

Special Server Setup Procedure: As in the previous submission, if your server has any advanced features that require additional setup besides npm install and node src/server.js, include a section in the report that describes how to set up these features. Include details on how to tell if the feature is working properly.

Clicking the “Save Progress” button on the homepage prompts a dialog box - if this does not pop up then you might need to enable pop ups in your browser.

Cytoscape.js is third-party code used to help render the home screen, although everything necessary to run it should already be set up.

You can get the feedback (which is sent by users through the about page) as an admin in Postman with GET localhost:3000/feedback/00000000000000000000000000000001 and Authorization = “Bearer eyJpZCI6MX0=” (Since user 00000000000000000000000000000001 is an admin).

The reset database button is on the right side of the top navbar.

Individual Contributions: Include a section that describes each startup founder’s contribution. Each startup founder should be responsible for at least one product feature and its HTTP route(s). Name the feature and the HTTP routes that the founder updated to use the database. Each founder must be responsible for converting at least one HTTP route.

Mike Keegan- Wrote the GET ‘/courses/:courseid’ route for the database to get course data, which is important for the course details page and the main page. I also wrote the GET ‘user/:userid’ route which gets the user data from the database for several pages. Both of these routes required objects to be resolved, either courses or majors, and I wrote the methods to resolve the objects, as well. Additionally, I completed the PUT ‘/user/:userid/majortoshow/:majorid’ and PUT ‘/user/:userid/minortoshow/:minorid’ routes to show elements on the graph and DELETE ‘/user/:userid/majortoshow/:majorid’ and DELETE ‘/user/:userid/minortoshow/:minorid’ to take elements off of the main app. Finally, I modified the POST ‘/savedgraph’ route to be used with MongoDB.

Anthony Depace- I completed the DELETE ‘/user/:userid/courses/:courseid’ route. This route is used to remove a class from a user’s list of classesTaken in the database. Taken classes are red on the main application. By clicking on one you will be brought to the course information page where you can press the button to remove the class (by using this route). I also added a graphical feature to the main app where when a user hovers their mouse on a node, the direct children of that node get styled with a green border.

Geon Yoon- I completed the GET ‘/user/:userid/page’ route and DELETE ‘user/:userid/page/pageid’ route. These show saved-pages in the page and make user delete the specific graph. By passing manually assigned page index rather the element index that are naturally generated, users can delete the graph that they are facing through search. Also, I updated reset database functions

Andrew Knowles-

Ben Reul- rewrote POST '/feedback', GET '/feedback', and PUT '/user/:userid/courses/:courseid'. The first two routes are used to post and retrieve feedback from users. The third adds a class to the students taken list. This changes the node in the graph from gray or light blue to red in order to signify this student has already taken the class.

Joseph Geneva-

Lingering Bugs / Issues / Dropped Features: If your application has any lingering bugs, issues, or missing/dropped features, include a section listing these. Similarly, if any particular features or components are not using the database, please mention those features here

The interactive graph on the main page gets some bugs if the same class is in multiple majors. This might be able to be fixed, but not without changing the way the data is loaded. We will look into it, but for now we simply keep classes for majors limited to their respective department so that there is no overlap. In order to show a course as “taken”, you must also remove it from “take next semester”. This is to benefit those who might need to retake a class, but the way we implemented it might be confusing for some. “Similar classes” in the course details page has not been implemented yet. The change password text boxes and button have not yet been implemented yet. The main app needs a second to refresh, sometimes flashing incorrect versions of the graph as it loads. This is a result of the asynchronous nature of the database and the fact that cytoscape is not really intended to be dynamically loaded. However, it still works as intended. Deleting saved graphs once one of them is added causes some issues as a result of the saved ObjectIDs for the graphs. We need to convert an int (the size of the array) to an ObjectID, but cannot figure out how to do that.