Project – Course Management System

If necessary, additional clarifications may be added to this document but no additional features will be added. Please check this document occasionally. Changes will be clearly indicated in this box.

A certain Head of Department at UDST observed that a large number of students hang around their office during the initial week of classes attempting to change sections, register for courses that have already reached the cap, register for capstone, etc. A large percentage of the problems can be handled very quickly with no need for the student to actually see the head but the department head needs to know what actions to take. Having the students use email to send in their requests is not effective because emails can be easily overlooked and not processed.

In this project you will be implementing a management system that allows students to request actions by the department head through a web application. Once the request has been entered, it will be placed into one of several queues for processing. In fact eventually it is hoped that some basic requests could be processed completely automatically.

Please note that a video presentation of your project is required for this semester.

Requirements

As with many programming projects in real-life, the requirements from the customer are not as detailed as you might hope. This is done to help provide flexibility for the project to allow you to make decisions about how the screens should look and what should be included in the screens. We provide descriptions only of the functionality that is expected.

User Registration and Profile Setup

- Students can set up a profile recording information such as who they are and what program they are in. Just a few fields like name, contact information and degree name is fine; no need to add dozens of fields here.
- Users must go through an email verification process (see note about emails in a later section) before their accounts can be used.
- No captchas are required, and two-factor authentication is not necessary.
- If you would like to use a 3rd party authentication system such as auth0, you are permitted.

Student Accounts

- Students will be able to enter and cancel requests. It must be possible to see cancelled requests. Requests should be recording the time of entry and an estimated processing time (see a later section for this details).
- The student must be able to see their past requests but of course filtered by semester.
- The student must receive an email when the request is processed.

Requests

- Each request must have a category (the categories are up to you but should reflect reality; ask your friends which reasons they see the department head and try to classify them).

- Each category will end up in a different “Queue” for processing and they will be stored in the queue on a first-come-first-served basis.
- When a request is entered into a queue, a processing time will be estimated. The estimate time should be based on the number of requests currently in the queue. For example if you think that a typical request takes 15 minutes and there are 20 requests in the queue then the estimated completion time will be around 5 hours from now. Of course you should only consider working hours so requesting something at 5pm will probably be processed around 1:00pm the next day.

Department Head Account

- The department head should be able to see each “Queue” of requests on a landing page when they sign in (there should be some statistics about how many requests are in the queue along with any other things).
- Clicking on a queue should provide a list of all current requests.
- Clicking on a request will show the details of the request and provide a place where the head can enter a note about the request and either resolve it or reject it. Either way an email gets sent to the student about the resolution.
- Provide a “I don’t know where to begin!” option for the department head that will select any random request from any queue and show the department head the details. This will provide a way for the department head to process things when they are overwhelmed and have no idea of where to start.

Email

- The system will not actually send any emails. Please just use console.log() to display information that would be sent but provide a module that would make it easy to integrate a real email system if you wanted to.

Technologies

The allowed technologies for this project are Node.js, Express.js, and MongoDB. You may utilize the Node modules covered in the course only. If you have some new module that you would like to include, please check with your instructor but do expect that the answer is no.

You are not permitted to use express-sessions or any existing package for CSRF protection. Penalties will be applied for solutions using this.

The only allowed client-side JavaScript is for form validation. Although this is very restrictive, it keeps everybody on the same level.

Security

Provide one example of CSRF protection.

CoreUI Template

Given that the INFS3201 course focuses on back-end programming, we will use an existing HTML5 template called CoreUI. You must use this template; it is not optional. A copy of the template is available in D2L.

The template includes a collection of CSS files and example pages that provide a consistent and professional design. Note that the template contains example pages—you will need to modify and extend these to fit the

project's requirements. The template shows a page called ‘Charts’ but you certainly do not need a page called charts in your project so just remove it from the menu.

GitHub

- You must use GitHub.com to store the code for this project. Changes to the code as you progress through the project must be committed to the repository.
- A regular free account is sufficient; there is no need for GitHub Pro or GitHub Education.
- You may use GitHub Desktop, Git Bash, or any other application to interact with the repository. GitHub Desktop is user-friendly and recommended.
- The use of features such as Issues, Projects, Pull Requests, and Branches is optional and up to individual groups. For this project, the essential operations are pull, push, and commit.
- You must sign up for an account on GitHub.com using your UDST email address. If you have difficulty configuring or using the account, assistance is available from the CCIT help center or the course instructors.
- One team member should create a private repository and add other team members and the instructor as collaborators. Please confirm the instructor's GitHub username or preferred email address for collaboration.

Grading Rubric

Grading rubrics will be provided in a separate document.

Due Dates

Team Formation and GitHub: March 6 @ 11:59pm (-2 points penalty for late)

Being able to work in a team environment is an important skill that is necessary to work in a real-life career outside of school. You are strongly encouraged to work with a group of people who are around your same academic level. Working with others who have similar skill levels will help everybody in the group to improve. We also understand that some situations might arise where collaborative work is not ideal for everyone, therefore it is allowed to work individually. There is no scope adjustment for individual project work.

Teams must consist of between 1 and 3 students. Any requests for groups of 4 or larger will be denied with a penalty of -1 point. Once you have decided on your group, email your LAB INSTRUCTOR (or see them in class) so they can configure the group dropboxes.

Once you have formed the team, create the project on GitHub (keep it private) and add the members plus the instructor as collaborators. Please check with the instructor about the email address to use; it is expected they will use their UDST credentials however they might have an existing account they would prefer. Upload a copy of the link to the ‘Project GitHub Account Dropbox’.

INFS3201Project

Milestone Submission: March 20 @ 11:59pm (8 points)

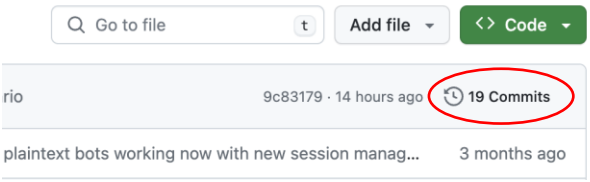
The milestone submission is an early submission to encourage students to work sooner on the project and not leave everything until the last minute.

- **Items to be Evaluated:**
  - The CoreUI template does not need to be fully implemented at this time, but integration is encouraged.
  - The user registration system is operational (students can sign up, validate their email, and log in; the department head can sign in and see something different).
- **Submission Instructions:**
  - Submit a zip file of the code to the **D2L dropbox**. While the instructor is a collaborator on GitHub, submitting to D2L is required for record-keeping.

You must submit a zip file of the code to the D2L dropbox. While the instructor is a collaborator on GitHub, we must have a copy of the project submitted to D2L.

Collaborative Work: April 13 (2 points)

Each team member is expected to be working on the project from the start until the end. Please submit screenshots of the commits for your project to the appropriate dropbox. You can find this history by clicking on the number of commits.



Put screenshot or screenshots of the page from GitHub.com into a word document and submit to the D2L dropbox. Although the instructor has an account, D2L is still the official place where things need to be submitted for archiving purposes.

The instructor will review to ensure that team members are contributing equally to the project through the project period. Make sure that you do not have just one person doing all the commits (unless of course you are a single person working in the project) and make sure that you do not have just a lot of activity during the last day.

All commits must have reasonable comments explaining what has been changed in the committed work.

The image(s) of the history must be submitted on March 13. You can submit this early if the project is completed.

Final Delivery: April 13 @ 8:00am (20 points)

All functionalities will be evaluated at this time.

Because of the Eid Holiday and the partial last week, you must provide a video demonstrating your completed project. Please prepare a 10-minute (no longer) video walking through the features of your project. Place the

video on YouTube and provide a link; the video of course can be made private so that only people with the link can view it.

If there are special configuration instructions, you need to inform the instructor in the form of a README.md file. Without such a file, the instructor will run “npm install” to install any packages specified and then run “node web” to run the application called web.js.

If there are concerns regarding the originality of the work, your lab instructor may request that you attend an interview.

You must submit a zip file of the code to the D2L dropbox. While the instructor is a collaborator on GitHub, we must have a copy of the project submitted to D2L.