KGSP web platform

Objective:

* The goal of the platform is providing students with a formal united tool to share experiences with other students. Experiences such as: past exams, past HWs, evaluations to past courses and professors, and general thoughts about degree plans. This will be provided to students based on their universities then their majors. The platform should be utilized as respectfully and as formally as possible. Students, through their platform profiles, are expected to turn in these information at the end of each semester. Each student will have a pre made profile that will contain their information. The information will be but not limited to: current university, current major, kaust email address, and a small profile picture.

API:

* The platform will be implemented as a restful API using Node.js Express platform. Through Express the platform will communicate with a MongoDB data base to store student’s data. Each student will be required to login into the platform using their KAUST email. Almost all pages in the platform will be generated based on the logged in student. The style of the platform will be minimal in order to ensure scalability and maintainability.

UI:

* The website will have links to the following pages in a navigation bar (except for login page):
  + Login page: Has kaust logo and fields for username and password.
  + Home page: Has a brief explanation of how to navigate the platform.
  + Rules page: Has rules and regulations for using the platform.
  + About page: Has the platform vision and goal.
  + Universities: A list of all universities attended by KGSP students.
    - Each university page will have a list of majors that KGSP students took in the university. The pages will also have a list of students in the university along with their email addresses.
    - Within each major page will be a list of courses that KGSP students took. The page will also have a section for general degree plan advice. Students will have the power to add courses to this list if they have any content to upload along with it and if the course was not available before.
    - Within each course page there will be the martial uploaded by each student along with the name of the professor they took it with, year, and semester. Each student’s content will be accessible as a whole. To keep the website simple, the website will not have any viewer to the content. Here students can add their content to the course page if it already exists. Any type of content can be uploaded.
  + Students will know when there upload of content was successful but the content will not show on website till after it gets reviewed.

Operating demands and cost:

* A computer running around the clock that has stable internet connection as the server. Given the limited demand of website, the server does not have need to have very high performance. The data base can be located at another computer or the same computer as long as it has decent storage capacity and regular back up. If demand proves to be very high for the server even though this is unlikely, the website can potentially be improved to schedule requests using multiple machines at once.
* The platform will require purchasing a domain name unless it will be integrated under KAUST’s public domain page. This is very cheap.

Maintenance and running costs:

* Aside from the electricity the platform will need some consistent monitoring over the content posted. This will include evaluating if the content adheres to the rules and standard of platform. This will also include processing the content to insure it does not breach any academic integrity rules and regulations. The moderator should ensure that content is backed up on an outside source just in case data loss gets lost from server.
* The moderator should have basic knowledge of how to use a MongoDB data base in order to view, approve and delete content.

Future improvements:

* The code for the platform will be uploaded and maintained on GitHub. This will allow collective collaboration by multiple people to improve the platform. Each time improvements are stable they can be compiled and published on the server.