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 AJAX to build responsive and dynamic Views in .NET MVC using Visual Studio.
3. Demonstrate basic knowledge for how to use Partial Views and Partial Page Updates in .NET MVC using Visual Studio.
4. Demonstrate basic knowledge for how to design and build a REST based service using Windows Communication Foundation.
5. Continue gaining experience using Agile Scrum practices.

**Activity**

1. Ability to save and resume playing a game.
2. Ability to save game results (number of clicks and time spent playing game).
3. Ability to publish game results over a REST based service.
4. Hold peer code reviews for all code that was developed.
5. Hold Agile Daily Scrum meetings on a frequent and planned schedule (ideally a daily frequency).
6. Hold an Agile Scrum Retrospective at the end of this Sprint (and document your results in the Design Report or Mind Mapping Tool).
7. Cleanup any code from prior topics code reviews and instructor feedback.

**Build**

1. In C# and ADO.NET, create backend Service classes to resume a game and save game results.
2. In C# and WCF, create a REST based web service to retrieve game results.
3. Refactor any code and database tables to support any new functionality or improve the application design.
4. Refactor code as identified during peer code reviews and instructor feedback (or place refactor work as future tasks in Scrum Back log).
5. Maintain all code across the team by checking code and syncing code daily into the GIT Repository.

**Deliverables**

1. Fully functional responsive .NET Minesweeper Application (Login, Registration, Game Board, Save/Restore Game Progress)
2. REST API's to consume game results
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. Final 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.