This document describes the **teamwork assignment** for the **ASP.NET Web Forms** course at Telerik Academy.

# Project Description

Design and implement a **data-driven ASP.NET Web Forms 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) and **administrative part** (available for administrators only).

## General Requirements

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

* Use **ASP.NET One** Web Forms and **Visual Studio 2013**
* Your UI should use **server-side Web Forms** UI rendering (ASPX pages and ASCX controls)
  + ASP.NET MVC and JavaScript UI controls are not allowed
* Use **MS SQL Server** as database back-end
* Use **Entity Framework** to access your database
* Use **data-binding** technique by choice. You are free to use data-source controls (like **EntityDataSource** and **ObjectDataSource**), model binding or manual binding in the C# code behind pages.
* Use at least **two data grids** (table-like data UI components) with **server-side paging** and **sorting**
* Adapt the **default ASP.NET Web site template** from Visual Studio 2013
  + You may use **Twitter Bootstrap** (<http://telerikacademy.com/Courses/Courses/Details/184>)
  + You may change the standard theme and modify it to apply own web design and visual styles
* Use a **Master page** to define the common UI for the public, private and administrative parts
* Use **Sitemap** and navigational UI controls to implement site navigation
* Use the standard **ASP.NET Identity System** for managing **users** and **roles**
  + Your registered users should have are least two roles: **user** and **administrator**
* Use the standard **ASP.NET Web Forms controls** (from **System.Web.UI**)
  + External UI controls from Telerik / Infragistics / DevExpress / etc. are not allowed
* Use **UpdatePanel**s and **AJAX** where applicable to avoid full postbacks
* Apply **error handling** and **data validation** to avoid crashes when invalid data is entered
* Handle correctly the **special HTML characters** and tags like **<br />**
* Use at least **two ASCX user controls** that encapsulate some functionality
* Use at least one **file upload** form to send files at the server side (e.g. profile photos for your users)
* Use **caching** of data where it makes sense (e.g. starting page)
* Use GIT as **source control system**

## 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 (User Area)

**Registered users** should have personal area in the Web application accessible after **successful login**. This area 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.

## Optional Requirements

* Nice looking UI supporting of all modern and old Web browsers
* Good usability (easy to use UI)

## Deliverables

Put the following in a **ZIP archive** and submit it (each team member submits the same file):

* The **source code** (ASPX pages, C# files, images, scripts, styles, etc.)
* **Don't submit the** **NuGet packages**! They are not needed and take too much disk space.
* Optional: brief documentation (few sentences in a **readme** file).

## Public Project Defense

Each team will have to make a **public defense** of its work to the trainers (in 10-15 minutes). It includes:

* Live **demonstration** of the developed Web application (please prepare sample data).
* Explain application structure and its **source code**: ASPX pages, C# code, data-bindings, ASCX controls, etc.
* Show the **commit logs** in the source control repository to prove a contribution from all team members.

## Give Feedback about Your Teammates

You will be invited to **provide feedback** about all your teammates, their attitude to this project, their technical skills, their team working skills, their contribution to the project, etc. The feedback is important part of the project evaluation so **take it seriously** and be honest.