CS 3733 – 2025 C Term
Term Project

I. Project Description

Although many students believe that research is only done by professors or graduate students, there are many research opportunities for undergraduate students. Typically, faculty present research opportunities in their classes, and students interested in the project contact the faculty directly. However, if the faculty member is not teaching a lower-level course, they will find it difficult to reach out to the sophomore and junior cohorts. There is a need for an online platform where faculty can advertise research positions and connect with qualified undergraduate students.

You will build a web application where:

- students can enter their contact information, completed coursework, research interests, and prior research experience,
- faculty can advertise research opportunities for undergraduate students,
- students can apply for research positions,
- faculty can select the candidates that they would like to interview for the position.

The application should have a "student page" and a "faculty page".

On the student page, a student user can:

1. Create a student account and enter the profile information:
   - Enter account username and password
   - Enter contact information (name, last name, WPI ID, email, phone)
   - Enter additional information (major, cumulative GPA, expected graduation date, etc.)
   - Select the research topics they are interested in (e.g., "Machine Learning", "High Performance Computing", etc.)
     - You should maintain a list of research fields in your application and have the students choose among those. Only faculty should be able to update this list.
   - Choose the programming languages that they are familiar with.
     - You should maintain a list of programming languages in your application and have the students choose among those. Only faculty should be able to update this list.
2. Login with username and password.
   - Students should be able to login either using their WPI email and password or using the Azure Single Sign-On (SSO) service through WPI.
3. View and edit their account profile.
4. View the open research positions.
   - Your app should list all open research positions.
   - In addition, it should identify the research positions that match the student's user profile and list them separately under the "Recommended Research Positions". For

example: *If the student's research interests include "Machine Learning", the positions in that field should be recommended to the student. Similarly, if the position requires Python experience and if the student chose Python in their profile, the position should be recommended to the student. You should come up with your own ranking algorithm to* find and rank the matching positions. All recommended positions should be ranked according to their relevance (from most relevant to least).

5. For each research position, various information should be displayed:

   - Research project title
   - A brief description of the project goals and objectives
   - Start date and end date
   - Number of students that will be hired to the position.
   - Required time commitment (e.g., 10 hours per week)
   - Research field(s) that the position is related to (e.g., "Machine Learning", "High Performance Computing", etc.)
   - Required programming language experience (e.g., "C++", "Java", "Python", etc.)
   - The name and contact information of the faculty who posted the position.

6. Apply for research positions. A student can apply for more than one research position. For each position they apply to,

   - They should submit a brief statement describing why they are interested in this research topic and what they hope to gain by participating in that project.
   - They should provide the name and email of one faculty who can provide them a reference for the position. This faculty should be one who is already registered in the system. When a student lists a faculty as reference for their application, the faculty member should receive a message asking them if they recommend the student for the position.

7. View the research positions they already applied to and check the statuses of their applications and reference requests.

   - When the application is submitted, its status will appear as "*Pending*".
   - When the owner of the research position (faculty) accepts this application, the application status should be updated as "*Approved*".
   - If the owner of the research position (faculty) rejects this application, the application status should be updated as "*Rejected*".
   - When the reference (i.e., the faculty listed as reference) confirms the recommendation, the recommendation standing should be updated as "Recommended".
   - Once updated, the changed status (for both application and recommendation) should be displayed on the student page.

8. Withdraw their "pending" applications.

   - If the student is no longer interested in a research position, they can withdraw their application.

– If the status of the application is “*Approved”*, they can’t withdraw it.

On the faculty page, a faculty user can:

1. Create a faculty account and enter profile information:
   - Set the account username and password
   - Enter contact information (name, last name, WPI ID, email, phone)
2. Login with username and password
   - Students should be able to login either using their WPI email and password or using the Azure Single Sign-On (SSO) service through WPI.
3. View their account profile.
   - In addition to their profile information, the recommendation requests the faculty received should be shown on the profile page. They should be able to approve or reject those requests.
4. Create undergraduate research positions. Faculty should enter the details of the position and qualifications needed, i.e.,:
   a. Research project title,
   b. A brief description of the project goals and objectives
   c. Start date and end date
   d. Required time commitment (e.g., 10 hours per week)
   e. The number of students that will be hired for the position.
   f. Research field(s) (e.g., “Machine Learning”, “High Performance Computing”, etc.). You should assume a predetermined list of research fields and have the instructor choose among those.
      - You should maintain a list of research fields in your application and have the students choose among those. Any faculty should be able to update this list.
   g. Required experience with programming languages (e.g., “C++”, “Java”, “Python”, etc.)
      - You should maintain a list of research fields in your application and have the students choose among those. Any faculty should be able to update this list.

   Of course, a faculty can create more than one positions.
5. See the list of the students who applied for their positions.
   - A faculty should be informed about the other positions students are approved. If a student was “*Approved*” for another position, those information should also be displayed.
6. View the qualifications of each student, i.e.,
   a. their GPAs,
   b. the research topics they are interested in,
   c. the programming languages they have experience with, and
   d. prior research experience,

e. the reference student provided in their application and whether the recommendations is approved.

7. The faculty can approve the application of one or more students and the status of those applications should be updated as “*Approved*”. The number of students faculty can approve for a position should be limited to the number of students who can be hired for that position.
8. The faculty can update the application status to “*Rejected* ” if the student is not qualified for the position. Student should see the status change for their application once it is updated.

Additional requirement: You should implement at least two features on your user interface using JavaScript. Make sure those are not very trivial use cases and the user experience is improved through the use of JavaScript.

Project Timeline:

i. **Requirements and Use-cases: (65pts + 5pts)**
   - Once we form the teams, you will have to start right away getting the team working together and collecting the requirements. In this stage you need to elaborate the details of the requirements for the complete project.
   - Since there are only a 6 days after we form the teams until the full **Requirements and Use-cases** document is due, it is important for the team to start meeting right away, because it takes time to figure out what the team will be building.
   - The requirements document should be pushed to GitHub repo before the deadline.
   - You must start using **GitHub Issues** (Issues, Labels, and Milestones) to manage your Product Backlog.**(10 pts)**

ii. **Design Document – draft (10 pts)**
   - You will summarize your design for your application.
   - We will provide feedback on your design and you will revise your design document accordingly.

iii. **Design Document – final (70 pts)**
   - This will be the revised version of your design document.

iv. **Iterations:** The main part of the project will be done in several iterations. In each iteration you will have to work on several tasks:

   - **Iteration 1 (75 pts):**
     - Complete a running version of your code with a partial set of features.
     - Your code should be pushed to GitHub repo before the iteration deadline. You will demo your application to the SAs and instructor.

- – Write a short group report where you briefly summarize the project progress. You will push this report to your GitHub repo (should be on main branch under “documents” folder)

– **Iteration 2: (85 pts)**

- – Complete a running version of your code with additional set of features.
- – Your code should be pushed to GitHub repo before the iteration deadline. You will demo your application to the SAs and instructor.
- – Write a short group report where you briefly summarize the project progress. You will push this report to your GitHub repo (should be on main branch under “documents” folder)

– **Iteration 3: (105 pts + 25 pts)**

- – A final version of your code. All code should be uploaded to GitHub repo before the iteration deadline.
- – Your code should be pushed to GitHub repo before the iteration deadline. You will demo your application to the SAs and instructor.
- – This iteration should include automated unit and functional tests. All test code should be uploaded to GitHub repo before the iteration deadline.

II. Project Deliverables:

1. Requirements and Use-cases:

There is no standard for requirements documents and in fact many organizations blend aspects of requirements, and specification in a single document. We will use a simple template for requirements and specification and document requirements as user stories and UML use cases. Inevitably in preparing this document you will describe of how the system should interact with the outside world.

Please see the “Requirements and Use-cases” document template that will be posted on Canvas.

2. Design Document:

This will be a living document. For the first iteration you will fill in the document with the draft of your software design. You should include as many design details as possible according to your current understanding of the final application. In subsequent iteration you will revise this document.

A template for the “Design Document” will be available on Canvas.

3. Group Report:

For iterations 1 and 2 you will write a short progress report covering the topics specified below. This report will be included at the end of your design document.

- **Main Difficulties:** What were the main difficulties so far? You should consider both technical and organization issues.
- **Features:** What user stories were completed? Were there any user stories you did not implement as planned? Are you pushing some features to later iterations, and if so, why?

III. Submission Instructions:

All project documents including “Requirements and Use-cases” and “Design Document” (both versions) should be committed to your team’s GitHub repo.
All code and test case implementations should also be committed to your GitHub repo.

Your GitHub repo should have the following branch structure :

- main:
  - Final version of your code (after iteration 3);
  - (under “tests” directory) all your test code (after iteration 3);
  - (under documents directory) Requirements and Use-cases document; draft version of your Design Document ; final version of your Design Document ; your group reports for iterations-1 and -2.
- iIterationX: (replace X with iteration number; don’t use and space or ‘- ‘ before X)
  - All the code you produced in iteration X;
  - Iteration3 only - (under “tests” directory) your test code for iteration 3.

Reading:

https://zube.io/blog/agile-project-management-workflow-for-github-issues/
https://about.gitlab.com/blog/2018/03/05/gitlab-for-agile-software-development/
https://robinpowered.com/blog/best-practice-system-for-organizing-and-tagging-github-issues/
https://about.gitlab.com/devops-tools/jira-vs-gitlab.html

Appendix:

Iteration-1 use-cases
Iteration-2 use-cases
Iteration-3 use-cases

On the student page, a student user can:

1. Create a student account and enter the profile information:
   - Enter account username and password
   - Enter contact information (name, last name, WPI ID, email, phone)
   - Enter additional information (major, cumulative GPA, expected graduation date, etc.)
   - Select the research topics they are interested in (e.g., “Machine Learning”, “High Performance Computing”, etc.)
     - You should maintain a list of research fields in your application and have the students choose among those. Only faculty should be able to update this list.
   - Choose the programming languages that they are familiar with.
     - You should maintain a list of programming languages in your application and have the students choose among those. Only faculty should be able to update this list.
2. Login with username and password.

- Students should be able to login either using their WPI email and password or using the Azure Single Sign-On (SSO) service through WPI.

3. View and edit their account profile.

4. View the open research positions.

   - Your app should list all open research positions.
   - In addition, it should identify the research positions that match the student's user profile and list them separately under the "Recommended Research Positions". For example: *If the student's research interests include "Machine Learning", the positions in that field should be recommended to the student. Similarly, if the position requires Python experience and if the student chose Python in their profile, the position should be recommended to the student. You should come up with your own ranking algorithm to* find and rank the matching positions. All recommended positions should be ranked according to their relevance (from most relevant to least).

5. For each research position, various information should be displayed:

   - Research project title
   - A brief description of the project goals and objectives
   - Start date and end date
   - Number of students that will be hired to the position.
   - Required time commitment (e.g., 10 hours per week)
   - Research field(s) that the position is related to (e.g., "Machine Learning", "High Performance Computing", etc.)
   - Required programming language experience (e.g., "C++", "Java", "Python", etc.)
   - The name and contact information of the faculty who posted the position.

6. Apply for research positions. A student can apply for more than one research position. For each position they apply to,

   - They should submit a brief statement describing why they are interested in this research topic and what they hope to gain by participating in that project.
   - They should provide the name and email of one faculty who can provide them a reference for the position. This faculty should be one who is already registered in the system. When a student lists a faculty as reference for their application, the faculty member should receive a message asking them if they recommend the student for the position.

7. View the research positions they already applied to and check the statuses of their applications and reference requests.

   - When the application is submitted, its status will appear as "*Pending*".
   - When the owner of the research position (faculty) accepts this application, the application status should be updated as "*Approved*".
   - If the owner of the research position (faculty) rejects this application, the application status should be updated as "*Rejected*".

- When the reference (i.e., the faculty listed as reference) confirms the recommendation, the recommendation standing should be updated as "Recommended".
- Once updated, the changed status (for both application and recommendation) should be displayed on the student page.

8. Withdraw their "pending" applications.

- If the student is no longer interested in a research position, they can withdraw their application.
- If the status of the application is "*Approved*", they can't withdraw it.

On the faculty page, a faculty user can:

9. Create a faculty account and enter profile information:
   - Set the account username and password
   - Enter contact information (name, last name, WPI ID, email, phone)

10. Login with username and password

- Students should be able to login either using their WPI email and password or using the Azure Single Sign-On (SSO) service through WPI.

11. View their account profile.

- In addition to their profile information, the recommendation requests the faculty received should be shown on the profile page. They should be able to approve or reject those requests.

12. Create undergraduate research positions. Faculty should enter the details of the position and qualifications needed, i.e.,:

h. Research project title,
i. A brief description of the project goals and objectives
j. Start date and end date
k. Required time commitment (e.g., 10 hours per week)
l. The number of students that will be hired for the position.
m. Research field(s) (e.g., "Machine Learning", "High Performance Computing", etc.). You should assume a predetermined list of research fields and have the instructor choose among those.
   - You should maintain a list of research fields in your application and have the students choose among those. Any faculty should be able to update this list.

n. Required experience with programming languages (e.g., "C++", "Java", "Python", etc.)
   - You should maintain a list of research fields in your application and have the students choose among those. Any faculty should be able to update this list.

Of course, a faculty can create more than one positions.

13. See the list of the students who applied for their positions.

- A faculty should be informed about the other positions students are approved. If a student was "*Approved*" for another position, those information should also be displayed.

14. View the qualifications of each student, i.e.,

    f. their GPAs,
    g. the research topics they are interested in,
    h. the programming languages they have experience with, and
    i. prior research experience,
    j. the reference student provided in their application and whether the recommendations is approved.

15. The faculty can approve the application of one or more students and the status of those applications should be updated as "*Approved*". The number of students faculty can approve for a position should be limited to the number of students who can be hired for that position.

16. The faculty can update the application status to "*Rejected* " if the student is not qualified for the position. Student should see the status change for their application once it is updated.

Additional requirement: You should implement at least two features on your user interface using JavaScript. Make sure those are not very trivial use cases and the user experience is improved through the use of JavaScript.