**CST-350 Project Status and Design Report**

**.NET Application Programming**

|  |  |  |
| --- | --- | --- |
| **Topic:** | Topic 2: Views in .NET MVC | |
| **Date:** | *3/26/23* | |
| **Revision:** | *1.0* | |
| **Team:** | 1. *Michael Melichar* | |
| 1. Luwam Tesfay | |
|  | |
|  | |
| **Milestone Task Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *As a user, I want to be able to fill out a registration form with my First Name, Last Name, Sex, Age, State, Email, Username, and Password.* | *Michael* | *1* | *0* | | *As a user, I want the registration form to use an HTTP POST to submit my information to a controller.* | *Michael* | *1* | *0* | | *As a user, I want the registration form to perform server-side form validation to ensure my information is correct.* | *Michael* | *1* | *0* | | *As a user, I want my information to be saved in a SQL Server relational database when I submit the registration form.* | *Michael* | *1* | *4* | | *As a user, I want to be forwarded to a success or error page after submitting the registration form.* | *Michael* | *1* | *1* | | *As a user, I want to be able to fill out a login form with my Username and Password.* | *Michael* | *1* | *0* | | *As a user, I want the login form to use the SQL Server database for authentication.* | *Michael* | *1* | *4* | | *As a user, I want to be forwarded to a success or error page after submitting the login form.* | *Michael* | *0* | *1* | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **Repository URL:** | *https://github.com/melichar-m/cst350/tree/main/Milestone-CST350* | |
| **Peer Review:** | *Y/N?* | We acknowledge that our team has reviewed this report and we agree to the approach we are all taking. |

**Planning Documentation**

**Agile Scrum Product Backlog:**

*Backlog so far includes getting the MySQL server linked to the application and receiving user data to then check against future logins for authentication.*

**Agile Scrum Sprint Backlog:**

*This needs to contain a URL to Scrum Sprint Backlog.*

**Agile Scrum Burn Down Chart:**

*This needs to contain a URL to a Scrum Burn Down Chart.*

**Agile Retrospective Results:**

*The following table should be completed after each Retrospective on things that went well (keep doing).*

|  |
| --- |
| **What Went Well** |
| Completing just about any work this week is a plus considering our pair didn’t meet and barely posted until the 11th hour. Getting about half of the work done is a miracle to me. |
|  |
|  |

*The following table should be completed after each Retrospective on things that didn’t go well (stop doing) and what would be done differently next time. An Action Plan is a short statement describing what you will do differently. The due date for the plan is when you will implement the change.*

|  |  |  |
| --- | --- | --- |
| **What Did Not Go Well** | **Action Plan** | **Due Date** |
| **Communication** | **Communicate earlier in the week** | **3/27** |
|  |  |  |
|  |  |  |

**Design Documentation**

**Install Instructions:**

*Step-by-step instructions for setting up your database and configuring, and deploying/installing your application. This section should also include detailed instructions for what configuration files are required by your application, what configuration settings need to be adjusted for various runtime (development or production) environments, and where the files need to be deployed to.*

**Key Technical Design Decisions:**

*Any final technical design decisions, such as framework decisions, addressing its purpose in the design and why it was chosen.*

**ER Diagram:**

*The ER diagram shows the design of database tables and foreign key relationships. Include an image file of your ER database diagram.*

**DDL Scripts:**

*The DDL script is the SQL export text file from a database. This should contain a link to a private repository wherefrom the DDL script can be downloaded.*

**Sitemap Diagram:**

*The Sitemap shows a navigation path that the user can take through the application. Include an image file of your Sitemap diagram.*

**Security Design:**

*This section should outline the design for how authentication and authorization was supported. This section should also contain all of the roles and privileges that are supported by the design.*

**Third Part Interface Design:**

*This section should fully document any Third Party Service Interface API’s, how to access the service, what parameters are required by the API, and the detailed JSON data format specification that could be used by a third party developer to integrate with the service and API.*

**Flow Charts:**

*You should insert any flow charts here. Flow charts should document algorithms or workflow that will be implemented in your program. At a minimum, this should contain a flow chart of the Minesweeper game logic.*

**User Interface Diagrams:**

*You should insert any wireframe drawings or whiteboard concepts that were developed to support your application.*

**Class Diagrams:**

*You should insert any class diagrams here. Your class diagrams should be drawn correctly with the three appropriate class compartments, + and – minus to indicate accessibility, and the data types for the state/properties, as well as method arguments and return types.*

**Pseudo Code:**

*You should provide a URL references to any code stubs and pseudocode. If you have no supporting documentation, please explain the rationale for why you are able to leave this section as N/A.*

**Other Documentation:**

*You should insert any additional drawings, storyboards, whiteboard pictures, project schedules, tasks lists, etc. that support your approach, design, and project. If you have no supporting documentation, please explain the rationale for why you are able to leave this section as N/A.*