This is a Collaborative Learning Community (CLC) assignment.

**Objective**

1. Demonstrate basic knowledge for how to build an N-Layer web application using Controllers, Models, Business Services, and Data Access Services.
2. Demonstrate basic knowledge for how to use the Razor Syntax to build dynamic Views in .NET MVC using Visual Studio.
3. Demonstrate basic knowledge for how to use the Razor HTML Helpers to build dynamic Views in .NET MVC using Visual Studio.
4. Demonstrate basic knowledge for how to create databases, tables, and columns in SQL Server Express.
5. Leverage prior C# Object Orientated programming techniques and ADO.NET to build a simple Form and persist its contents in a relational database.
6. Continue gaining experience using Agile Scrum practices.

**Activity**

Review the Minesweeper Game requirements from your prior programming class to understand the Game Logic and Game Rules for Minesweeper.

Build the Minesweeper Game Board:

1. Login and Register functionality available via the Main Menu.
2. Game Board available after a user logs into the application successfully.
3. Uses Razor Syntax and appropriate HTML Helpers to display the Game Board.
4. Uses appropriate backend Controllers and Business Services to implement the Game Logic and Game Rules.
5. As necessary refactor the Login and Registration Forms from Topics 1: (a) Perform Form Validation on the server side. (b) Save the Form Fields in a SQL Server relational database. (c) Forward the results of the Form Submit to a success or error page.

Hold peer code reviews for all code that was developed.

Hold Agile Daily Scrum meetings on a frequent and planned schedule (ideally a daily frequency).

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

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

**Build**

1. In HTML, Razor C#, and ASP.NET MVC, create the Game Board pages.
2. In C# and .NET MVC, create the Game Board Controllers to handle and process the View requests.
3. In C# create the appropriate backed Services to support the all Game Logic and Game Rules.
4. Refactor any necessary updates to the tables to store the information required during the registration and login processes.
5. Refactor code as identified during peer code reviews and instructor feedback (or place refactor work as future tasks in Scrum Back log).
6. Maintain all code across the team by checking code and syncing code daily into the GIT Repository.

**Deliverables**

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

**What to Submit:**

1. All planning and design artifacts updated in a GIT Repository, and in a folder called Planning and Design.
2. All code artifacts checked into a GIT Repository, and in a folder called Application.
3. 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.