

Interactive Course Tracking

Team Members:

Andrew Knowles, Mike Keegan, Joe Geneva, Ben Reul, Geon Yoon, Anthony Depace

Known Bugs:

- The settings of the main application do not change the displayed graph.
- Filling out the password change text areas and clicking submit does not change any data

Removed Features:

- Removed the “filter gen-ed dependencies” & “show non-major dependencies” from the settings panel on the main application page because we do not have that data represented in our current mock database.
- Sort function from SavePages sidebar.
- Options button from SavePages such as enlarging, editing, printing, saving as PDF, and send it to the email.
- Saving the map as a pdf
- Removed account settings page

Individual Contributions:

Andrew Knowles: Implementation of routing so that URL's work properly when not leaving page. Created links to course description page for each course so that the pages can actually be reached. Implemented the course description page and hooked up information from the mock database. Cleaned code after all group members finished implementation,

Ben Reul: implementation of profile page and all elements associated with it. This required a rework of the CSS, HTML, and adding of a few JS classes. Debugged profile loading. Moved reset button to navbar, css for sidebar

Tony DePace: text graph representation (text_graph.js), adding functionality to the two + buttons on sidebar.js to display which majors/minors should be displayed. Download pdf functionality on sidebar.js. Mouse hover events on the graph in hometwo.js. Save a graph functionality in sidebar.js.

Geon Yoon: SavePage, savePageItem, database data, and server function to communicate with database between(database and savePage.js & savePageItems). Save Page is dynamic but doesn't afford saving of maps. Only has placeholders. Added Database information for supporting saved maps.

Mike Keegan: Made the about page functional including the feedback form, main page graph functionality and design, database data organization, and server functions to communicate with database. Feedback is loaded to the database for access by us. Main page displays courses in a tree using an external library Cytoscape.

Joe Geneva: Link to and implementation of Course History page, placeholder data, generating rows. Table shows classes taken, their number, and a small description. Table items also link to the course to the page themselves.

React components:

Navbar, Homepage Sidebar, Home, Saved Pages, Course Details, Course History, About, Feedback Form, Profile, text_graph

The About page is responsible for showing the user information about the app. At the bottom is the feedback form React component, which allows the user to enter information and submit it to be written to the database with their user number, where we can read it, but it is not accessible by users. The GraphHome component holds a cytoscape model of the course network with arrows denoting prerequisite requirements and colors to show which classes have been created. So far the main shortcoming of this is incorporating the graph with the stateful react components. The graph so far can not rerender appropriately. This is moreso a problem with cytoscape and the way we are building the graph than it is with the use of react.

Sidebar.js holds options for updating and saving the graph on the hometwo.js page and the text_graph.js component. It contains a selection form to show majors/minors, a button to download the graph as a pdf(currently serving dummy data), and a button to save the graph to the Saved Maps page.

Text_graph.js was the initial textual representation of the graph that we will be displaying. It was created primarily to test the retrieving of data from the mock database and printing the nodes/edges that could be displayed by our main app. It has since been replaced with hometwo.js which constructs a graph using a javascript graphing library.

Home.js was the initial react component which holds the settings sidebar (sidebar.js) and the textual representation of the graph (text_graph.js) of our main application. It has since been replaced with hometwo.js which is still the parent of sidebar.js, but does the graphing within the component.

Coursedetails.js is the component that holds a courses information. Data from the server about a course is pulled and placed here. Similar courses are displayed below, but currently they are all CS 326. In the future, there may be some algorithm to dynamically create suggested course components, but right now they are just static links.

CourseHistory.js is the component responsible for displaying the user's past courses. This is currently done in a table. The first entry in each row is a link to the specific course. They link to the classes the user has taken, one row per class. The rows are generated, so they are unique per user and update as courses are taken.

Save Page is responsible for showing the user the graphs that they saved through Main App.

Save Page is consist of savePage.js and savePageItem.js. savePage.js is a threat that has a sidebar and multiple saved graphs items. savePageItem.js renders the user's list of saved graphs within a user's savePage.js

Profile page currently shows most of the students information, such as their name, student id, majors, minors, and email. Eventually this page will be host to some changeable settings, such as password, profile photo, and others.