# ASP.NET Web Forms – Sample Exam – September 2013

## Poll System in ASP.NET Web Forms

You are assigned to design and implement a **Poll System** where visitors (without authentication) should be able to view random polls and vote, as well as to register and login in the system. Registered users (after login) should be able to edit the questions and answers to each question. The system should be implemented as server-side Web application in ASP.NET Web Forms.

### Poll System Data Layer

Design a simple data layer to hold **users**, **questions** and **answers**. Each user has username and password. The password should be stored in the DB encrypted (not as clear text). Questions are in plain text and hold a sequence of answers. Each answer holds the number of received votes. Fill some sample data in the DB to simplify any further testing. You may use code first, model first or database first approach to store your data.

Use Entity Framework as ORM engine and MS SQL Server Local DB as database storage engine. Your database should be created on demand. If you run the application and the database is missing, your application should create an empty database and populate in it the database schema.

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

Important: your project should run without any installation and configuration.

15 score

### Poll System ASP.NET Web Forms Application

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

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

5 score

* **Configure ASP.NET Membership / Identity System** – configure the SQL Server membership providers for ASP.NET.

5 score

* **Display random questions** – the system should display **3 random questions** along with their answers. Ensure you handle properly any special HTML characters, e.g. a question could say "*Do you know how to work with the <div> and <span> tags in HTML?*" and should be displayed correctly.

5 score

* **Vote** – the system should allow selecting one of the answers for some of the questions currently shown and voting for it. After successful voting, the "Show voting results" page should be shown.

5 score

* **Show voting results** – after a new vote is received for a certain question, the system should show the voting results for it. Use a simple bar chart that visualizes the percentages for each answer.

5 score

* **Register user** – by username and password the system should be able to register new user in the system. After successful registration, the user should be redirected to the "View all results" page. In case of error, an appropriate error message should be displayed and the user should be able to try to register again.

5 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 "View all results" page.

3 score

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

2 score

* **View all results** – successfully logged in users should be able to view all questions in the system and the voting results for each question. When a certain question is selected, its voting results should be shown below.

10 score

The questions should be shown in alphabetical order with paging (use page size 3 to simplify testing).

5 score

* **Add / Edit / Delete questions** – successfully logged in user should be able to add / edit / delete questions. Ensure your UI behaves correctly when the users enter invalid data, e.g. too long text in a text field. Validate the data in your forms.

10 score

* **Add / Edit / Delete answers** – successfully logged in user should be able to add / edit / delete answers for certain question. Ensure your UI behaves correctly when the users enter invalid data, e.g. too long text in a text field. Validate the data in your forms.

15 score

* **Error handling** – in case of error (e.g. Internet connection lost, DB connection lost, incorrect request, etc.), an appropriate error message should be displayed. You are free to decide how exactly.

10 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).

To pass the exam you need to gain at least **70 score** (out of 100 score total).

## 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**.