



­

Assignment Document:

ASP.NET MVC Practice Case Study

Version: 1.0

Cognizant

500 Glen Pointe Center West

Teaneck, NJ 07666

Ph: 201-801-0233  
[www.cognizant.com](http://www.cognizant.com)

­

Contents

[Business Requirement 3](#_Toc448494252)

[Scope of the Case study 3](#_Toc448494253)

[Requirement 1: Creating a Controller with default view as Index 4](#_Toc448494254)

[Requirement 2: Data Validations 4](#_Toc448494255)

# Business Requirement

Student Registration portal enables the students to register into the system to browse through the portal for various activities.

Student Registration System has the following features:

For Admin:

* + Approve/Reject Student Registration’s.
  + Add Approved Students to the Department table.

For Normal User:

* + Register them to the Registration system. View the approved/rejected status and the department details entered by the admin.
  + About page which gives detailed description of the Student Registration System with the current registered students.

90% of the development task of the site is completed but while being reviewed it was found that certain features were not implemented properly.

You as a .Net developer have been assigned to fix the missing features that are not implemented in the application.

# Scope of the Case study

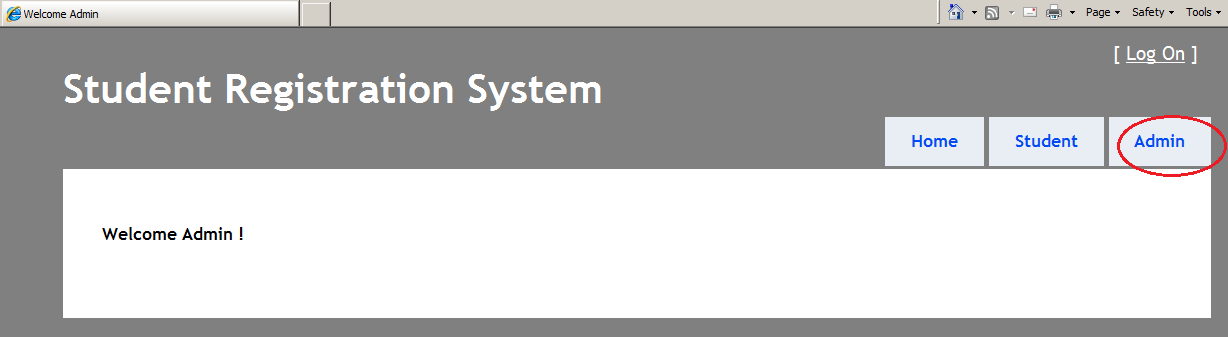
Below files will be impacted while implementing the requirement.

* \StudentRegistration\Controllers\AccountController
* \StudentRegistration\Controllers\HomeController
* \StudentRegistration\Controllers\StudentController
* \StudentRegistration\Controllers\AdminController (New controller has to be created)
* \StudentRegistration\Models\AccountModel (New Model has to be created)
* \StudentRegistration\Models\LoadData
* \StudentRegistration\Models\Student
* \StudentRegistration\Models\StudentEntities
* \StudentRegistration\Resources\HomeResources.fr.resx
* \StudentRegistration\Resources\HomeResources.resx
* \StudentRegistration\Views\ Account\LogOn\
* \StudentRegistration\Views\Admin\Index (New View has to be created)
* \StudentRegistration\Views\Home\
* \StudentRegistration\Views\Shared\
* \StudentRegistration\Views\Student\
* \StudentRegistration\Global.asax
* \StudentRegistration\Web.config
* \StudentRegistration\Content
* \StudentRegistration\Scripts

# Requirement 1: Creating a Controller with default view as Index

**Requirement 1a:**

Create a controller with the name “Admin” with the view named “Index”. Add a menu for the Admin Controller in the Home page. When the Admin menu is clicked, the index view should be rendered and should display the message “Welcome Admin” as shown below



# Requirement 2: Data Validations

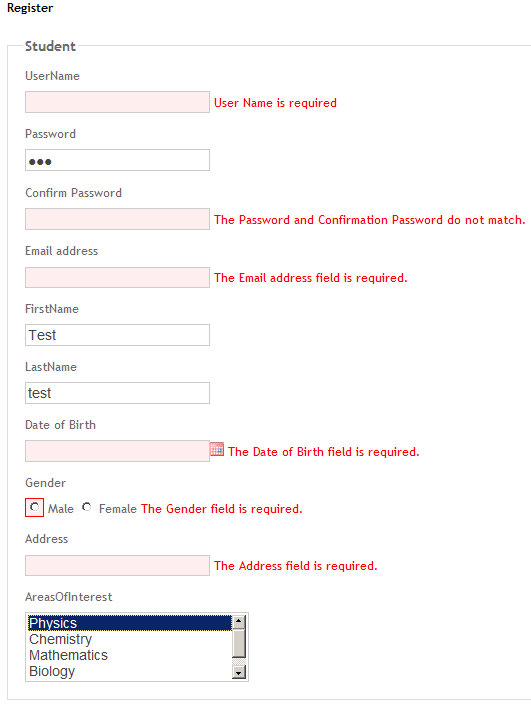
**Requirement 2a:**

Currently in the Register Screen, the students can enter any value for email address and date fields. Validation has to be done for the fields so that email address should be in correct format and valid date should be entered by the Students. Below validations has to be done for the fields.

|  |  |
| --- | --- |
| User Name | Should not be more than 10 digits  *Error Message:*  The field User name must be a string with a maximum length of 10. |
| Password | Entered text should be displayed as \* instead of the actual text. |
| Confirm Password | Entered text should be displayed as \* instead of the actual text.  Error message should be displayed immediately if both the Password and Confirm Password are not same before clicking on the Register button.  *Error Message:*  The Password and Confirmation Password do not match. |
| Email | Should contain @ Symbol. Should be valid email.  *Error Message:*  Invalid Email Address |
| Date Of Birth | Should contain valid date field.  *Error Message:*  Invalid Date Of Birth |
| Address | This should allow user to type only 100 chars max.  *Error Message:*  The field Address must be a string with a maximum length of 100. |

\*All the fields are Mandatory except Last Name.

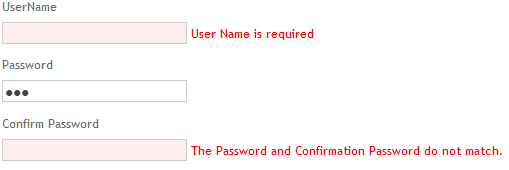
Please find below screenshot with Sample error messages,



**Requirement 2b:**

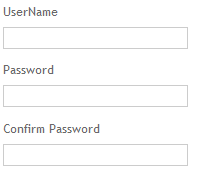
Currently there is no way to reset the values entered by the students. Add a button named “Reset” to reset the entered values in the form. The Reset button should reset not only the entered values but it should also clear the validation error messages if any.

***Screenshot before Reset:***



***Screenshot after Reset:***

All the fields should be cleared along with the error messages.



The reset button should be available in both the Create screen and Edit Screen.