TEESSIDE UNIVERSITY

SCHOOL OF COMPUTING, ENGINEERING AND DIGITAL TECHNOLOGIES

**Web Apps and Services (CIS2021-N) - ThAmCo Events Assessment Element Two Checklist**

| Functional Requirements | Status | Notes |
| --- | --- | --- |
| **General** |  |  |
| * Starter Project has been copied | ✔ |  |
| * Read Me file is created with key information (as a minimum the Project Name and Description should be included). | ✔ | Instructions of how to get the web services and the web application up and running |
| * Code documentation (Read Me) identifies the key problems and challenges of the assignment and details alternative solutions to them. | ✔ |  |
| * Justification (Read Me and/or code comments) is given for choices made with reference to suitable knowledge sources. | ✔ | https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0 |
| * Appropriate code comments are included | ✔ | Inline code comments |
| * Documentation (Read Me or code comments) is short because it is clear and concise. | ✔ |  |
| * **Customise the user interface and the application workflow** beyond that provided by the ASP.NET scaffolding. | ✔ | Added styling in **site.css** file in each static root folder for each web app, and in the **\_Layout.cshtml** stylesheet in the shared views folder. |
| **MUST Requirements** |  |  |
| * Web Services (ThAmCo.Catering) to create, edit, delete and list food items - see the ERD above for details; | ✔ | **<https://localhost:7173/menu>/<MenuId>**  View the menu list and the food items associated with a specific menu associated with that **MenuId** |
| * Web Services (ThAmCo.Catering) to create, edit, delete and list the details of food Menus - see the ERD above for details; | ✔ | **<https://localhost:7173/menu>**  View the available menus, and perform the CRUD operations |
| * Web Services (ThAmCo.Catering) to add and remove a food item from a menu - see the ERD above for details; | ✔ | Each food item in the table of food items include a **DELETE** link used to discard that particular food item from the records |
| * Web Services (ThAmCo.Catering) to book, edit and cancel Food for an Event - see the ERD above for details. The service should return the FoodBookingId as confirmation of the booking; | ✔ | This service serves the purposes of both ASP.NET ApiControllers and the ViewControllers to accomplish these obligations |
| * Create, list and edit Guests; | ✔ | <https://localhost:7004/Guests>  Lists and provides additional endpoints for editing against each guest |
| * Create a new Event, specifying as a minimum its title, date and EventType; | ✔ | To create a new guest:  <https://localhost:7004/Guests/Create> |
| * Edit an Event (except its date and type); | ✔ | Editing a specific event exclusive of date and time:  **<https://localhost:7004/Events/Edit/><EventId>** |
| * Book a Guest onto an Event; | ✔ | In the table of events,each entry has an endpoint for booking guests for that particular event. |
| * List Guests (including a total count) for an Event and register their attendance; | ✔ |  |
| * Display the details of a Guest, including information about the Events with which they are associated and their attendance; | ✔ | The user can view all the Guests associated with a specific Event and further perform CRUD operations on that Guest via the **GuestsController** |
| **SHOULD Functional Requirements** |  |  |
| Via the web app, the user should be able to: |  |  |
| * Cancel the booking of a Guest from an upcoming Event; | ✔ |  |
| * Reserve an appropriate, available Venue for an Event via the *ThAmCo.Venues* web service, freeing any previously associated Venue; | ✔ | Able to reserve and free venues for an event |
| * Display a list of Events that includes summary information about the Guests and Venue within it; | ✔ | The **EventsController** includes respective views for listing Events, Guests and available venues |
| * Create, list and edit Staff; | ✔ | The Staff Controller provides the functionality of creating, listing and editing Staff |
| * Adjust the staffing of an Event, adding available staff or removing currently assigned staff; | ✔ | Using the “**AssignStaff**” and “**UnassignStaff**” endpoints, one is able to assign and free staff associated with a specific event |
| * See appropriate warnings within event list and staffing views when there is not a first aider assigned to an Event; | ✔ |  |
| * Display the details of a Staff member, including information about **upcoming** Events at which they are assigned to work; | ✔ | The staff controller provides views for viewing the details of staff |
| * Cancel (soft delete) an Event, freeing any associated Venue and Staff; | ✔ | The Events List includes a n endpoint against each event for soflt deleting an event |
| WOULD Functional Requirements |  |  |
| The user should be able to: |  |  |
| * Via the web app, the user should be able to: |  |  |
| * Display the details for an Event, which must include details of the Venue, Staff and Guests – this should be more detailed that the summary information found in the Event list; | ✔ | All guests associated with an event can be accessed via the provided endpoints against each Event |
| * Permanently remove personal data by anonymising their Guest entity; | ✔ | Deleting a Deleting a Guest via the Delete Endpoint permanently removes a Guest including all the data he/she is associated with |
| * Display a detailed list of available Venues, filtered by EventType and date range, and then create a new Event by picking a result; | ✔ | The Events Controller’s Index view includes a Venues endpoint in the navigation bar to access all the available venues, with included customized filtering using dates and Event Types |
| * See appropriate warnings within event list and staffing views when there are fewer than one member of staff per 10 guests assigned to an Event. | ✔ | A tick is shown if this is not the case, but the text “**Not enough staff**” is shown in red |
| * User access control should restrict (see assignment for details) |  |  |
| Report |  |  |
| * Prepare a short (500-word) report to evaluate the completeness of the solution, own working practices, and highlight security features built or planned for the app/service. | ✔ | This report provides a summary of the three services and outline all the endpoints to be used to access different services offered.  Security features that may need implementation and some already implemented also outlined. |