uAssign - Progress Report 1

Tennyson Demchuk - td16qg@brocku.ca - 6190532
Mutaz Fattal - mf17lg@brocku.ca - 6362156
Joel Gritter - jg17uy@brocku.ca - 6331763 (leader)
Kindeep Singh Kargil - kk17xg@brocku.ca - 6329817
Aditya Rajyaguru - ar18xp@brocku.ca - 6582282
Daniel Sokic - ds16sz@brocku.ca - 6164545

COSC 4P02
March 3, 2021

Stated goals (as stated in project proposal):

- Basic project setup, hosting setup, deployment automation and integrations
- Basic web development complete (i.e. website skeleton done)
- Template dockerfiles for container templates complete

Completed work to date:

- Setup React Project
    - Set up the work environment and installed needed dependencies to get the project up and running.
- Added main theme for the LMS
    - Follows the Brock colour scheme at the moment, red navbar and white and gray tones on other areas.
- Created website skeleton
    - Setup all the general files involved with each page, with basic links that allows you to navigate amongst all the pages.
- Created login page
    - The front-end half of the login page was created in React, containing a login box and a general theme.
- Created home page
    - A general home page layout was created in order to show what it will look like once courses can be added to student accounts.
- Site hosting & auto deployment
    - Implemented this using Firebase hosting and Github Actions auto-deployment. On every pull request to our dev branch, Github Actions will automatically re-deploy the site.
- AWS Lambda
    - Completed setup for API with AWS Lambdas, serverless framework and MongoDB.
    - Set up automated deployments for API with Github actions.
    - Added sample endpoint for CRUD operations
- Authentication
    - Completed AWS Cognito setup for authentication
    - Implemented framework for role based access in AWS Lambda
    - Setup frontend for user login, sign up, and account recovery (forgot password)
- Created Admin Page
    - Added API endpoints to add and delete admins and professors
    - Added an admin page to add and delete admins
- Docker containers
    - Completed the configuration of Dockerfiles for Docker containers to run student submissions
    - The current supported programming languages are C++, Java, and Python

In Progress on March 3, 2021

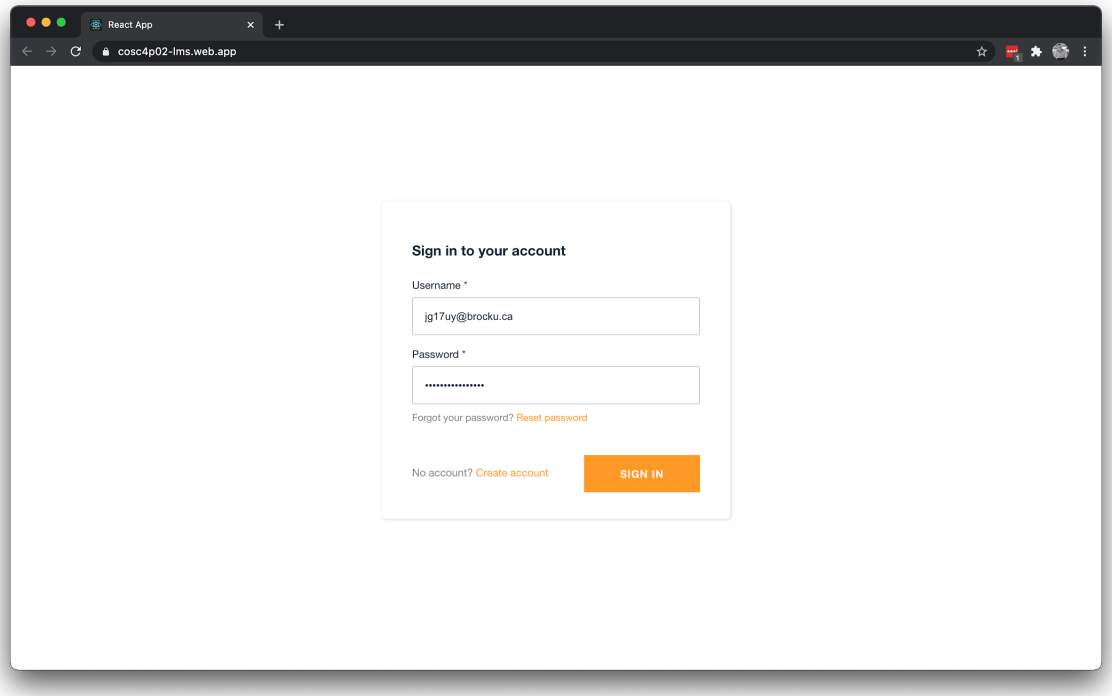
- GridFS FileStorage setup with lambdas
    - Integrated GridFS with current backend to support scalable read and write operations on files with the database
    - Completed POST and DELETE requests for files with error handling
    - Complications with GET request, may lead to using additional services to accommodate
- Editing profile, involves frontend and backend
    - Need to add the fields to show on the website for a particular profile, i.e. their name. Also needs to be able to work with the API so that these fields can be changed and saved per profile.
- Create new courses: API
    - Need an API endpoint to add a new course to the DB.
- Create new courses on the front end
    - Need to create a page that will allow for creating new courses, and assign various data fields to a course.
- Adding and deleting professors from admin page
    - Need to use existing API endpoints to enable creating and removing professors from the system.

Peer Reviews & Testing

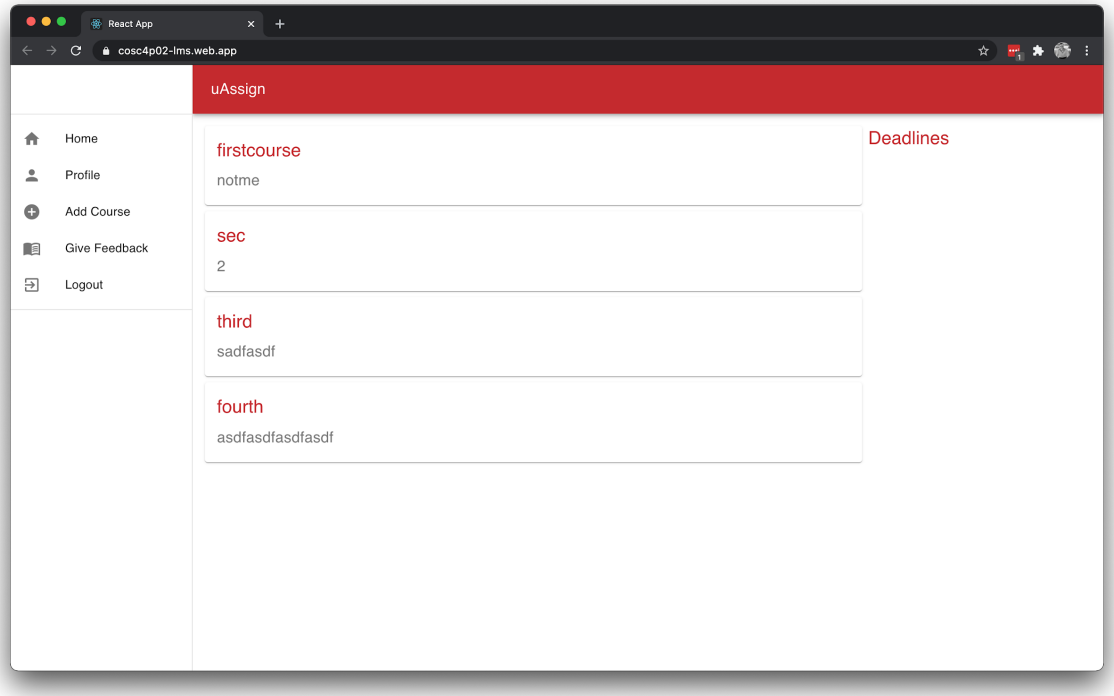
All code is required to be peer reviewed by at least 1 person before merging into the dev branch and tested locally as needed, the build is also automatically tested as part of the deployment, using Github Actions. If there are any errors during deployment, the error is displayed on the repository and emails are sent to those watching the repository.

Demo - https://cosc4p02-lms.web.app/

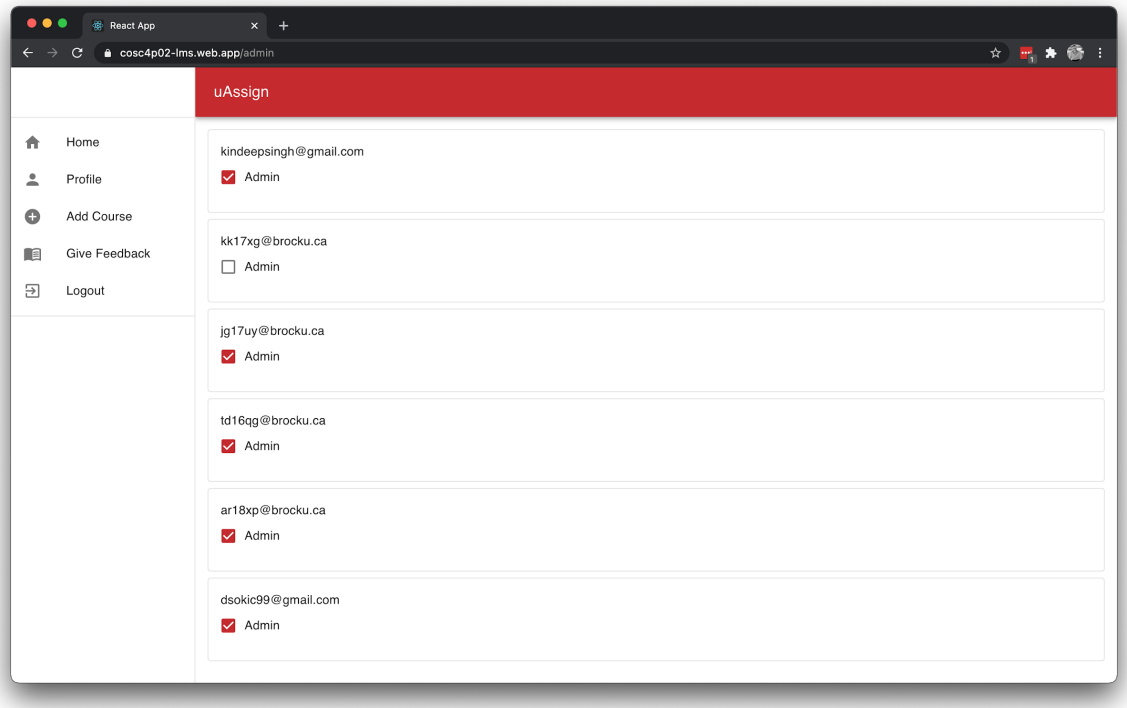
Logging in:



Home page:



Admin page:



Meetings/Teamwork

There are no issues to report. To date, all members of the team have been present at all team meetings, and have consistently been reliable and responsible. All members have completed the work to which they have committed to in a timely manner, and notified the team when issues did arise. When issues did arise, the team collaborated together effectively to resolve the issue.

Meetings have been conducted according to the schedule, as laid out in the project proposal. (Weekly meetings on Wednesday afternoons) Task management has been facilitated through Jira to good affect. Change management with the use of the git-flow strategy as laid out in the project proposal has also been going well through the use of Github, including the use of pull request reviews.