COMP229 – Advanced Web Application Development

Team Project 1 – Game Tracker - Value 20%

Part 1 (Project Concept & Setup) – Value 5% – 17 Marks

- Initial Project Structure Setup (4 Marks: Site Structure).
- Page Comment Headers (2 Marks: Internal Documentation).
- First Draft of External Document (5 Marks: External Documentation).
- GitHub Repo (2 Marks: Version Control).
- Cloud Provider Setup (4 Marks: Cloud Provider).

Due Week # 5 (Friday October 7, 2016) @ midnight

Part 2 (Main Game Tracking Page, DB & Partial Function) – Value 5% – 17 Marks

- One Game on Main Page (one Game tracked / week) (5 Marks: GUI, 5 Marks: Functionality)
- Main Game Tracking Page Setup (1 Marks: Site Structure).
- Database Structure Setup (1 Mark: Site Structure)
- Method Comments & Contextual Variables (3 Marks: Internal Documentation)
- Updated External Document (1 Mark: External Documentation)
- Updated GitHub Repository (1 Mark: Version Control).

Due Week # 6 (Friday October 14, 2016) @ midnight

Part 3 (Full Functionality & Site Security) – Value 10% – 34 Marks

- Site Security and Admin Pages (6 Marks: GUI, 6 Marks: Functionality)
- Full Site Functionality – 4 Weekly Games (8 Mark: GUI, 8 Marks: Functionality)
- Updated Responsive Design and Structure (2 Marks: Site Structure).
- Add any additional Inline Comments (1 Mark: Internal Documentation).
- Final Version of External Document (2 Marks: External Documentation).
- Updated GitHub Repository (1 Mark: Version Control).

Due Week # 7 (Friday October 21, 2016) @ midnight

Team Project Project 1 – Game Tracker **Maximum Mark: 68**

Overview: Working with a partner (or solo), you will build an **ASP.NET Web Forms Application** that will track 4 different games or contests for each calendar week. Your **Main Game Tracking Page** will display the **weekly statistics** of each game. Each game will track **two teams** and your web application will record which team won the game. Your Web App will also measure how many points were **scored** and how many points were **allowed** (or lost) for each team and how many **spectators** watched each game. You must ensure that you deploy appropriate User Input Validation throughout your web application. The weekly results will be recorded in a database using appropriate tables and relations.

Instructions :

Your Web Application will include the following requirements and constraints:
(19 Marks: GUI, 19 Marks: Functionality, 8 Marks: Site Structure, 6 Marks: Internal Documentation, 8 Marks: External Documentation, 4 Marks: Version Control, 4 Marks: Cloud Hosting)

1. **Main Game Tracking Page (13 Marks: GUI, 13 Marks: Functionality):**
    a. This Main Game Tracking Page will allow a **registered user** who is **logged in** to enter the statistics and data related to 4 games or contests held between two teams during any calendar week (2 Marks: GUI, 2 Marks: Functionality).
    b. Game statistics and data will be displayed within various HTML elements and ASP.NET server controls contained in the page. All Controls will be appropriately styled and positioned to enhance the overall user experience (2 Marks: GUI, 2 Marks: Functionality).
    c. During each calendar week, each **Game** will include a **name**, a **short description**, **total points scored** by both teams combined, the **number of spectators** and a designated **winning team**. (3 Marks: GUI, 3 Marks: Functionality).
    d. During each calendar week, each **Team** will include a **name**, **a short description**, **total points scored** for a weekly game, total points allowed (lost) for a weekly game (3 Marks: GUI, 3 Marks: Functionality).
    e. After all data is entered for each game and team for each Calendar week, a **registered user** who is **logged in** can use a button, link or other control to submit the weekly game and team statistics to be recorded on the database server (2 Marks: GUI, 2 Marks: Functionality).
    f. Anonymous users may view any calendar week including Game and Team statistics but will prevented from making any changes (1 Marks: GUI, 1 Marks: Functionality).
2. **Site Security.** Your web application must be secure **(6 Marks: GUI, 6 Marks: Functionality):**
    a. **User Registration** must be included. A form will allow the user to enter profile information (**username**, **password, email address, etc.),** which will be stored in a database table (2 Marks: GUI, 2 Marks: Functionality).
    b. A form will allow a **registered user** to **Login**. (1 Marks: GUI, 1 Marks: Functionality).
    c. Upon successful **Login**, other forms or controls will allow the user to **Logout** and / or **modify** his or her profile (2 Marks: GUI, 2 Marks: Functionality).
    d. **Site security** will prevent non-registered users from modifying any Game or Team statistics on any Calendar week. (1 Marks: GUI, 1 Marks: Functionality).
3. **Web Application Structure.** Your Web App must be designed using well-structured semantic HTML, CSS, JavaScript and implement a responsive front-end framework (e.g. Bootstrap) along with a dynamic back-end (using ASP.NET and MS Azure) **(8 Marks: Site Structure):**

   a. Your Site must be **responsive,** and adapt to various viewport sizes. **Note:** Using a mobile-first framework such as Bootstrap is highly recommended (2 Marks: Site Structure).
   b. Your **CSS** rules reside in separate file(s) in the **Content** folder and adhere to best practices. You may use SASS and Compass for additional functionality (1 Mark: Site Structure).
   c. Your **JavaScript** files, libraries (and other external code) are contained in **Scripts** folder and are appropriately linked to your site. **Note:** It is highly recommended that you use CDNs where you can in the development release of your App (2 Marks: Site Structure).
   d. Your **images and multimedia assets** are contained in the **Assets** folder and appropriately linked to your site. **Note:** You may need to provide additional versions of your multimedia assets to accommodate various viewport sizes (1 Mark: Site Structure).
   e. All Your Code (HTML, CSS, JavaScript, jQuery, C#, etc.) is error free (1 Marks: Site Structure).
   f. Your SQL database must be hosted online. **Note:** It is required that you use Microsoft Azure as a Cloud Provider (1 Marks: Site Structure).

4. **Internal Documentation** for your site **(6 Marks: Internal Documentation):**
   a. Ensure you include a **Page Comment header** for your **HTML**, **CSS, C#, and JavaScript files** that indicate: The **Author's name**, **Student Number**, **Date Modified**, **Short Version History**, **and File Description** (2 Marks: Internal Documentation).
   b. Ensure you include **Method Comment headers** for all of your **HTML structure, CSS style sections, C# classes** and **methods**, and any **JavaScript functions**. It is recommended that all comment C# and JavaScript comment headers adhere to YUIDoc or JSDoc syntax (2 Marks: Internal Documentation)
   c. Ensure all your code uses **contextual variable names** that help make the files human-readable (1 Marks: Internal Documentation).
   d. Ensure you include **inline comments** that describe your GUI Design and Functionality. **Note**: please avoid “over-commenting” (1 Marks: Internal Documentation)
5. Create an **External Document** for your Web App that includes **(8 Marks: External Documentation)**:
   a. A **team Logo** (1 Marks: External Documentation).
   b. **Table of contents** (1 Marks: External Documentation).
   c. A **Detailed description** of your Web App including its core functionality (1 Mark: External Documentation).
   d. A **Version History** of your Web App. A screen capture of your GitHub Commit history and a link to your GitHub Repository is required (1 Mark: External Documentation).
   e. A **Typography** and **Colour** Section that describes any fonts and colours you will use for your Web App. Include Colour Swatches with Hex or RGB codes (1 Mark: External Documentation).
   f. A **Wireframes Section** that includes a wireframe image and appropriate arrows and labels for each page template of your Web App (2 Marks: External Documentation)
   g. **Screen Capture Section** that includes Screen Shots (samples) of each of your site’s templates. (1 Marks: External Documentation).
6. Share your files on **GitHub** to demonstrate Version Control Best Practices and push your site to a cloud host **(4 Marks: Version Control, 4 Marks: Cloud Hosting).**
   a. Your repository must include **your code** and be well structured (2 Marks: Version Control).

b. Your repository must include **commits** that demonstrate the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).

c. You must deploy your site to your Cloud Server using **git** (4 Marks: Cloud Hosting).

SUBMITTING YOUR WORK

Your submission should include:

1. An external document (MS Word or PDF).
2. A link to your GitHub Repo
3. A link to your cloud provider live site
4. A zip archive of your website's Project files submitted to blackboard

Feature	Description	Marks
GUI / Interface Design	Display elements meet requirements. Appropriate spacing, graphics, colour, and typography used.	19
Functionality	Site deliverables are met and site functions are met. No errors, including submission of user inputs.	19
Site Structure	Your site files are well organized. Your HTML and CSS are kept in separate folders. All external documents and code are appropriately linked to your site. You code is minified where possible. Your code is error free.	8
Internal Documentation	File header present, including site & student name & description. Functions and classes include headers describing functionality & scope. Inline comments and descriptive variable names included.	6
External Documentation	An external document (MS Word or PDF format) has been created that includes a company logo, table of contents, brief site description, etc.	8
Version Control	GitHub commit history demonstrating regular updates.	4
Cloud Deployment	Deploy site to Cloud Service.	4
Total		**68**

This assignment is weighted **20%** of your total mark for this course.

All Assignments are due at the beginning of class.
Late submissions:

- 10% deducted for each day late.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.