This is a Collaborative Learning Community (CLC) assignment.

**Objective**

Demonstrate the ability to analyze an existing desktop application and determine what software components could be reused or refactored to build a web-based application.

Demonstrate the following Agile Scrum practices:

1. Complete Daily Standups
2. Write User Stories
3. Create a Sprint Product Log
4. Create a Sprint Back Log
5. Create a Sprint Burn Down Chart
6. Complete a Retrospective

Demonstrate basic knowledge for how to build Views and Controllers using .NET MVC using Visual Studio.

Demonstrate basic knowledge for how to design and build tables using SQL Server (Express) in Visual Studio.

Leverage prior C# Object Orientated programming techniques and ADO.NET to build a simple Form and persist its contents in a relational database.

Continue gaining experience using Agile Scrum practices.

**Activity**

Scrum Planning:

1. Identify team member.
2. Hold daily standups with your team members.
3. Review project requirements (both functional and nonfunctional).
4. Analyze existing Minesweeper code and determine what code can be reused or refactored for this project.
5. Review the Registration and Login pages requirements and develop some UI wireframes or mock diagrams.
6. Write user stories required to support the Registration and Login requirements then update the Product Back Log.
7. Complete Sprint 1 Planning and Estimation and update the Sprint 1 Back Log.
8. Complete the Sprint 1 Burn Down Chart.
9. Complete the initial Design Report.
10. Hold a "Practice" Retrospective for this assignment.

**Application**

Review the registration and login pages you created in prior programming classes.

Build a registration form that meets the following minimum requirements:

1. Takes First Name, Last Name, Sex, Age, State, Email Address, Username, and Password as Form Fields.
2. Use an HTTP POST to a Controller.
3. Perform Form Validation on the server side.
4. Save the Form Fields in a SQL Server relational database.
5. Forward the results of the Form Submit to a success or error page.

Build a Login Form that meets the following minimum requirements:

1. Takes Username and Password as Form Fields.
2. Uses the SQL Server database for authentication.
3. Forward the results of the Form Submit to a success or error page.

Hold peer code reviews for all code that was developed.

Hold an Agile Scrum Retrospective at the end of this Sprint (and document your results in the Design Report or Mind Mapping Tool).

Analyze the Default .NET MVC application to understand how bootstrap and master pages are organized and leveraged.

Analyze future functional and technical requirements to ensure that the design has considered future application functionality.

**Build**

1. In SQL Server, build the necessary tables to store the information required during the registration and login processes.
2. In HTML and ASP.NET, create the Registration and Login View pages to input user data and POST the form requests.
3. In C# and .NET MVC, create the Registration and Login Controllers to handle and process the form requests.
4. In C# create the appropriate data validation rules within the Controller's implementation.
5. In HTML and ASP.NET, create the appropriate Response View pages.
6. Maintain all code across the team by checking code and syncing code daily into the GIT Repository.

**Deliverables**

*Scrum Planning:*

1. Initial Sprint Product Log
2. Initial Sprint Back Log
3. Initial Sprint Burn Down Chart
4. Design Report

*Application*

1. Fully functional Registration Page and a fully functional Login Page.
2. Updated Sprint Product Log
3. Updated Sprint Back Log
4. Updated Sprint Burn Down Chart
5. Results of Scrum Retrospective (in the Design Report or Mind Mapping Tool).
6. Design Report (with ER diagram, Class diagrams, etc.)
7. All code artifacts and design artifacts (with GIT URL's to the above planning artifacts) checked into the GIT Repository.

**What to Submit:**

1. The above Scrum artifacts in the deliverables section uploaded to a GIT Repository, and in a folder called Planning and Design.
2. Design Report (with GIT URL's to the above artifacts).
3. The Design Report uploaded to LoudCloud.
4. All code artifacts checked into a GIT Repository, and in a folder called Application.
5. All planning artifacts, design report, and code zipped into a single file and uploaded to LoudCloud.

GCU style is not required, but solid academic writing is expected.

This assignment uses a scoring guide, provided by the instructor. Please review the rubric prior to beginning the assignment to become familiar with the expectations for successful completion.

You are not required to submit this assignment to LopesWrite.