## Ms Web Tech – Final Exam

You're working on a membership web site for a sailing club, currently doing edits on the ***Member*** table.

1. Your initial project and database script is on your exam account's G: drive. Work on C: or G: but **only what's on the exam account's G: drive can be marked.** You're responsible for copying the project from C: to G:.
    a. Create the *sail* database from the SQL script. The models, context, and Session support have already been added to the project.
    b. In ***_layout.cshtml***:
        i. Add a menu entry for the ***Member*** controller
        ii. Display the TempData variable "message" before the guest page, in bold & red.
        iii. Put your name and section in the footer.
2. Generate the ***Member*** controller with full CRUD support for the *member* table.
    a. In the Create view, replace the *ProvinceCode* drop-down with a textbox and add a span tag for any field validation errors. Remove *FullName* from the view … it's derived later.
    b. On Edit, the province drop-down should display the province name, sorted
    c. In the Index & Edit views, display the *MemberId*, but don't let the user modify it.
    d. Use a try-catch in all actions that modify database tables:
        i. Display a message to the user if the action succeeds.
        ii. Display the innermost exception's message if an exception is thrown.
3. ***Member*** edits must be centralised in the model except for Remote & custom annotation code:
    a. Do not apply edits or annotations directly to the generated model or to controller actions.
        i. Use a metadata class to apply them to the model through a like-named partial class. **[ModelMetadataType(typeof(xxx))]** might be helpful.
        ii. Add the following to the top of the partial class, to provide access to database tables: **SailContext _context = new SailContext();**
    b. The *memberId* is manually entered (not an autonumber). Ensure that users don't try to create a duplicate on file … but allow them to edit an existing member.
    c. The *fullName* and the member's *firstName* and *lastName* are required. ***fullName*** should be on all views except Create but, as a generated field, it should be protected from change.
    d. The *provinceCode* is required and must be exactly 2 letters long.
        i. Accept upper or lower case, but force it to upper case before writing to the database.
        ii. Verify the *provinceCode* using the province table.
    e. The *postalCode* must match a Canadian postal pattern … don't worry about forbidden letters.
        i. Accept upper or lower case, with an optional space
        ii. Shift to upper case and add the space (if missing) before writing to the database
    f. The *homePhone* is optional but if provided, must be 999-999-9999, with or without the dashes.
        i. Insert the dashes before writing to the database.
    g. If there is no spouse, the *fullName* should be like this: "Turton, David". If there is a spouse:
        i. If the spouse's last name is the same or is not provided, *fullName* should be like this: "Turton, David & Diana"
        ii. If the spouse's last name is different, *fullName* should be like this: "Turton, David & Rigg, Diana"
    h. If *useCanadaPost* is true, *Street* & *city* are required, *email* is optional (Some members do not have a good Internet connection). If *useCanadaPost* is false, *email* is required, *street* & *city* are optional.

| | |
|---|---|
| **_layout.cshtml:** | |
| • TempData variable displayed before the guest page in red and bold | _______/ 1 |
| • Student name & section is in the footer | _______/ 1 |
| **Member controller & views** | |
| • Is accessible from the menu | _______/ 1 |
| • The Create view has a textbox instead of a drop-down for the *provinceCode* | _______/ 1 |
| • The Create view allows the user to specify the *memberId* | _______/ 1 |
| • The Edit & Index views show *memberId*, but it cannot be modified | _______/ 1 |
| • Create, Edit & Delete work, displaying a message if successful | _______/ 3 |
| • Create, Edit & Delete show the innermost exception (if thrown) with edit errors and user input | _______/ 3 |
| **Patient Edits:** these must be in the metadata & partial class: **50% loss if done in controller** | |
| • Annotations and edits are not added to the generated model or controller actions | _______/ 1 |
| o Annotations are applied to model via a metadata class to a like-named partial class | _______/ 1 |
| o The like-named partial class is self-validating | _______/ 1 |
| • String fields are trimmed of leading/trailing blanks before writing to the database | _______/ 2 |
| • *fullName*, *firstName* and *lastName* fields cannot be empty, null or just blanks | _______/ 2 |
| • *fullName* is correctly derived from the member's and spouse's names | _______/ 3 |
| • *provinceCode* is accepted mixed-case, forced to uppercase, and verified using the *province* table | _______/ 3 |
| • *homePhone*, if provided, is the correct pattern and the dashes are inserted if missing | _______/ 2 |
| • *postalCode*, if provided, is a correct Canadian pattern (A3A 3A3), case-insensitive, optional space | _______/ 3 |
| o … it is shifted to uppercase with a single space before being written to the database | _______/ 2 |
| • If *useCanadaPost* is checked, *email* is optional (but allowed), *street* & *city* are required. | _______/ 2 |
| o If not checked, *email* is required, *street* & *city* are optional (but allowed) | _______/ 2 |
| • Code correctly determines if this is creating a new record or editing an existing one: | _______/ 1 |
| o Creating a new record does not attempt to create a duplicate key | _______/ 2 |
| o Updating an existing record does not require any fields to be changed. | _______/ 1 |
| Base mark | _______/ 40 |
| -1@ for fixes Build Errors Run Errors | -_______ |
| Final Mark | _______/ 40 |