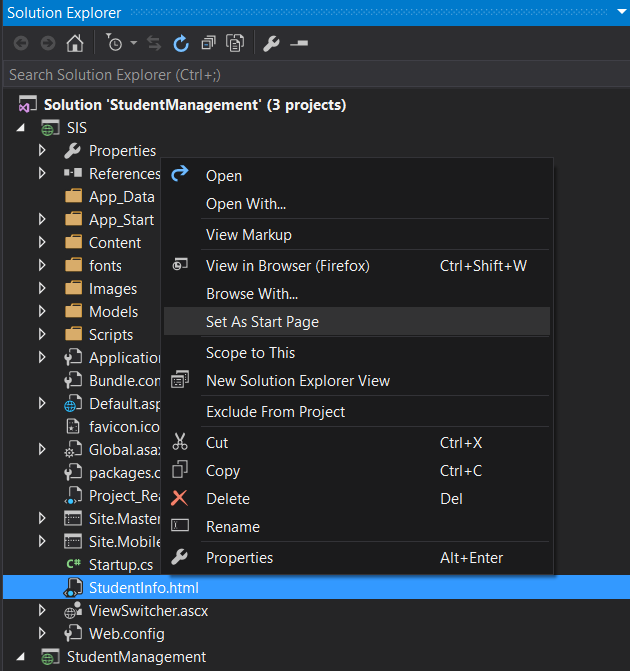
**How to run:**

Steps to be followed after download the project from git.

1. Right click Student Management Solution go to properties and select multiple projects (Set Action to Start for SIS and StudentManagement) and click apply.



1. SIS project => set Studentdetails.html page as start page.



1. Studentmanagement project=>Right click properties and under web set specific page as” api/student”

