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** | | **Status** |
| Must demonstrate at least one data constraint. | Example required | **Done** |
| User input is sanitized before being saved to the database. | Example required | **Done** |
| Data constraint must be protected at every layer of the application. | Example required | **Done** |
| Records must exist after the user has logged out, cleared their browser cache, and returned to the application | Example required | **Done** |
| **Read Records From Database** | |  |
| Sample data or dummy data must be stored in a database. | Example required | **Done** |
| You must include a .sql script with INSERT statements for your database. | In codebase already | **Done** |
| README must include instructions on how to run this script. | Procedure | **Done** |
| **Update Persistent Records In Database** | |  |
| Updates must enforce the same data constraint(s) that were mentioned in the create. | Example required | **Done** |
| 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 | **Done** |
| **Archive Records in Database** | |  |
| Must demonstrate the safe archive of records. | Example required | **Done** |
| Child records must also archived. | Example required | **Done** |
| 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 | **Done** |
| 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 | **Done** |
| Record must remain archived after user has logged out, cleared their browser cache, and returned to the website | Example required | **Done** |
| **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 | **Amr** |
| Project Plan must include a breakdown of tasks. | Example required | **Done** |
| Task breakdown hosted on Trello. | Example required | **Done** |
| 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 | **Amr** |
| Link to public Trello must be included in the README. | README not created yet | **Done** |
| Project Plan must include some form of wireframe (MS Paint, Draw.io, Figma, Balsamiq, etc.) | Cleanup required | **Amr** |
| Include the final wireframe(s) and planning documentation in a “Planning” folder in your GitHub repo. | All project files to be loaded to GitHub | **Amr** |
| Testing Plan must include instructions on how to test each feature. | Testing plan procedure is required | **Atinder**  **Before testing** |
| List of test cases and testing instructions are included in README. | README not created yet | **Done** |
| **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. | **Ayesha**  **After testing** |
| Dead code must be deleted, not just commented out. | Commented code to be removed. | **Ayesha**  **After testing** |
| The problem that you defined in your problem statement must be solved. | Potential gap | **May be** |
| Any features not related to the problem or that are out of scope should be deprioritized or removed completely. | Potential gap | **Done** |
| Presentation must include a demonstration of the product and the following talking points: | Prepare use demo cases in front of audience | **Amr** |
| **Final Project Report:** | |  |
| * Did you complete all in-scope tasks?   We have implemented all features perfectly from backend perspective, however one of the features (grading) implementation in the frontend was a bit tricky, and due to time constraint we had to deprioritises it to fulfill the other project deliverables. | Project features tracker | **Amr** |
| * Did you complete any extra tasks?  1. We have enabled multi-cohort platform where more than one cohort can be full managed for courses, homework, timesheets, grades, etc. 2. We have created course to cohort assignment feature. 3. Yes, we have implemented the infrastructure work to up scale the project features in future, example shoutout form, notification in the DB. | Project features tracker | **Amr** |
| **Satisfaction Assessment** | |  |
| * Does your app solve your problem?   Yes of course | Potential gap | **Amr** |
| * Have you or someone else started using the app? What do they think of it so far?   Not yet, we hope TechCareers embrace the application so we can further improve it and have it fully deployed | Not applicable | **Amr** |
| **Lessons Learned** | |  |
| * If you had to build another full-stack CRUD app what lessons would you apply that you have learned from this project? * Full stack development might be time consuming with so many details, specially if product quality is critical (i.e., debugging, validation layers implementation, etc.). * Scope the project with full understanding of the team available resources in terms in availability and schedule and level of expertise. * Follow the team norms and good team software development is critical to avoid rework. * Better communication is required to avoid over estimated scope, any concerns must be communicated early on and any features that the team is not sure how to implement, must be demoed on spot or deprioritized. | Group answer is required | **Amr** |
| **Technical Requirements** | |  |
| Intuitive User Interface | |  |
| * Consistent navigation across all pages. | To be verified | **Done** |
| * + Links don’t grow or shrink or run away from the mouse. | To be verified | **Done** |
| * + Links are in the same place every time. | To be verified | **Done** |
| * + The link for the page we are currently on is highlighted in navigation in some way. | Potential gap | **Alex** |
| * Buttons do what they say they are going to do. | To be verified | **Done** |
| * Error messages explain how to fix the error. | Potential gap | **Atinder** |
| * Web controls are used appropriately and consistently. | To be verified | **Done** |
| * On-screen instructions - if necessary - are easy to understand. | To be verified | **N/A** |
| **Mobile-First** | |  |
| * All content is legible on a small screen. | To be verified | **TBA (Team)** |
| * Content is contained on the screen without horizontal scrolling. | Potential gap | **TBA (Team)** |
| * Buttons and forms all work on mobile. | To be verified | **TBA (Team)** |
| * Buttons are not too close together. | To be verified | **TBA (Team)** |
| * Form fields and labels are visible while typing with on-screen keyboard. | To be verified | **TBA (Team)** |
| * All key features needed to solve your problem are available in mobile mode. | To be verified | **TBA (Team)** |
| **Responsive** | |  |
| * Uses min-width media queries to manage different browser viewport sizes. | Not sure if applicable | **TBA (Team)** |
| * Uses fluid units of measurement consistently. | Not sure if applicable | **TBA (Team)** |
| * May use px measurements for min-width. | Not sure if applicable | **TBA (Team)** |
| * All HTML pages have the appropriate meta tag for accessing the device’s width. | Not sure if applicable | **TBA (Team)** |
| **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 | **TBA (Team)** |
| **W3C Compliance** | |  |
| * HTML passes [W3C Validator](https://validator.w3.org/). | To be verified | **TBA (Team)** |
| * CSS passes [W3C Validator](https://jigsaw.w3.org/css-validator/). | To be verified | **TBA (Team)** |
| **Separation of Concerns** | |  |
| * Presentation layer contains only presentation code, and some data validation code to prevent users from entering erroneous data. | Example required | **Done** |
| * Business logic held in appropriate services, single-purpose principle is applied throughout the project. | Example required | **Done** |
| * Data storage and data access layer does not contain any data transformations and reinforces data validation. | Example required | **Done** |
| **Browser Console** | |  |
| * No error messages when using the application. | To be verified | **TBA (Team)** |
| * No console log messages while using the application. | Code clean up required, and to be verified | **TBA (Team)** |
| * Does not reveal sensitive user information. | To be verified, make sure no password is in the state | **Done** |
| **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 | **TBA (Team)** |
| * Citations must be duplicated in README. | README not created yet | **Amr** |
| * Comments describe what methods are meant to do. | Code clean up required, need to verify all docstrings present and inline comments as well | **TBA (Team)** |
| **Naming Conventions** | |  |
| * Proper casing on file names. | Example required | **Done** |
| * Proper casing on variables/functions/modules. | Example required + Include the code style guidelines | **Done** |
| * Variables/functions/modules have descriptive names. | Example required | **Done** |
| **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 | **Done** |
| * Programming patterns are followed consistently within the project. | Example required | **Done** |
| * Simplify “too smart” and over-engineered code. | Example required | **Done** |
| * No hardcoded values, use constants values. | To be verified, potential gap | **Done** |
| **Spelling and Grammar** | |  |
| * Contents of each web page is spell-checked. | To be verified | **TBA (Team)** |
| * Contents of each web page is grammar-checked. | To be verified | **TBA (Team)** |
| * Contents of README are spell-checked. | To be verified | **TBA (Team)** |
| * Contents of README are grammar-checked. | To be verified | **TBA (Team)** |
| **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 | **Done** |
| **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 | **Ayesha / Amr** |
| * + Instructions for using the application are complete. | Link to a procedure | **Ayesha/ Amr** |
| * + List of all citations is complete. |  |  |
| * + List of test cases and testing instructions are included. | Link to a procedure | **Atinder / Amr** |
| * + Link to the public Trello board is included |  |  |
| * Main branch contains all final code needed to run the project. |  | **Amr** |
| **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 |  |