ApplyDirect

12.07.2020

Jantzen Allphin, Alexander Stewart, Tri Tran

University of Utah | CS4540: Web Software Architecture

Professor H. James de St. Germain

Abstract

This report details our changes to the URC project started in class. Our key additions to the base project include: the ability for students to apply to opportunities, the tools for professors to manage these applications, additional security considerations, changes to the ui and the inclusion of useful opportunity data for students. These additions range from being fundamental functionality for any research coordinator to useful features that take our coordinator beyond the educational baseline and we believe that all of them will improve the core user experience.

URLs

Jantzen Allphin: https://ec2-107-21-81-94.compute-1.amazonaws.com

Alexander Stewart: https://ec2-54-209-6-159.compute-1.amazonaws.com

Tri Tran: https://ec2-34-232-70-54.compute-1.amazonaws.com

1. Introduction

The original URC project that was started in class lacked a lot of key features that prevented it from being useful to users aside from being a learning experience. We noticed in particular that the ability for students to interact with opportunities was completely absent from the starting point completed in class. This results in a research coordinator that lacks the ability to actually coordinate research. To alleviate this, we added the ability for students to directly apply to research opportunities (hence the group name). The major change needed to allow for this functionality was the inclusion of a student profile model that holds information about each student. This profile is then visible to the student who created it as well as any professors that host opportunities the student has applied to. Since students can apply to opportunities, we created a way for professors to manage the application which allows them to either accept or reject any applications.

To further facilitate research coordination we implemented features that improve the ability for students to obtain research opportunities. These include suggesting relevant opportunities on the student profile as well as displaying the most popular student skills and the most popular required skills for opportunities. The thought process behind these additions is that if the project is a success, there will be a plethora of opportunities available to students, but it would be on the students to gauge what opportunities to pursue and what skills to continue to hone. By supplying recommended research positions, students can quickly apply to positions that are likely to be a good fit, and if there are no recommendations, they can use the most popular required skills as a guide for what skills are currently desired.

With the addition of core functionality to the URC project, the website still lacked the professional feel and security of a trustworthy website. All pages have been redesigned and unified to give a more seamless experience, including CRUD scaffolded pages and more backend pages like the professor management page. With the UI handled, we began to notice some glaring security issues that would leave the website vulnerable to attacks from hackers. The main attack vectors were image uploading and resume uploading/downloading which previously used user input as part of the download field. Functionality was added to protect against sql-injection attacks and others like it.

2. Feature Table

Feature Name	Scope	Primary Programmer	Time Spent	File/Function	LoC
New Application model including controller/views	Full	Jantzen Allphin	4hrs	Application.cs, ApplicationController.cs, Views/Applications/*.cshtml	609
Students can now directly apply to a specific opportunity	Full	Jantzen Allphin	4hrs	Opportunities/Details.cshtml, Application.cs, ApplicationController.cs	159
Professors can manage submitted applications	Full	Jantzen Allphin	10hrs	OpportunitiesController.cs:: Manage(), Manage.cshtml, AcceptDenyApplication.js, Opportunity.cs	270
Old Application model converted to represent a Student Profile	Back-end, DB	Jantzen Allphin	4hrs	Application.cs, Student.cs, ApplicationsController.cs, StudentsController.cs, Views/Application/*.cshtml, Views/Student/*.cshtml	800
Improved UI/UX design and made consistent across entire website	Front-End	Jantzen Allphin	10hrs	wwwroot/css/site.css, Views/*	3095
Students receive relevant opportunity	Full	Alex Stewart	8hrs	OpportunitiesController.cs, StudentsController.cs, URC_Context.cs,	300

suggestions				Recommendation.cs, Student.cs	
Display popular student skills	Full	Alex Stewart	5hrs	StudentsController.cs, URC_Context.cs, PopularStudentSkill.cs,	200
Display popular required skills	Full	Alex Stewart	3hrs	OpportunitiesController.cs, StudentsController.cs, URC_Context.cs, PopularRequiredSkill.cs	200
Unified code base	Full	Alex Stewart	3hrs	All files were touched, but most changes were just choosing the correct code to retain.	400
Validate upload file	Back-end	Tri Tran	3hrs	Change to ApplicationController.cs OpportunityController.cs	100
Sanitized User inputs to prevent Javascript injection	Full	Tri Tran	4hrs	Change to all controllers and cshtml that require user input	50

3. Individual Contribution Table

Team Member	**Time Spent on Project**	**Lines of Code Committed**
Jantzen Allphin	32hrs	4933
Alexander Stewart	19hrs	1100
Tri Tran	11hrs	150

Note: Jantzen Allphin's LoC is much greater than the rest of the team. However, this is simply because of the tasks he completed. For example, he re-worked the website design and made that consistent across the entire website. This caused him to modify almost every front-end file in the project. Naturally this will cause his LoC to be higher.

4. Individual Contribution Summary

Jantzen Allphin

Jantzen's main contributions were in implementing the "direct apply" functionality, including both student and professor experience. This feature allows the student to apply directly to a specific opportunity using a unique application and manage all their applications. It also allows the professor to manage the applications to his/her opportunity, including an accept/deny function for easy acceptance/denial of applicants. Jantzen also designed and implemented the website UI/UX look and feel. Completing this task made the website more professional and consistent across all web pages. His contributions required work full-stack.

Alexander Stewart

Alex's main contributions were handling the implementation of data focused changes to improve the student experience. These changes include providing relevant research opportunities based on student skills, displaying the most popular student skills and displaying the most popular skills required for opportunities. The addition of these features required adding models and changing some of the existing models, as well as more extensive usage of LINQ than required in class. Alex's changes were mostly on the model and controller side with only some changes to the view for displaying the updated values from the model. Alex also handled the merges of the team's code base.

Tri Tran

Since Tri worked on adding a security part of the website, he implemented the validation to make sure the user submits a correct file format such as image and pdf for the student

application. This helps to ensure no malicious file is uploaded by changing the extension. Because some of our web pages expect user input, we also needed a way to sanitize user input in order to prevent javascript attacks which hackers can use to steal data or delete data.

5. Performance Level

Good

Despite running into a few obstacles in our team development, we performed at a good level. Some of the obstacles we faced included bugs in the pre-existing code, merge conflicts, and task/schedule management. The majority of the most important parts of our project regarded the professor and student experiences. Thus, the MVC components for opportunities, applications, and students were a point of merge conflicts. We tackled merge conflicts by adequately preparing our separate git branches for the merging process. We communicated well to ensure the merging process was done correctly and at the right time. We masterfully overcame our task/schedule management problems by using a kanban board from Atlassian's JIRA. This allowed us to see each other's progress on tasks and easily assign out other needed tasks. Lastly, we collaborated efficiently via a Discord server to fix any bugs found in the pre-existing code. We provided substantial functionality while overcoming and developing as a team.

6. Summary

Our project adds substantial functionality to the URC program that enhances the direct research coordination tasks between any student and professor. It simplifies professors' task in managing applicants to their respective research opportunities, and it significantly improves students' application process and profile management. The website design is polished and attractive with a uniform look across all web pages, and the user experience is intuitive with seamless navigation between all web pages. Students will find it far easier to utilize the URC program to find research opportunities with the addition of research suggestions and by knowing the most sought after skills, they can learn what to focus on.

Website security was improved to protect against attacks specifically when uploading and downloading files to the program. This provides comfort to the professor who is likely to download student resumes which are uploaded as part of the application to their research opportunity.