# ASP.NET Web Forms – Practical Exam – September 2014

## News System in ASP.NET Web Forms

You are assigned to design and implement a **News System** where visitors (without authentication) should be able to **view categories** and **articles**, as well as to **register** and **login** in the system.

Registered users (after login) should be able to edit the categories and articles. The system should be implemented as server-side Web application in ASP.NET Web Forms.

### News System Data Layer

Design a simple data layer to hold **users**, **categories, articles** and **likes**.

* **User** has **username** and **password**. The password should be stored in the DB encrypted.
* **Category** has **name** (unique and mandatory).
* **Article** has mandatory **title, content, category, author** and **likes**.
* **Like** has **value**.

Fill the database with the data from SeedData.cs.

3 score

Use the ASP.NET Identity system to keep the users and their encrypted passwords.

1 score

Use **Entity Framework** as ORM engine and **MS SQL Server Local DB** as database storage engine. Your project should run after "copy/paste" deployment, without changing connection strings or other settings. You may use code first, model first or database first approach to access your data from Entity Framework.

1 score

### News System ASP.NET Web Forms Application

Design and implement the News System as a server-side web application in ASP.NET Web Forms. You may use the following steps during your work:

#### Common Features

* **Master Page** – design an ASP.NET Master Pages to reuse the common page elements like headers and footers and navigation in all other pages in the project.

1 score

* **Configure the ASP.NET Identity System** to enable user management functionality (register / login / logout).

1 score

* **Register user** – by username and password the system should be able to register a new user in the system. After successful registration, the user should be redirected to the start page.

1 score

* **Login user** – by username and password the system should be able to login an existing user. After a successful login, the user should be redirected to the start page.

1 score

* **Logout** – successfully logged in user should be able to logout from the system. After a successful logout the start page should be shown.

1 score

#### Public Area

* **Home page** – at the application start page display the 3 most popular articles (top 3 by likes) in the required format (see screenshots or html).  
  List all categories and latest 3 articles for each category (show their title and author).

20 score

* **View article details** – clicking on an article title from the start page should display the article details page - title, content, category, author, date of creation and an option to like/dislike.

5 score

#### Administration Area

* **Categories** – successfully logged in users should be able to create / edit / delete categories. The categories should be shown in sortable table with paging (use page size 5). For each category in the table there should be "edit" and "delete" buttons. The forms implementing create, edit and delete operations could be embedded in the same page below the table or in separate page. When a category is deleted all its books are deleted as well.

Ensure your UI behaves correctly when the users enter invalid data, e.g. too long text in a text field or HTML special characters like "<br/>". Validate the data in your forms.

* **Show, Edit, Delete** – 5 score
* **Insert** – 5 score
* **Sorting, Paging** – 5 score

Total 15 score

* **Articles** – successfully logged in users should be able to create / edit / delete articles. The articles should be shown in the format given (html or screenshots) with paging (use page size 5). Article description should be trimmed to 300 symbols ending with "…". Sorting should work correctly for Title and DateCreated. For each article there should be "edit" and "delete" buttons. The forms implementing create, edit and delete operations could be embedded in the same page below the table or in separate page.

Ensure your UI behaves correctly when the users enter invalid data, e.g. too long text in a text field or HTML special characters like "<br/>". Validate the data in your forms.

* **Show, Delete** – 10 score
* **Edit** – 5 score
* **Insert** – when adding a new article the Author is set to the current user’s name and the date is set to the current date and time - 5 score
* **Paging** – 5 score
* **Sort by Title, Sort by Date** – 5 score

Total 30 score

#### Other Requirements

* **User interface (UI)** – the user interface is intuitive, looks nice and is easy-to-use.

#### 5 score

* **Validation** – validate required data when inserting or editing categories (name) and articles (title, content). There should be appropriate messages.

5 score

#### Advanced Requirements

* **\* Like** – implement the like/dislike functionality with a UserControl. Allow an article to be voted on only once by each user. Only logged-in users can see the control. Use AJAX

#### 5 score

* **\* Edit Articles page** – enable sorting by Category name and by Likes **count** (like value doesn’t matter).  
  hint: you may use System.Linq.Dynamic

#### 5 score

## Evaluation Criteria

The evaluation criteria include: correct and complete fulfillment of the requirements; good technical design and appropriate use of technologies; high-quality code (correctness, readability, maintainability).

## Other Terms

During the exam you are allowed to use any teaching materials, lectures, books, existing source code, and other paper or Internet resources. Direct or indirect communication with anybody in class or outside is forbidden. This includes but does not limit to technical conversations with other students, using mobile phones, chat software (Skype, ICQ, etc.), email, forum posts, etc.

## Exam Duration

Students are allowed to work up to **8 hours**.