

CS Capstone Design

Final Project Demo Grading Sheet (50 pts)

	50
A: +90%	45
B: +80%	40
C: +70%	35
D: +60%	30
F: <50%	25

TEAM: What's Up Doc

Overview: The main purpose of the “Technical Demos” is to very clearly explore the extent to which the team has implemented the key functional and performance requirements for their project. Grading is based on two factors:

- **Completeness:** Have all of the key requirements been implemented. To what extent does the product have all of the functionalities and performance that was promised.
- **Quality:** Just having basic functionality is the bare minimum. What is the quality of the implementation? Is the resulting product aesthetically pleasing, easy to use, and a pleasure to work with; to what extent is it “ready for market”?

This template is fleshed out by the team, **reviewed and approved by CS mentor beforehand**, and then brought to demo in hardcopy for the mentor to use as a grading sheet.

Requirements Review

Based on our requirements acquisition work and evolution during implementation, the following are the key technical requirements driving of our product:

R1: Intuitive Administrator View: The administrator home page should be intuitive and drive the users eyes to the key features of the page. There should be no question as to what this admin page has to offer and no features should be a “surprise” or be hard to find for the admin user. Every button should clearly be named to best describe what each button is doing. All searching and sorting for the admin tables should be intuitive to perform.

Specific implemented functionalities that satisfy this requirement:

- The most prominent features the admin should see when they first access the admin.html page is the task bar at the top of the page. The task bar is a div named “adminControlsTaskBar” and is made up of three buttons, “Add/Remove Students”, “Phase Review”, “Student Progress Viewer”. The first two buttons lead to the two tables at the admin’s disposal. The third button leads to the search bar that allows the admin to view the students home page and access their uploaded documents. These buttons do a good job of separating the three key features on the admin page from each other. Without this taskbar, the tables and search bar would be stacked on top of each other on the same page and the web page would not be intuitive.

R2: Intuitive Student View: The student home page should be intuitive and drive the users eyes to the key features of the page. There should be no question as to what this student page has to offer and no features should be a “surprise” or be hard to find for the student user.

Specific implemented functionalities that satisfy this requirement:

- Again, the most prominent feature the student should see is the large color coded horizontal taskbar at the top of the page. The horizontal taskbar for students will contain a varying amount of “phases” that may change depending on the graduate program itself. By default, the first phase and its accompanying vertical milestone taskbar will be open.

Clicking on any of the “phases” will cause a second, vertical, taskbar to open up. This vertical taskbar will contain a varying amount of milestones. Each milestone will have a brief description of it that can be read by hovering over the small “?” underneath each milestone tab on the vertical taskbar. By clicking an individual milestone, students will be able to upload/download/delete their deliverables. Completing an individual milestone changes the milestone tab on the vertical taskbar from blue to gold. Completing an entire phase changes the phase tab on the horizontal taskbar from blue to gold as well. The color coding makes it very clear to the student that a change has been made, and that progress is being updated.

R3: Allow for Upload/Download: From the student home page, the students should be able to intuitively select exactly which milestone/phase they want to update. The students should be able to upload a deliverable for each milestone in each phase, without fail. Following a successful upload, the student should then be able to click the “Download” button and download their newly uploaded file back from where it is saved in the database.

Specific implemented functionalities that satisfy this requirement:

- First, the student selects exactly which milestone they want to update using the two taskbars available.
 - They do so by clicking an overlying “Phase” from the horizontal taskbar at the top of the page.
 - Then, the student clicks a “Milestone” tab from the vertical taskbar on the left side of the page.
- Once a milestone has been chosen/selected, the student clicks the “Upload” button and will be prompted with a file manager popup.
 - Here the student will choose a PDF file (must explicitly be a PDF file, both the client and developers have agreed on this requirement) from their local device and click Yes. Assuming a PDF has been uploaded, the file will appear within the milestone window in a PDF viewer.
 - The built-in PDF viewer allows for viewing, pagination, printing, etc.
 - Successfully uploading a file will cause the corresponding milestone tab (within the left side vertical taskbar) to change from blue to gold, in order to specify that the milestone has been completed.
 - Note: If the user uploads a file that is NOT a PDF, then the file will be uploaded to the database and the color coding will change, however, the PDF viewer will not accurately display the non PDF file that has been uploaded.
 - Note: If a student were to click the “Upload” button on a milestone that is already marked “Completed” and has a file currently uploaded, the old file will be overwritten by the new file that is to be uploaded in its place.

- Once a milestone deliverable has been uploaded, the student may select the milestone again and click on the “Download” button. Clicking this button will download the file straight from the database as it was uploaded.
- If a given milestone needs to be removed from the database/website, the student may select the milestone from the taskbar, and they can click the “Delete” button. This will remove the selected milestone’s file (if one exists) from the database, reload the page, and remove the PDF viewer for that given milestone, showing that the file has been successfully removed.
- From the administrator home page, the admin may click the third button, titled “Student Progress Viewer” to access a search bar. The intention of this search bar is to allow the admins to search for students by their unique userID and have their corresponding student home page show up in embedded form on the admin home page. This will give admins access to any student home page simply by searching, and will allow admins to upload/download/delete deliverables straight from the students home page after it has been submitted.
 - Note: Due to the fact that Team What’s Up Doc did not get token authentication/authorization fully integrated, this feature is not fully functional. As of right now, the search bar accepts what the user submits and saves the userID into a variable. Upon form submission, the page simply reloads right now. For the acceptance demo, the same static student home page (eml292) is embedded on the “Student Progress Viewer”.
 - Users are able to upload/download through the embedded home page, as expected.

R4: Administrative Accessibility: From the admin.html home page, Administrators should be able to easily access all CRUD features for the database. Admins should also be able to quickly and effectively analyze how students are doing within the graduate program.

Specific implemented functionalities that satisfy this requirement:

- By clicking the “Add/Remove Users” button on the horizontal taskbar at the top of the admin.html page, the admin will open up the “adminStudentControlsTable” (AKA the “Add/Remove Users DataTable”)
 - Within this DataTable, admins will have access to a plethora of CRUD and non CRUD buttons/features that will aid them in managing and tracking their graduate program.
 - Admins have great accessibility to each DataTable by having access to features such as pagination, sort by column, and being able to search the table by keyword.
 - “New User” button: Clicking this button will prompt the admin to enter in some user information (userID, last name, first name, advisor last name, term activation, enrollment status, and admin status) and then the admin will click “Submit”. Once the form is submitted, a brand new user entry should be visible within the “Add/Remove Users DataTable”, indicating that the user has also been entered into the database as well.

- “Edit User” button: Administrators will first have to select a specific user within the DataTable before this button is clickable. Once a user has been selected, their cell will be highlighted blue and the “Edit User” button will become clickable. Once this has been clicked, another modal will popup prompting the user to reenter all the user information except the userID. (Note: if the admin clicks the “Edit User” button, they must explicitly reenter all fields for the selected user, even if all the fields did not necessarily have to be updated. This inconvenience would need to be refactored in future iterations of the product.) Once the user information has been entered, the admin will click “Submit” and the form will update the selected user’s information within the table/database.
- “Delete User” button: Admins will, again, have to first select a user within the “Add/Remove Users DataTable” before they are allowed to click the “Delete User” button. Upon clicking the “Delete User” button, nothing, graphically, will happen at first. However, on the backend, the deletion has already been made. Before the admin’s deletions will be reflected on the frontend DataTable, the admin must click the “Refresh Table” button. Once the “Refresh Table” has been clicked, the table should reload and the deletion made should be reflected within the DataTable.
- This DataTable also features buttons that: copy the table to clipboard, export the table to csv, export the table to excel, and export the table to PDF.
- By clicking the “Phase Review” button on the horizontal taskbar at the top of the admin.html page, the admin will open up the “Phase Review DataTable”. This table exclusively focuses on the analytical requirements of the project.
 - At first glance, the admin should be able to recognize the first four columns as they are directly related to the first DataTable and will dynamically change as changes are made to the “Add/Remove Users” DataTable. The final four columns directly relate to the graduate program and are subject to change as the graduate program is updated and as students upload their deliverables.
 - This DataTable also features buttons that: refresh the page, copy the table to clipboard, export the table to csv, export the table to excel, and export the table to PDF.
 - Admins have great accessibility to each DataTable by having access to features such as pagination, sort by column, and being able to search the table by keyword.

R5: NAU Login: All users should be required to flow through the NAU/Gmail login workflow. This includes CAS, DuoMobile, and giving consent to the Google APIs being used within the application.

Specific implemented functionalities that satisfy this requirement:

- Note: Looking at the index.html page, all users can see a prominent login window in the middle of the landing page. All users should be able to recognize the “Sign in with Google” button and they should be comfortable clicking it. This will take the user through the normal NAU gmail sign in workflow. This workflow will be described in the sections below.
- As of the Acceptance Demo iteration, the NAU Login feature is about 75% complete. The current login feature accurately displays how it is linked to both Google networks

and NAUs CAS and Duo Mobile. The only shortcoming is that, upon successful login, the user is statically redirected to the same destination URI instead of a check being done (with proper authorization/authentication tokens) to properly identify the user as a student or an administrator and then dynamically redirecting them to their corresponding student/admin home page.

- For the time being, we have circumvented this issue by integrating what we have of the login feature and including a “For Developer Use Only” box at the bottom of the index.html page that allows our users to redirect to our two other completed pages.

Demonstration Sequences:

This section outlines the demonstration sequences prepared to prove the above functionalities. Each sequence is a coherent walk-through of some piece/area of the product, designed to highlight implementation of specific requirements/functionalities outlined in the last section.

Demonstration Sequence 1: User Login/Home Page Access

Requirements demonstrated: R1 , R2, R4, R5

Flight Plan for this demo sequence:

1. Users go to doctracker.org.
2. Users click the recognizable “Sign-In With Google” button.
3. Google Sign-In popup window appears.
4. If the user is already signed into their NAU associated gmail account in their browser, they can select their Gmail account and likely skip steps 6-12.
5. Users are prompted to confirm that they want to sign into the “NAU C&I Doctoral Tracker” using their entered NAU gmail account. The popup will ask the user to click “Confirm” and consent to any Google APIs that the application may be utilizing.
 - a. At the time of the Acceptance Demo, the only Google API being used is the standard gmail user API which gives the application access to the user’s name, email address, and profile picture.
6. Users enter in their NAU gmail and click “Next”.
7. NAU Central Authentication Service (CAS) popup window appears.
8. Users enter in their NAU User ID and Password and click “Sign-In”.
9. Users approve the Duo Push Two-Step verification process (if they have two-step authentication implemented).
10. User clicks the “Yes, trust browser” button (if they have two-step authentication implemented).
11. Users reach the Choose an Account pop page.
12. Users click their appropriate NAU gmail.
13. As of the Acceptance Demo, all users are statically sent to the same static redirect URI.
 - a. At this point, this feature SHOULD:

- i. Perform a check on the user that has signed in and save their userID, and both access tokens given to us by Google Identity Services upon successful login.
 - ii. Send the userID and both tokens to the backend for authentication. The backend should identify whether the user is a student or admin and properly redirect them to their corresponding home page.
14. Alternatively: users can scroll down to the bottom of the index.html page and access the “Developer Only” buttons which allow the user to temporarily redirect to either the student home page or the admin home page until the login page is fully functional.

Evaluation (filled in real-time by mentor):

- ✓ Convincingly demo'd each of targeted requirements?

- ✓ Quality, aesthetics and other evaluative comments:

Demonstration Sequence 2: Student Progression Tracking

Requirements demonstrated: R2, R3

Flight Plan for this demo sequence: **NOTE:** This demo sequence will assume you have already followed the “User Login/Home Page Access” demo sequence above.

1. Looking at the home.html page, students now have the option to click either the “phase progress bar” going horizontally across the page or the vertical “milestone taskbar”.
2. If a Phase is clicked on the “phase progress bar” it will be outlined in silver (if the phase is blue and not completed) or black (if the phase is gold and completed), the vertical “milestone taskbar” will update with the appropriate milestones for that phase.
3. If an individual milestone is clicked on the vertical “milestone taskbar” it will, similarly, be outlined in silver (if the milestone is blue and not completed) or black (if the milestone is gold and completed), the associated upload/download/delete window displays in the middle of the page.
4. If the milestone has a deliverable already uploaded, the student can see the PDF file in the embedded PDF viewer and the user may click the “Download” button to download the previously uploaded deliverable straight from the database.
5. Students can upload a completed version of a document to the upload/download window through the intuitive file manager.
 - a. Note: Both the client and the developers have agreed that students are to be explicitly told that only PDF files are to be uploaded in order to guarantee the PDF viewers functionality.

- b. If a file has already been uploaded to the selected milestone, the old PDF file will be overwritten with the new PDF file.
6. Students can view the uploaded PDF in a PDF viewer within the upload/download window.
 - a. This includes all the standard features of the PDF viewer, including another download button, a print button, pagination, page scrolling, page magnification, fit to page, page rotation, and the PDF menu button.
7. Students can click the NAU logo in the top right corner to be redirected to the nau website.
8. Students can hover any of the “?” icons under any milestone in order to read a short description regarding each particular milestone.
9. Students can sign out of their NAU gmail using the “Sign-Out” button which will redirect them to the landing page/login page.

Evaluation (filled in real-time by mentor):

- ✓ Convincingly demo'd each of targeted requirements?

- ✓ Quality, aesthetics and other evaluative comments:

Demonstration Sequence 3: Administrative Program Management and Analysis

Requirements demonstrated: R1, R3, R4

Flight Plan for this demo sequence:

NOTE: This demo sequence will assume you have already followed the “User Login/Home Page Access” demo sequence above.

NOTE: For readability purposes, I have bolded and underlined the three distinct button clicks that the administrators will be using to navigate the admin.html page.

1. Looking at the admin.html page, Administrators should see a blue horizontal taskbar at the top of the page with three blue buttons that display three different windows.
 - a. The “Add/Remove Users” button brings up the user populated table.
 - b. The “Phase Review” button brings up the Phase Review table.
 - c. The “Student Progress Viewer” brings up the search bar and the embedded view of the student home page.
2. **Administrators click the “Add/Remove Users” button.**
 - a. Administrators are able to view a searchable and sortable DataTable populated with sensitive user information including: userID, last name, first name, advisor last name, term activation, enrollment status, and admin status. Unique DataTable features include:
 - i. Page limit: Choose how many items you want each page of the table to hold (10, 25, 50, 100)
 - ii. Search bar: Any value in the table is searchable by keyword or phrase.

- iii. Sorting: any column in the table is individually sortable numerically or alphabetically
 - 1. “individually”: meaning if you select another column to sort, it will overwrite the last sort performed.
- iv. Pagination: If the table is large enough to contain multiple pages, the admin can move between the pages using “Previous”, “Next”, or the page numbers themselves.
- b. Administrators can add a user to the table/database.
 - i. Must explicitly enter in the user’s userID, last name, first name, advisor last name, term activation, and admin status.
 - 1. **Note:** if the “New User” button is pressed and the “Submit” button is pressed immediately afterwards (with no changes made to the form), then the default values of the form will be added to the DataTable as a user. If the “Default” user information has already been stored in the database as a single user, then nothing will happen upon submission of the default form (i.e. no duplicate users with default user information will be added to the database).
 - 2. If the administrator clicked the “New User” button by mistake, they may click the “X” button at the bottom of the modal to exit out of the “New User” modal without submitting the form.
 - ii. Once completed, assuming the new user has a *unique* userID, the table should reload and the administrator should see the new user within the DataTable.
- c. Administrators can select a single cell within the DataTable and the “Edit User” button should now become clickable.
- d. Click the “Edit User” button and a similar looking modal should popup with a form prompting the admin to reenter the selected user’s last name, first name, advisor last name, term activation, enrollment status, and admin status.
 - i. **Note:** In the Acceptance Demo iteration, the user must explicitly re enter ALL of the selected user’s information, once again, even if they only meant to edit a single field.
 - ii. If the administrator clicked the “Edit User” button by mistake, they may click the “X” button at the bottom of the modal to exit out of the “Edit User” modal without submitting the form.
- e. Administrators can select a single cell within the DataTable and the “Delete User” button should now become clickable.
- f. Administrators can click the “Delete User” button to remove a single user from the DataTable and the database.
 - i. **Note:** The user deletion will happen in real time on the back end database, but it will not automatically be reflected on the frontend DataTable. Therefore, the user must press the “Refresh Table” button after every deletion, in order to see the change made in the DataTable.
- g. Administrators can click the “Refresh Table” button to reload the table and view any possible changes that may have been made to the table that might have required a page reload.

- h. Administrators can click the “Copy” button which will copy the current DataTable iteration to the clipboard as CSV.
 - i. **Note:** Clicking this button will copy the immediately viewable version of this DataTable, meaning if the admin was searching by keyword and/or sorting a specific column, then the DataTable outcome from that specific search and sort would be what is copied to the clipboard.
 - i. Administrators can click the “CSV” button which will export the current DataTable iteration as a CSV file.
 - i. **Note:** Clicking this button will export the immediately viewable version of this DataTable, meaning if the admin was searching by keyword and/or sorting a specific column, then the DataTable outcome from that specific search and sort would be what is exported as a CSV file.
 - j. Administrators can click the “Excel” button which will export the current DataTable iteration as an Excel file.
 - i. **Note:** Clicking this button will export the immediately viewable version of this DataTable, meaning if the admin was searching by keyword and/or sorting a specific column, then the DataTable outcome from that specific search and sort would be what is exported as an Excel file.
 - k. Administrators can click the “PDF” button which will export the current DataTable iteration as a PDF file.
 - i. **Note:** Clicking this button will export the immediately viewable version of this DataTable, meaning if the admin was searching by keyword and/or sorting a specific column, then the DataTable outcome from that specific search and sort would be what is exported as a PDF file.
3. **Administrators can click the “Phase Review” button on the horizontal taskbar at the top of the admin.html page.** This table is made up of relevant user information identical to the information in the “Add/Remove Users” DataTable, and four new columns that display a milestone count for each phase, for every user in the graduate program. All of these rows dynamically change as the information in the database is altered.
- a. Administrators are able to view a searchable and sortable DataTable populated with sensitive user information including: userID, last name, first name, advisor last name, term activation, enrollment status, and admin status. Unique DataTable features include:
 - i. Page limit: Choose how many items you want each page of the table to hold (10, 25, 50, 100)
 - ii. Search bar: Any value in the table is searchable by keyword, number or phrase.
 - iii. Sorting: any column in the table is individually sortable numerically or alphabetically
 - 1. “individually”: meaning if you select another column to sort, it will overwrite the last sort performed.
 - iv. Pagination: If the table is large enough to contain multiple pages, the admin can move between the pages using “Previous”, “Next”, or the page numbers themselves.

- b. Administrators can click the “Refresh Table” button to reload the page and view any possible changes that may have been made to the table.
 - i. **Note:** As of the Acceptance Demo iteration, the “Refresh Table” button attached to the “Phase Review” DataTable simply refreshes the entire webpage and brings the admin back to the “Add/Remove Users” DataTable. Therefore, the user would have to click the “Phase Review” button in the taskbar in order to return back to the table they were viewing.
 - c. Administrators can click the “Copy” button which will copy the current DataTable iteration to the clipboard as CSV.
 - i. **Note:** Clicking this button will copy the immediately viewable version of this DataTable, meaning if the admin was searching by keyword and/or sorting a specific column, then the DataTable outcome from that specific search and sort would be what is copied to the clipboard.
 - d. Administrators can click the “CSV” button which will export the current DataTable iteration as a CSV file.
 - i. **Note:** Clicking this button will export the immediately viewable version of this DataTable, meaning if the admin was searching by keyword and/or sorting a specific column, then the DataTable outcome from that specific search and sort would be what is exported as a CSV file.
 - e. Administrators can click the “Excel” button which will export the current DataTable iteration as an Excel file.
 - i. **Note:** Clicking this button will export the immediately viewable version of this DataTable, meaning if the admin was searching by keyword and/or sorting a specific column, then the DataTable outcome from that specific search and sort would be what is exported as an Excel file.
 - f. Administrators can click the “PDF” button which will export the current DataTable iteration as a PDF file.
 - i. **Note:** Clicking this button will export the immediately viewable version of this DataTable, meaning if the admin was searching by keyword and/or sorting a specific column, then the DataTable outcome from that specific search and sort would be what is exported as a PDF file.
4. **Administrators can click the “Student Progress Viewer” button on the horizontal taskbar at the top of the admin.html page.**
- a. Administrators should be able to see a search bar and an embedded view of the student home page. The intention of this feature is for the administrator to search for a user (specifically by userID) and have an embedded view of their unique student home page appear in the current window.
 - i. For the current Acceptance Demo iteration, the search feature simply saves the submitted form as a variable and prepares it for future use. The embedded student home page is a static embedded homepage that is meant to display what the embed would look like.
5. Administrators can sign out of their NAU gmail using the “Sign-Out” button which will redirect them to the landing page/login page.
- a. **Note:** The “Sign Out” button will not sign the user out of their Gmail account. The user will have to explicitly sign out of their google account via their browser.

Evaluation (filled in real-time by mentor):

- ✓ Convincingly demo'd each of targeted requirements?

- ✓ Quality, aesthetics and other evaluative comments:

Other challenges recognized by not addressed by demo:

As previously mentioned, the only shortcoming of the project was that we did not successfully integrate Google's authentication token and refresh token into our web application. Integrating this feature would have required extensive research, implementation, and cohesion across the frontend, backend, and server side of the project, which proved to be a demanding feature to learn and incorporate, amidst satisfying all the other requirements. As it stands in the index.html file, there is code in place to allow Team What's Up Doc to contact Google networks and save an encoded Google Identity Services (GSI) JSON Web Token (JWT) payload as a variable. This encoded JWT would need to be decoded (either on the frontend or the backend) and then processed on the backend. The userID, admin status, authentication token, and refresh token need to be saved and/or verified by the database. Once the tokens and admin status have been verified and match a user within the database, the valid tokens should be sent back as confirmation and the user should be redirected to their respective home page, be that their unique student home page or the admin home page.

Until the login page is fully integrated and the token system is fully functional, users will be required to navigate between the student home.html page and the admin.html page using the redirect buttons within the green "Developer Only" box from the index.html page.

The "Student Progress Viewer" is not fully functioning, also due to the token system not being fully implemented.