CMPE 492

A Web Service for Senior Projects

Nurlan Dadashov

Burak Yılmaz

Advisor:

Doğan Ulus

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Broad Impact

By creating a comprehensive repository of information regarding the senior projects accomplished at Bogazici University, our project strives to offer a valuable resource to various groups of stakeholders. Our project also aims to help junior and project observer students by allowing them to familiarize themselves with the senior project processes and requirements. Furthermore, we hope to aid prospective students in the decision-making process regarding their choice of a university by showcasing the academic achievements of the senior project students at Bogazici University. Additionally, our project intends to offer valuable insights into the senior project outcomes to outsiders, such as employers, foreign professors, ABET Judges, and similar parties. Lastly, by creating a user-friendly and informative website, we aim to provide a practical and efficient platform for website maintainers to manage and update the information related to senior projects at Bogazici University.

1.2. Ethical Considerations

Our site is a static one, stored in a public GitHub repository, which means that anyone can access the content we gather, including senior project information. We must be particularly careful with the ethical considerations of our project as a result. While we won't collect personal information, we may collect public links users provide. Therefore, we will communicate our intentions clearly and obtain consent from anyone whose information we collect.

Regarding advisors posting project offers on the website, we will grant them admin permissions but limit their actions to creating new project offers. They will not be able to edit or delete posts created by other individuals. Maintaining transparency and fairness in the process of posting and sharing project offers on the website is

important to us.

Site administrators will review the content provided in the pull requests for appropriateness and the absence of harmful or malicious material.

We will also ensure that the site design and content are accessible to all users, regardless of abilities or disabilities. We will follow web accessibility guidelines to ensure that the site is usable by everyone.

2. PROJECT DEFINITION AND PLANNING

2.1. Project Definition

The goal of this project is to design and implement a web service that publishes senior project offers from faculty to students and disseminates project results to a wider audience. The project roadmap included requirements collection, the survey of available technologies, service design, implementation, and testing. To ensure simplicity, the focus was on maintenance and reliability. For this, we have containerized authentication server for the cms and defined a workflow for building and deploying automatically from the repository via cloud-native tooling.

2.2. Project Planning

2.2.1. Project Time and Resource Estimation

- Requirements collection: 2 weeks
- Survey of available technologies: 1 week
- Designing the service: 2-3 weeks
- Implementation and testing: 4 weeks

This timeline includes some buffer time in case there are any unexpected delays or challenges that arise during the project.

2.2.2. Success Criteria

- Department using and highlighting the project as the main page for senior projects.

2.2.3. Risk Analysis

Project Risks

There are several risks associated with this project, including:

- Department not using the project or advisors not adapting.
- Using Github: If they decide to go private or premium, then the project would suffer severely. In that case, the project would require some other host with version control system ‘git‘ support. That would also create compatibility issues with DecapCMS.
- Decap CMS being deprecated.
- Custom auth server malfunctioning and being out of support.

Risk Mitigation

To mitigate these risks, we will:

- Conduct thorough research and testing of available technologies before making any decisions.
- Establish a rigorous testing and maintenance plan to ensure the service is reliable and well-maintained.
- For Github case, we don’t have much that we can do, since most of the backbone for all open-source projects relies on GitHub. We just hope that this won’t be the case.
- Decap CMS was named NetlifyCMS and archived. However, it was migrated to another repository to keep its availability of it. Now Decap CMS is open source and with lots of stars on GitHub. If Decap CMS again becomes deprecated, we are sure that someone else will take over and continue the project.
- Instead of writing our own auth server and maintaining it, we have found a custom

external auth server for github authentication for the CMS[1] . For least amount of issues, we have published a docker image[2] of this.

2.2.4. Team Work (if applicable)

As a team, we understand the importance of effective communication and collaboration in ensuring the success of this project. To accomplish this, we decided to use GitHub Projects for issue management and Discord for communication and project management. GitHub Projects provided us with the progress of each task and enabled us to track any issues or bugs that could arise during the development process. We used Discord to communicate with each other, share ideas, and discuss anything that we felt needed discussion. Discord was also the main communication channel with our advisor as well.

We believed that regular communication with our advisor was really important for the success of this project. We planned to meet with our advisor regularly to discuss our progress and receive feedback on our work, and we did so. We believed that this would help us stay on track and make sure that the project met all of the goals and success criteria.

Overall, we committed to working as a team and to leveraging each other's strengths to deliver a quality senior project website. We are confident that our collaboration, communication, and dedication lead to the successful completion of this project.

[1]https://github.com/vencax/netlify-cms-github-oauth-provider
[2]https://hub.docker.com/r/yilmazburak/decap-cms-oauth-provider

3. RELATED WORK

The current senior projects site for Bogazici University is hosted on

https://www.cmpe.boun.edu.tr/undergraduate/senior-projects

So, the main reference we use is this website. The main differences with the existing website are as follows:

- Overall design is much more digestible and easy on the eyes.
- We offer a much more collected section page.
- We offer a digest mode, where the projects can be easily overviewed, enabling skimming.
- We are also using this website to gather info about the previous years' projects.

For our project, we needed to research many technologies and frameworks such as the following:

- `https://decapcms.org/`
- `https://github.com/vencax/netlify-cms-github-oauth-provider`
- `https://github.com/igk1972/netlify-cms-oauth-provider-go`
- `https://nextjs.org/`
- `https://jekyllrb.com/`

In addition to this, we have also looked for similar projects, such as:

- `https://www.ctis.bilkent.edu.tr/`
- `https://senior.ceng.metu.edu.tr/2022/mainpage/`
- `https://rsl.ethz.ch/education-students/student-projects0.html`
- `https://www.apple.com/newsroom`

4. METHODOLOGY

As the main guideline, we are using trunk-based development, where we periodically update the main branch on our GitHub repository. This way, our advisor is able to see the progress as we do improvements. In support of this, we believe in iterative processing. We develop some features or designs, and then ask the stakeholders about them. According to their feedback, we reiterate the designs and features. Following are our methodologies on the respective topics:

4.1. Requirements Collection

To collect the requirements for the web service, we reached out to the faculty members who supervise senior projects, as well as junior and project observer students. We conducted interviews and surveys to understand the needs and expectations of various stakeholders.

4.2. Survey of Available Technologies

We surveyed available technologies to determine the most suitable options for the web service. We evaluated various technologies based on factors such as ease of use, scalability, maintainability, and reliability. After careful consideration, we selected the Hugo framework, Decap CMS for content management, and custom auth server for the required github oauth server.

4.3. Service Design

Based on the collected requirements, we designed the web service to be user-friendly, informative, and accessible.

4.4. Implementation and Testing

After completing the design phase, we began implementing the web service. We used Decap CMS to build the content management system. We conducted rigorous testing to ensure the reliability of the service. We also ensured that the design and content were accessible to all users, regardless of abilities or disabilities. We have also run 'Lighthouse'[3] . performance tests, which will be shared in detail in the following sections.

[3]More details: `https://developer.chrome.com/docs/lighthouse/overview/`

5. REQUIREMENTS SPECIFICATION

5.1. Functional Requirements

1. **User Requirements**

1.1. **Advisor**

1.1.1. Advisors shall be able to log in to access the CMS.

1.1.2. Advisors shall be able to create project offers via CMS.

1.1.3. Advisors shall have access to the repository in order to give consent to students to take their projects (by merging PRs).

1.2. **Student**

1.2.1. Students shall be able to view project offers for the current semester.

1.2.2. Students shall be able to view past projects.

1.3. **Admin**

1.3.1. Admin shall have access to the repository in order to be able to merge PRs created by students.

2. **System Requirements**

2.1. The system shall support the addition of new project offers (markdown files) via CMS.

2.2. The system shall redeploy webpages upon each change in the repository.

2.3. The system shall preserve past projects.

2.4. The system shall support search.

5.2. Non-Functional Requirements

3. **Accessibility and Availability**

3.1. The platform shall be accessed via a web service.

3.1.1. The web service shall support modern browsers (Chrome, Firefox, Safari).

3.1.2. The web service should be responsive.

3.2. The platform language shall be English.

3.3. The platform should support UTF-8 character encoding.

4. **Security**

4.1. Access to the GitHub repository should be restricted to website admin and advisors.

4.2. The platform shall use the HTTPS Protocol.

5. **Performance and Reliability**

5.1. The platform shall respond to any request in at most 5 seconds.

5.2. The platform should have an uptime of at least 98%.

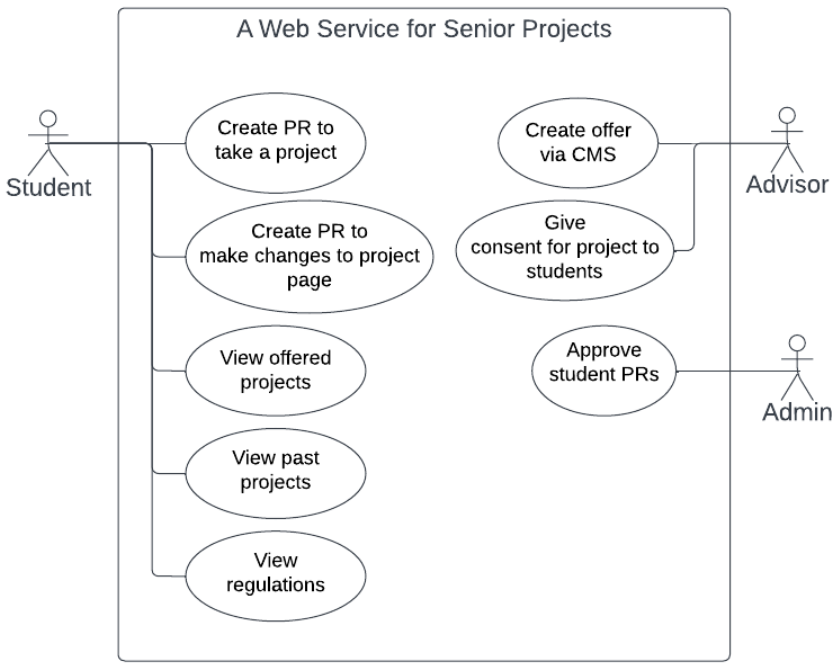

Figure 5.1. Use case diagram

6. DESIGN

6.1. Information Structure

Hugo uses some predefined folder structure for storing the content and applies the relevant styling with html files, which also use similar kind of folder structure.

For our project, we have a main 'contents' folder, which consists of the following directories:

- Offers: This directory holds the current project offers by the advisors.
  - Each of the .md files are created by the Decap CMS. They have relevant advisor info, relevant info about the offer such as the title and body of it.
- Ongoing
  - Each of the .md files are created by the students, who take the project defined in offered pages. They need to remove the project from "Offers" directory and add it to this directory with addition of their names under "students" meta data.
- Semesters
  - <year>-<semester> directory: This directory hosts the completed projects for the relevant year and semester.
    - Each of the .md files corresponds to a page on the relevant year and semester. After removing their projects from the ' Ongoing ' folder, this file is expected to be created by the students.
    - images: This folder hosts the images used in the .md file in the same directory.

6.2. Information Flow

The main flow is as follows:

- Advisor creates an offer through the CMS via signing in as a user. They submit their project offer.
- Senior students head to the website, to the offered projects section.
- They email or somehow create a communication channel with the offering advisor.
- Once they decide on doing the project together, the student sends a PR to the GitHub repo, moving the offered project to the Ongoing Projects section.
- Site admin merges the PR, if everything is good.
- The student can iteratively update the page for the project on ongoing projects throughout the semester.
- Or, they can create their final project file at the end of the semester, again by creating a PR.
- If all is good again, the site administrator merges the changes to the repo.
- As with all the merges, the GitHub action is triggered and the latest form of the main branch is deployed to the URL.

6.3. System Design

- Hugo is used as the site generator for the web application.
- Bootstrap provides the CSS framework for styling and layout.
- Decap CMS is used as the content management system for managing the content of the site.
- GitHub Actions are used to automate tasks and workflows related to the development and deployment of the web application.

The following relations exist between these technologies:

- Hugo and Decap CMS work together to manage the content of the site. Hugo generates the static files based on the content managed by Decap CMS. Bootstrap is used to provide the styling for the generated site.
- GitHub Actions are used to automate tasks related to the development and deployment of the web application. Workflows are defined in YAML files, and

actions are executed by the action executor.

In summary, the system design for the web application includes Hugo as the site generator, Bootstrap for styling, Decap CMS for content management, and GitHub Actions for automating tasks and workflows related to development and deployment. The modular architecture of these technologies allows for easy customization and integration, and the plugin frameworks provide a way to extend the functionality of the system.

6.4. User Interface Design

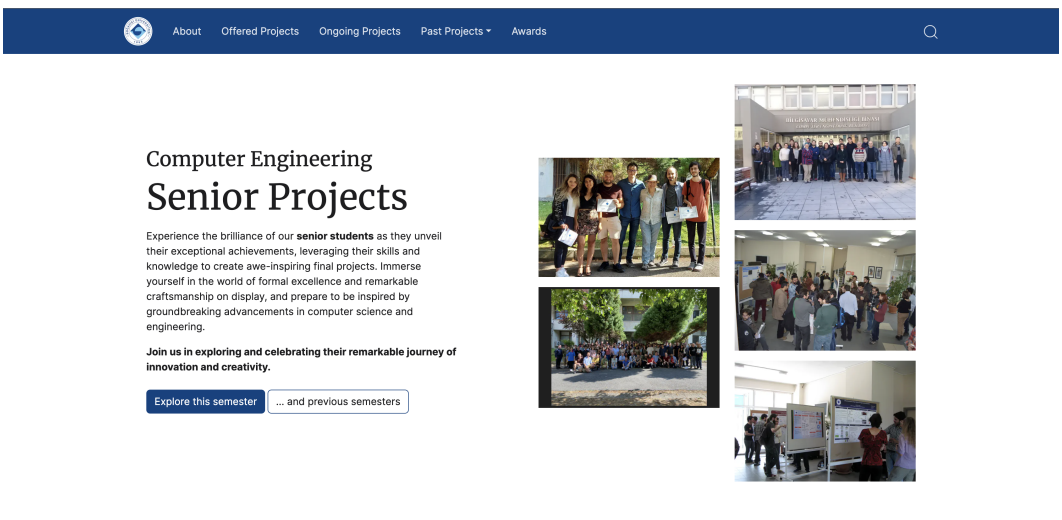

Figure 6.1. Main page design

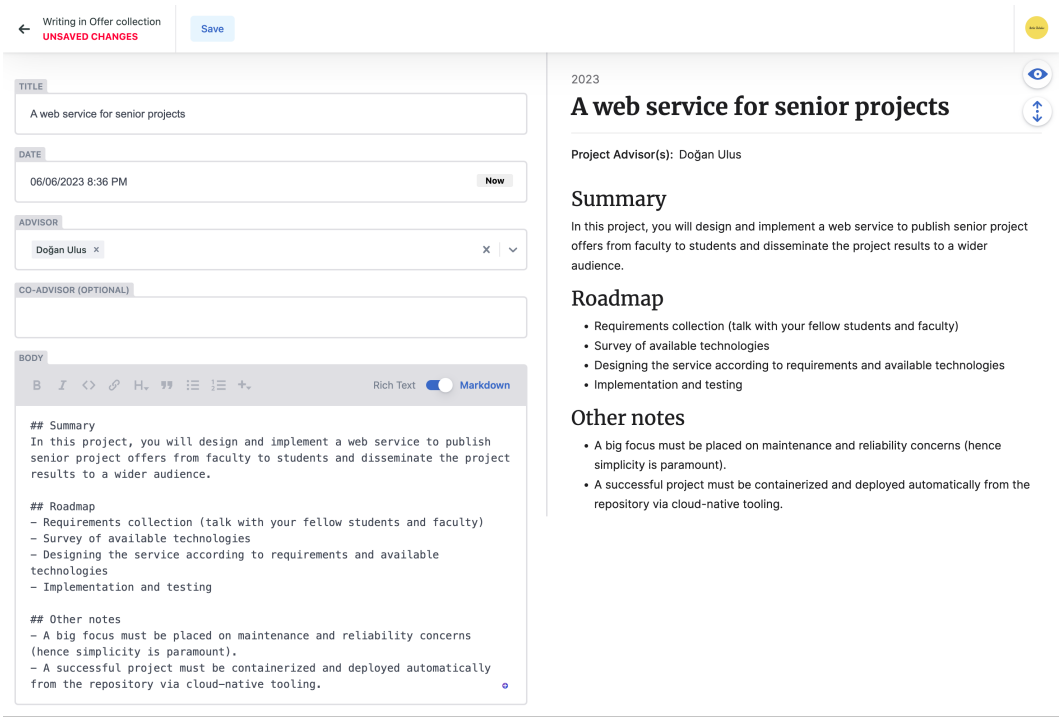

Figure 6.2. CMS page design

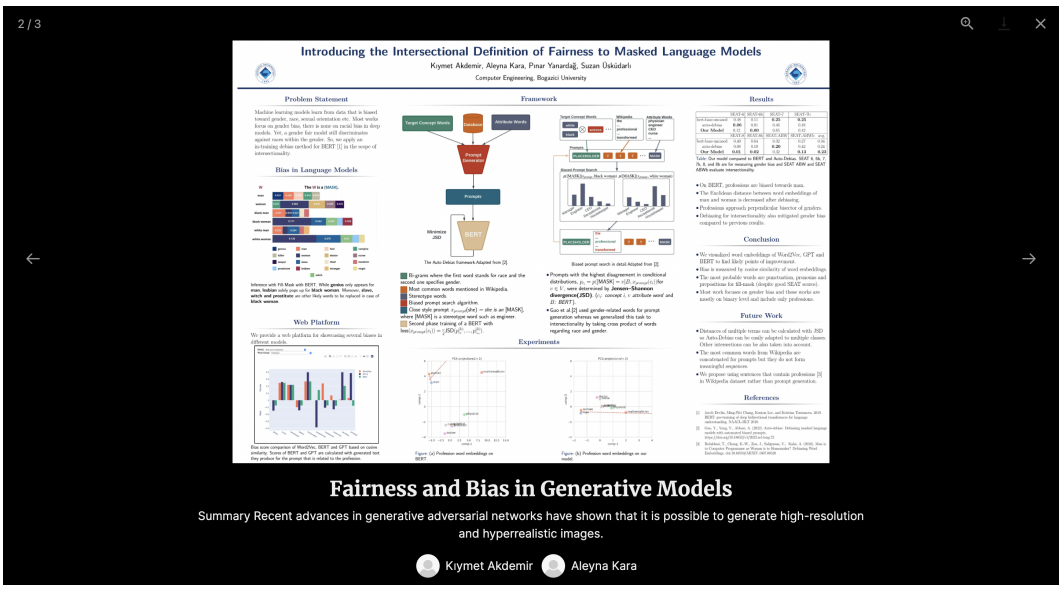

Figure 6.3. Digest mode design

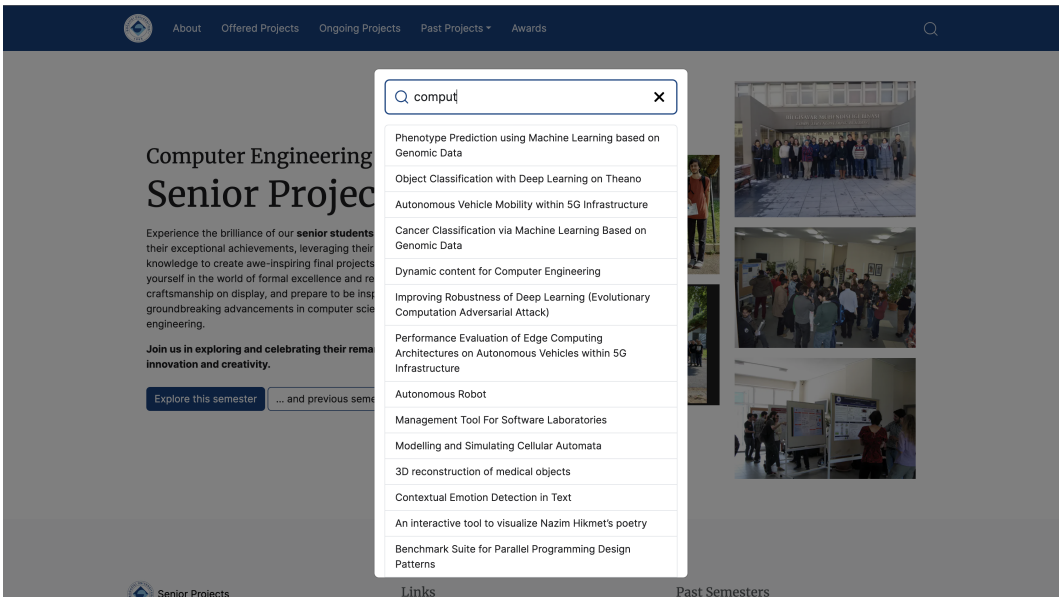

Figure 6.4. Search bar design

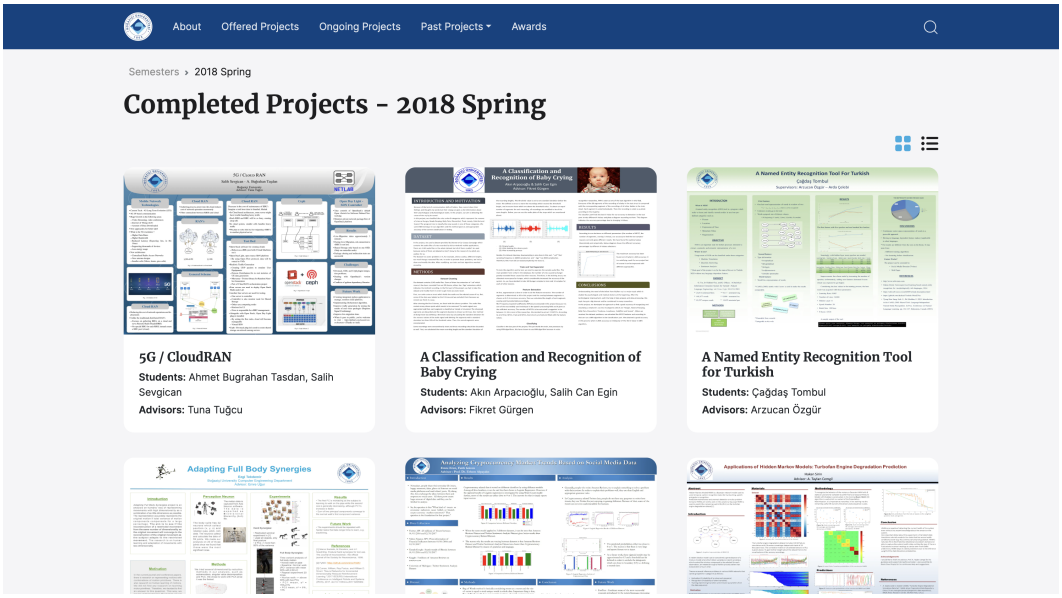

Figure 6.5. Projects page design

Figure 6.6. Completed project page design

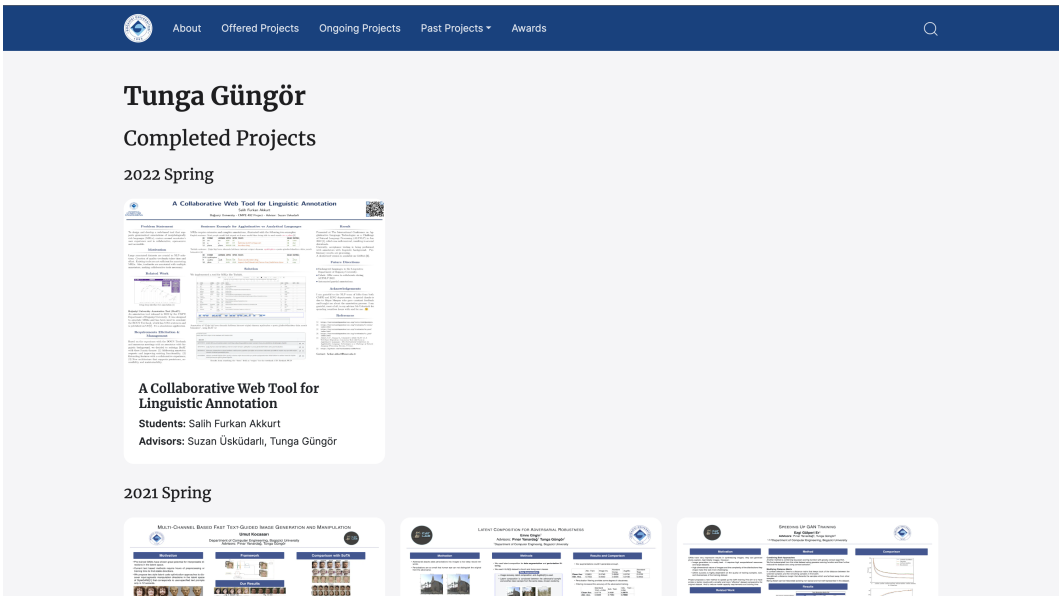

Figure 6.7. Advisor page design

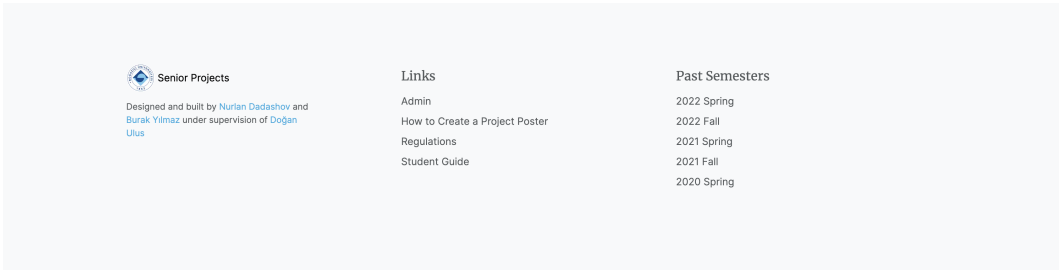

Figure 6.8. Footer design

7. IMPLEMENTATION AND TESTING

7.1. Implementation

We divided our project tasks into two categories: Frontend development and operational tasks. This parting allowed us to work in parallel and enabled us to progress on multiple parts simultaneously, increasing our productivity and efficiency.

The frontend development primarily focused on designing the website's user interface and ensuring that it is intuitive and easy to use. We also worked on developing the frontend logic for the project's search and filtering functionality. This involved working with the Hugo framework, bootstrap, and other related technologies.

For the operational tasks, on the other hand, we tried to focus on developing the remaining functionality of the web service, integrating it with other services, and ensuring that it is reliable and easy to maintain. This involved working with GH actions, Decap CMS, custom oauth server, and other related technologies.

7.2. Testing

For the website's SEO performance, we have run some loading benchmarks. There are several tools for this purpose. We have used Google's 'Lighthouse'[4] for performance testing. It basically showcases 4 different metrics:

- Performance
- Accessibility
- Best Practices
- SEO

[4]More details: `https://developer.chrome.com/docs/lighthouse/overview/`

Below are the results for the pages:

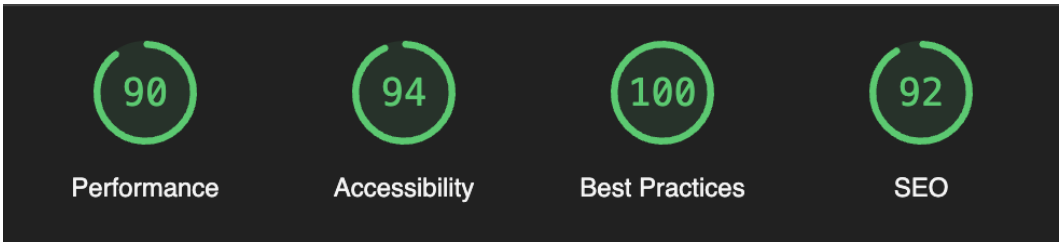

Figure 7.1. Main page performance results

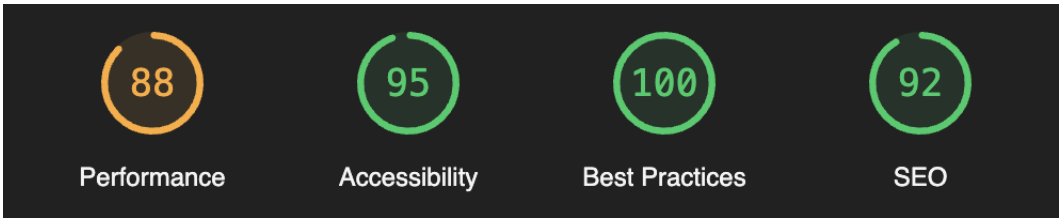

Figure 7.2. Offers page performance results

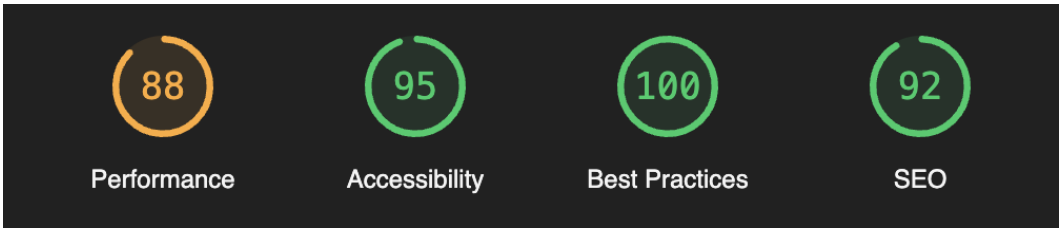

Figure 7.3. Projects page performance results

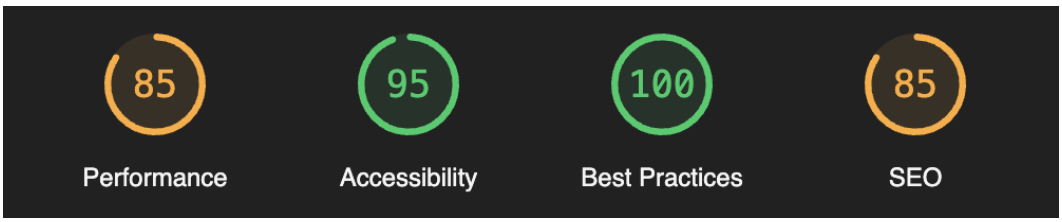

Figure 7.4. Completed project page performance results

7.3. Deployment

For deployment, we have used GitHub Actions to automate the process of building and deploying the website. We have also used GitHub Pages to host the website. The deployment process is triggered only on the main branch to ensure that only stable versions of the website are deployed. Additionally, we have deployed our custom oauth server, since that was a requirement from the github oauth application. We have published the docker image, so that it is easier to track the current application.

For Local Deployment

- `git clone git@github.com:bouncmpe/seniorprojects.git`
- `cd seniorprojects`
- `npm start`

8. RESULTS

We think the web service's deployment was successful and met the project's objectives. We think the web service's implementation, which complied with the project's criteria, was successful. We think that the new platform gives advisers an easy-to-use interface for creating and managing project offers.

Through continuous development, testing, and user feedback, we believe that we have achieved the following results:

- Developed a robust content management system (Decap CMS) that offers efficient project management capabilities for advisors.
- Designed an intuitive and accessible website interface for students to easily navigate and interact with project listings.
- Integrated user feedback to improve the overall user experience and address any usability issues.
- Ensured compatibility with various devices and browsers to accommodate a wide range of users.

9. CONCLUSION

In conclusion, the goal of our project was to construct a web service for senior projects at Bogazici University that would give advisors an effective platform for creating and managing project offers as well as a platform for students to browse and choose projects. We believe we have succeeded in achieving our objectives through careful planning, testing, and user feedback.

Throughout the project, we thought there were some crucial parts: Continuous improvement, seeking out feedback from the potential users and implementing the necessary adjustments. We were able to align the system with the changing needs and expectations of the users thanks to this iterative process, assuring its ongoing efficacy.

We have optimized workflows by using GitHub Actions into our development process, which requires less time and effort to make updates and adjustments. Fast reaction times and dependable performance have also been guaranteed via performance testing and optimization, satisfying the stated non-functional requirements.

Overall, we think that our project was effective in providing Bogazici University with a comprehensive and user-friendly web solution for senior projects. We hope that the Bogazici University senior project experience will continue to be supported and improved through this web service, which will be advantageous to both the academic and non-academic population.