**ASP.NET MVC Final Project**

This document describes the **individual project assignment** for the **ASP.NET MVC** course at ITStep.

**Project Description**

Design and implement an **ASP.NET MVC application**. It can be a discussion forum, blog system, e-commerce site, online gaming site, social network, or any other web application by your choice.

The application should have:

* **public part** (accessible without authentication)
* **private part** (available for registered users)
* **administrative part** (available for administrators only)

**Public Part**

The **public part** of your projects should be **visible without authentication**.

This public part could be the application start page, the user login and user registration forms, as well as the public data of the users, e.g. the blog posts in a blog system, the public offers in a bid system, the products in an e-commerce system, etc.

**Private Part (Users only)**

**Registered users** should have private part in the web application accessible after **successful login**.

This part could hold for example the user's profiles management functionality, the user's offers in a bid system, the user's posts in a blog system, the user's photos in a photo sharing system, the user's contacts in a social network, etc.

**Administration Part**

**System administrators** should have administrative access to the system and permissions to administer all major information objects in the system, e.g. to create/edit/delete users and other administrators, to edit/delete offers in a bid system, to edit/delete photos and album in a photo sharing system, to edit/delete posts in a blogging system, edit/delete products and categories in an e-commerce system, etc.

**General Requirements**

Your Web application should use the following technologies, frameworks and development techniques:

* Use **ASP.NET MVC** and **Visual Studio 2015**
* Have at least **15 controllers**
* Have at least **40 actions**
* You should use **Razor** template engine for generating the UI
  + You may use any JavaScript library of your choice
    - For example Kendo UI widgets (with the ASP.NET MVC Wrappers), Chart.js for charts, etc.
  + Use at least **10 partial views**
  + Use at least **3 editor or display templates**
* Use **MS SQL Server** as database back-end
  + Use **Entity Framework 6** to access your database
  + Using **repositories and/or service layer** is a advisable
* Use at least **2 areas** in your project (e.g. for administration)
* Create **tables with data** with **server-side paging and sorting** for every model entity
  + You can use Kendo UI grid, jqGrid, any other library or generate your own HTML tables
* Create **beautiful and responsive UI**
  + You may use **Bootstrap** or **Materialize**
  + You may change the standard theme and modify it to apply own web design and visual styles
* Use the standard **ASP.NET Identity System** for managing users and roles
  + Your registered users should have at least one of the two roles: **user** and **administrator**
* Use **AJAX form and/or SignalR** communication in some parts of your application
* Use **caching** of data where it makes sense (e.g. starting page)
* Use **Ninject** (or any other dependency container) and **Automapper**
* Write at least **30 unit tests** for your logic, controllers, actions, helpers, routes, etc.
* Apply **error handling** and **data validation** to avoid crashes when invalid data is entered (both client-side and server-side)
* Prevent yourself from **security** holes (XSS, XSRF, Parameter Tampering, etc.)
  + Handle correctly the **special HTML characters** and tags like <script>, <br />, etc.
* Use GitHub

**Project Defense**

Each one will have to make a **public defense** of his work in front of everyone (in 15 minutes). It includes:

* Live **demonstration** of the developed web application (please prepare sample data).
* Explain application structure and its **source code**: C# code, data-bindings, etc.
* Show the **commit logs** in the source control repository .