In your documentation and your presentation

* Solve a problem with a full-stack web application? Define the problem:
* Demonstrate how your application solves the problem?
* Tech stack in use:
* Represent professional level of quality:

|  |  |
| --- | --- |
| **Create Persistent Records** | |
| Must demonstrate at least one data constraint. | Example required |
| User input is sanitized before being saved to the database. | Example required |
| Data constraint must be protected at every layer of the application. | Example required |
| Records must exist after the user has logged out, cleared their browser cache, and returned to the application | Example required |
| **Read Records From Database** | |
| Sample data or dummy data must be stored in a database. | Example required |
| You must include a .sql script with INSERT statements for your database. | In codebase already |
| README must include instructions on how to run this script. | Procedure required |
| **Update Persistent Records In Database** | |
| Updates must enforce the same data constraint(s) that were mentioned in the create. | Example required |
| Record’s state must remain updated after user has logged out, cleared their browser cache, and returned to the website (this includes dummy data or any starting data your app has). | Example required |
| **Archive Records in Database** | |
| Must demonstrate the safe archive of records. | Example required |
| Child records must also archived. | Example required |
| No view may display any part of the deleted record, or any part of the deleted child records. | If the filter is not available in front end, potential gap |
| If records are archived then user is presented with text stating that the record was archived and not permanently deleted. | Potential gap, the messages must be changed |
| Record must remain archived after user has logged out, cleared their browser cache, and returned to the website | Example required |
| **Project Management Principles and Practices** | |
| Problem Definition must be included in README. | README not created yet |
| Scope (in-scope items, and out-of-scope items) must be included in the README | README not created yet |
| Project Plan must include a breakdown of tasks. | Example required |
| Task breakdown hosted on Trello. | Example required |
| Each card in the completed column must have a Member attached to it so we can see who did what at a glance. | Example required |
| Link to public Trello must be included in the README. | README not created yet |
| Project Plan must include some form of wireframe (MS Paint, Draw.io, Figma, Balsamiq, etc.) | Example required |
| Include the final wireframe(s) and planning documentation in a “Planning” folder in your GitHub repo. | All project files to be loaded to GitHub |
| Testing Plan must include instructions on how to test each feature. | Testing plan procedure is required |
| List of test cases and testing instructions are included in README. | README not created yet |
| **Scope of the project must be well managed.** | |
| Unfinished features must not be apparent anywhere in the application. | Code clean up to be done to remove un used models, endpoints, actions, components, actions, and reducers. |
| Dead code must be deleted, not just commented out. | Commented code to be removed. |
| The problem that you defined in your problem statement must be solved. | Potential gap |
| Any features not related to the problem or that are out of scope should be deprioritized or removed completely. | Potential gap |
| Presentation must include a demonstration of the product and the following talking points: | Prepare use demo cases in front of audience |
| **Final Project Report:** | |
| * Did you complete all in-scope tasks? | Project features tracker |
| * Did you complete any extra tasks? | Project features tracker |
| **Satisfaction Assessment** | |
| * Does your app solve your problem? | Potential gap |
| * Have you or someone else started using the app? What do they think of it so far? | Not applicable |
| **Lessons Learned** | |
| * If you had to build another full-stack CRUD app what lessons would you apply that you have learned from this project? | Group answer is required |
| **Technical Requirements** | |
| Intuitive User Interface | |
| * Consistent navigation across all pages. | To be verified |
| * + Links don’t grow or shrink or run away from the mouse. | To be verified |
| * + Links are in the same place every time. | To be verified |
| * + The link for the page we are currently on is highlighted in navigation in some way. | Potential gap |
| * Buttons do what they say they are going to do. | To be verified |
| * Error messages explain how to fix the error. | Potential gap |
| * Web controls are used appropriately and consistently. | To be verified |
| * On-screen instructions - if necessary - are easy to understand. | To be verified |
| **Mobile-First** | |
| * All content is legible on a small screen. | To be verified |
| * Content is contained on the screen without horizontal scrolling. | Potential gap |
| * Buttons and forms all work on mobile. | To be verified |
| * Buttons are not too close together. | To be verified |
| * Form fields and labels are visible while typing with on-screen keyboard. | To be verified |
| * All key features needed to solve your problem are available in mobile mode. | To be verified |
| **Responsive** | |
| * Uses min-width media queries to manage different browser viewport sizes. | Not sure if applicable |
| * Uses fluid units of measurement consistently. | Not sure if applicable |
| * May use px measurements for min-width. | Not sure if applicable |
| * All HTML pages have the appropriate meta tag for accessing the device’s width. | Not sure if applicable |
| **Accessible** | |
| * Source code and website passes [WCAG validator](https://achecker.ca/checker/index.php). (React projects may require the use of a browser plugin for testing accessibility). | To be verified, Google for react validators |
| **W3C Compliance** | |
| * HTML passes [W3C Validator](https://validator.w3.org/). | To be verified |
| * CSS passes [W3C Validator](https://jigsaw.w3.org/css-validator/). | To be verified |
| **Separation of Concerns** | |
| * Presentation layer contains only presentation code, and some data validation code to prevent users from entering erroneous data. | Example required |
| * Business logic held in appropriate services, single-purpose principle is applied throughout the project. | Example required |
| * Data storage and data access layer does not contain any data transformations and reinforces data validation. | Example required |
| **Browser Console** | |
| * No error messages when using the application. | To be verified |
| * No console log messages while using the application. | Code clean up required, and to be verified |
| * Does not reveal sensitive user information. | To be verified, make sure no password is in the state |
| **Code is commented** | |
| * Citations must include full urls for any code found, borrowed, modified, from the internet, from the class, or from any source other than yourself. If you borrow code from a book, provide the title of the book, the author(s), and the ISBN. | We need to cite the classroom code we used for validation in the backend. General citation should be fine I believe |
| * Citations must be duplicated in README. | README not created yet |
| * Comments describe what methods are meant to do. | Code clean up required, need to verify all docstrings present and inline comments as well |
| **Naming Conventions** | |
| * Proper casing on file names. | Example required |
| * Proper casing on variables/functions/modules. | Example required + Include the code style guidelines |
| * Variables/functions/modules have descriptive names. | Example required |
| **General Good coding practices** | |
| * Look for blocks of code with more than a few lines of code that look similar. Is it possible to refactor to reduce duplication? DRY your code! | Use Re-Sharper before final GitHub commit |
| * Programming patterns are followed consistently within the project. | Example required |
| * Simplify “too smart” and over-engineered code. | Example required |
| * No hardcoded values, use constants values. | To be verified, potential gap |
| **Spelling and Grammar** | |
| * Contents of each web page is spell-checked. | To be verified |
| * Contents of each web page is grammar-checked. | To be verified |
| * Contents of README are spell-checked. | To be verified |
| * Contents of README are grammar-checked. | To be verified |
| **Use up-to-date language features** | |
| Example: const and let instead of var, section instead of div, string interpolation instead of string concatenation, etc. | To be verified, might not be applicable |
| **Design Requirements** | |
| **Contrast** | |
| Font colours must pass the [Web AIM contrast checker](https://webaim.org/resources/contrastchecker/). | To be verified |
| Contrast does not cause eye strain (black on white, red on green, etc.). | To be verified |
| Size contrasts are used correctly if size contrast is used at all.  *(big buttons for important actions, small buttons for unimportant actions)* | To be verified |
| **Alignment** | |
| Content does not overflow from it’s boundaries. | To be verified |
| Content appears organized and structured. | To be verified |
| Text content is not center-aligned.  (exceptions permitted for poetry and wedding invitations) | To be verified |
| **Repetition** | |
| Navigation placement is consistent across the whole application. | To be verified |
| Button and link styles are consistent across the whole application. | To be verified |
| Button and link behaviour is consistent across the whole application. | To be verified |
| Text styles are consistent across the application. | To be verified |
| **Proximity** | |
| * Alike content items are grouped closer to one another. | To be verified |
| * No content is crowded against other content or the edges of the browser. | To be verified |
| * Controls are grouped together logically with their labels. | To be verified |
| * Proper use of whitespace. | To be verified |
| **Deliverables Checklist** | |
| **GitHub link** | |
| * README.md |  |
| * + Name of the project is included. |  |
| * + List of contributors is included. |  |
| * + Problem Statement is included. |  |
| * + Description of how the app solves the stated problem is included. |  |
| * + Instructions for installing the application are complete. | Link to a procedure |
| * + Instructions for using the application are complete. | Link to a procedure |
| * + List of all citations is complete. |  |
| * + List of test cases and testing instructions are included. | Link to a procedure |
| * + Link to the public Trello board is included |  |
| * Main branch contains all final code needed to run the project. |  |
| **Presentation** | |
| * Final Project Report | As per the project spec doc |
| * Satisfaction Assessment | As per the project spec doc |
| * Lessons Learned | As per the project spec doc |
| **Group Member Performance Review** | |
| * Each group member has been evaluated. | Post presentation |
| * All fields filled out. | Post presentation |
| **Personal Reflection** | |
| * Written reflection. | Post presentation |
| * Video reflection. | Post presentation |