Justin Haycraft

Sharan Ravula

Jonah Burgess

Nnaemeka Okafor

Description of Tasks

- *What project task did your team select? What factors went into that selection? In retrospect, given what you have learned by working on the tasks, would you have selected a different task?*

  We decided to do our own website for the project instead of working on Dungeon Crawl: Stone Soup, although our final decision was between that and adding UI elements to the settings. Originally we had a few members who were just interested in pursuing the DC:SS UI option because it seemed like a unique experience that would crack open the open-source contribution experience while learning new skills in the process, but ultimately we made the team decision towards the website because we wanted something that aligned more with the corporate workplace atmosphere, focusing on organizing resumes. In retrospect, I think taking what we've learned from this project will be useful—and it's difficult to add much more on to this or expect a much more fruitful alternative, but we do feel that we could have spiced it up in our planning, even if we still pursued a website. Our expertise starting out at the beginning of the semester was quite limiting, though, so coming into it with more knowledge would already help with that hunch.

Description of User Stories

- *What user stories did your team select for Milestone 2 and, if applicable, as the additional story for this assignment? How did you refine those stories from the original stories you wrote for Milestone 1?*

  The user stories we chose were complementary functionalities. We wanted to allow a user to upload a resume to the database and view it on the website, and also to be able to delete it from the website. The story we added onto it was a search functionality to view all of the resumes in the database. We refined them by connecting them to each other to make each user story feel like a whole project.

Code Location

- *Where can your team's source code be found? List at least the github repository and the branch, but also describe which files you modified and/or added, and give a brief summary of the modifications you made.*

  https://github.com/Autoume/autoume.git

  https://autoume-41f5f.web.app/

  The files we modified can be primarily found under ~/autoume/src/app/

  They were for giving the specific functionality on the pages where the vast bulk of the other files supported the backend and fundamental functions of the site.

Design Elements

- *What functions, classes, and other design elements did your team implement? How do the new elements relate to one another? List important relationships such as*

caller-callee relationships between functions, and inheritance and "contains" relationships between classes.

Our design was very modular in nature, so it primarily consisted of building little sections for the backend to put together and display on the page.

To provide specific examples for upload management:

- handleFileChange : Validates file type and size
- handleUpload : Manages file storage and database entry creation
- handleDelete : Removes files from storage and database

Examples for database interactions:

- fetchUploadedPapers : Retrieves resume listings
- addDoc : Creates new database entries
- getDocs : Retrieves existing documents

We utilized an encapsulated structure where our page implementations borrowed from other areas in the repository. For example, our pages inherit from the Card, Button, Alert, and Input components from the src/components/ui directory.

Sources of Additional Information

- *What sources of additional information, help, or advice did your team seek out? Which of those, if any, did you find the most helpful?*

  A lot of our information gathering was done mostly independently on google and youtube, looking up and sharing tutorials to catch each other up with the skills necessary to develop the website. Most of it was very new to us so we spent a lot of time learning what we needed.

Software Engineering Techniques

- *What software engineering techniques, whether from Agile development or other methodologies, did your team use? In retrospect, are there any other techniques you wish you had used?*

    We primarily just took it head-on in a similar way to waterfall. Taking each step one at a time and implementing it all one by one without much concern for organization. In retrospect, prioritizing organization would make returning to fix bugs and manage additions easier. Using the Agile methodology seems to be best for our needs, and would be wise to do if we were to do this project again.

Technologies Learned

- *What technologies, programming techniques, and/or libraries did your team have to learn to accomplish your tasks? What previous experience of your team members turned out to be useful?*

    We used firebase to get the backend up and running for the site. Experience with this program proved to be very useful because it was like skipping an entire step in the setup process, but only one of our team members had experience, so the rest of us had to learn it.

Lessons Learned

- *What non-technical lessons has your team learned from your work this project? What advice would you give to a future team working on this or a similar project?*

    We've learned and practiced essential communication and planning skills. A project like this takes a lot of initiative and teamwork, and having effective communication to piece everything together and on time is essential. My

advice would be to make sure communication is heavily prioritized and to set specific deadlines to give your team the wiggle room should something unexpected happen. We've gotten into the habit of giving two extra days, so we set the deadline two days prior to the due date at latest.