Contents

[Online (10 Marks) 1](#_Toc509899974)

[GitHub (10 Marks) 1](#_Toc509899975)

[Functional Requirements Implementation (30 Marks) 2](#_Toc509899976)

[API 2](#_Toc509899977)

[Documentation (10 Marks) 2](#_Toc509899978)

[Formatting (20 Marks) 3](#_Toc509899979)

[Packaging (15 Marks) 3](#_Toc509899980)

[Presentation – Not for Marks 3](#_Toc509899981)

[Deductions 3](#_Toc509899982)

|  |  |
| --- | --- |
|  |  |
| Due Date: | April 16, 12:00PM (Archived project uploaded to D2L. A text file with a link to the live version is also included. )  April 16, 1:00PM (Project presentations start) |
| Submission Location: | Project Phase 3 Drop Box and Online |
| Submission Details: | Include all deliverables inside a folder named with the members of your team \_phase3. Your deliverable includes:   1. One code project. 2. A link to your online site in a text file.   Marks will be deducted for additional documents submitted.  Delete your git folder from your project before submitting it to reduce the project size. |
| Late Penalty: | 20% deducted each day this assignment is late. |

### Online (10 Marks)

* At the login page, please show a summary of user names and logins for each role which your project users need for testing.

### GitHub (10 Marks)

I will be looking for significant activity and regular activity from each person on the team on GitHub. Ensure that all team members are using GitHub. I expect two commits per project day as a minimum but really there should be more. Remember to start each code session with a fresh build. As well, occasionally set aside your project outside Git and zip it up so you have a safe copy in case the GitHub repository gets corrupted.

### Functional Requirements Implementation (30 Marks)

A successful application:

* Implements all essential features that make it commercially viable.
* Uses the Repository Model and View Model patterns.
* Uses the identity security model to manage users, roles, and passwords.
* Allows users to email themselves a reset password if they forgot their current one.
* Allows users to update their passwords when they are logged in.
* Allows users to register and they must confirm their registration by email before logging in.
* Properly validates inputs and outputs if the data source is not trusted both on the client and especially the server.
* Uses image verification such as Recaptcha on any page that enables account creation or password reset externally.
* Provides helpful error messages to guide users while not revealing sensitive information.
* Protects against brute force attacks. In addition to using the Identity security model to freeze accounts for 10 minutes after a specific number of failed login attempts, be sure to add a 1 second delay between the time a login request is made and the response.
* Provides an area with simple but complete information on how to use the API on your site. Include easy-to-use JavaScript samples in this area for third party developers to use to connect to your API from a different domain.

### API

As outlined in the requirements for phase 2, your API:

* Implements an API to serve JSON to 3rd party developers to help promote the use of your application.
* The API is documented in your design documents.
* The API is documented on the site so any third-party user can use it.
* Be sure to include a JavaScript code sample on your site so the developer can easily embed this script in their application to obtain data from your site.

### Documentation (10 Marks)

Also at your site, please keep a link to your most current design document which shows:

* A brief description about the project and the problem you are trying to solve.
* Functional requirements listed under essential, important and nice-to-have categories.
* The USE Case diagram.
* The ERD.

### Formatting (20 Marks)

* Focus on content, control size and alignment, intuitiveness, mobile responsiveness and professional appearance. Data must look realistic.

### Packaging (15 Marks)

In addition to hosting your application online, submit your code project to the drop box along with a step-by-step document which explains how to set up your application. Remember it is really important to make it easy for your stake holders to install the application. Incomplete documentation or broken software will waste their time and they will not be happy.

In the upload to D2L, be sure to provide clear instructions on how to build and deploy the application for its first run. The stakeholder will need this. Please make it easy for your stake holder. This entire mark could be lost if the project cannot be deployed due to some error or oversight so please double check your work and have more than one team member test this.

### Presentation – Not for Marks

Presentations will start at 1:00PM on April 16. Each team has 10 to 20 minutes to speak. Show us a sound working product. Suggested features to highlight are:

* A description of why you built the project and the problem you are trying to solve.
* The use case diagram which explains who the stakeholders are and their needs. (Ensure this looks professional and is easily displayed on an overhead.)
* A description of the code platform and any extra libraries or frameworks that you may have used. If you used other frameworks beyond the standard ASP.NET and SQL Server tell us why.
* The essential features in your online application.
* The API.
* Outstanding features that did not get implemented but could be added in later.

To avoid getting too technical or detailed, I recommend that you do not show:

* Your ERD.
* Your SQL.
* Your code.
* How to create an account or reset your password or similar features that are just standard for any site. Please have your accounts created ahead of time.

# Deductions

* Not uploading one code project to D2L.
* Marks will be deducted for code duplication, non-standard code practices, and poor presentation.
* Submitting more than one project within your final solution and submitting multiple design document files.
* All team mates are expected to be present before 9:15AM or a small deduction will apply.
* Not completing SCRUM reports each week.
* Missing or inaccurate links from scrum report.
* Not keeping the “functional build-wire frame prototype hybrid” relatively current so all stakeholders can quickly check the latest progress and design (Please update this at least once a week or more).
* Not updating the last publish date and time label on the public facing site.