Computer Science Lab Portfolio Project Report https://keeremaltun.github.io/portfolio/

Kerem Altun
Student ID: GH1043227

June 20, 2025

Abstract

This report details the development of my personal software engineering portfolio. It covers the design choices, implementation process using technologies like HTML, CSS, Python, Git, GitHub, and LaTeX, and the content included within the portfolio. The aim is to showcase my skills as a first-year Software Engineering student and reflect on my learning journey throughout this project.

1 Introduction

As a first-year Software Engineering student at Gisma University, this portfolio project serves as a comprehensive demonstration of my acquired skills and understanding in various aspects of software development. The project includes a static website, a simple Python application, and relevant documentation. My personal brand is evolving as I gain more experience, currently focusing on my interest in data visualization, web development, and open-source contributions. This report will walk through the conceptualization, development, and outcomes of this portfolio.

2 Portfolio Design and Structure

When I designed my portfolio, I really wanted it to be easy to understand and use. I also tried to make sure it showed who I'm becoming as a software engineer.

2.1 Website Structure (HTML/CSS)

The core of my portfolio is a static website built using HTML and CSS. I chose these technologies for their fundamental role in web development and to demonstrate a solid understanding of basic web principles.

- **HTML (`index.html`):** The website is structured into clear sections: a header for personal introduction, an "About Me" section to describe my academic background and interests, and a "Projects" section to showcase my work. A footer provides contact information. The layout prioritizes readability and a clean visual hierarchy.

- **CSS (`style.css`):** For styling, I opted for a clean and functional design. The goal was to create an interface that is visually appealing without being overly complex, typical for a student portfolio. I used a consistent color palette, legible fonts (like Arial), and managed spacing effectively to ensure a good user experience. Key design decisions included using a neutral background color to make the text content stand out, applying consistent padding and margins to sections for a clean layout, and implementing simple hover effects for links to clearly indicate interactivity.

- **Responsiveness:** While basic, I considered how the website would appear on different screen sizes by using simple media queries. This ensures that the portfolio remains accessible and navigable on both desktop and mobile devices.

2.2 Personal Branding

My portfolio serves as a digital representation of my personal brand. It highlights my academic journey, technical skills, and genuine interest in software engineering. The clean design and direct presentation of my projects aim to convey professionalism and a clear focus on my career aspirations in areas like data visualization and open-source contributions.

3 Implementation Details

The implementation phase involved utilizing several key tools and programming concepts.

3.1 Simple Python Project: Motivation App

My Python project is a ”Motivation App,” a straightforward console application designed to provide daily motivational messages.

- **Purpose:** This project allowed me to practice fundamental Python programming concepts, including basic input/output operations, list manipulation for storing quotes, and functions for organizing code.

- **Implementation:** The application functions by reading a collection of motivational quotes from a text file. Each time the app is run, it randomly selects one of these quotes and displays it in the console. This simple setup demonstrated my ability to handle basic file operations and implement random selection.

- **Learning Outcomes:** Through this project, I strengthened my understanding of modular programming principles, how to manage simple data structures, and the importance of clear variable naming for code readability. I also gained practical experience in debugging my own code.

3.2 Tools and Technologies Used

The development of this portfolio heavily relied on several essential tools and technologies:

- **HTML CSS:** As detailed, these were the foundational languages for the website.

- **Python:** Used for the ”Motivation App” project.

- **Git GitHub:** These were indispensable for version control and collaborative development (even for a solo project).
    - **Git:** Allowed me to track changes, revert to previous versions, and manage different development stages efficiently. I learned how to use commands like `git init`, `git add`, `git commit`, and `git status` effectively.
    - **GitHub:** Provided a remote repository for my project, enabling easy sharing and deployment via GitHub Pages. It also serves as a public record of my coding contributions.
- **LaTeX:** Used for generating this project report and my CV. LaTeX provided me with valuable experience in professional document preparation, demonstrating attention to detail and ability to use advanced typesetting tools.
- **VS Code:** My primary Integrated Development Environment (IDE) for writing code, managing files, and integrating with Git.
- **Overleaf:** An online LaTeX editor that greatly simplified the process of writing this report and my CV, allowing for real-time collaboration and easy compilation.

4 Challenges and Learning Outcomes

Throughout this portfolio project, I encountered various challenges that contributed significantly to my learning.

- **Version Control Management:** Initially, understanding Git’s branching and merging concepts, especially resolving conflicts, was challenging. For example, I encountered an issue when trying to add my ’motivation-app’ folder to the main ’portfolio’ repository. Since ’motivation-app’ had its own hidden ‘.git‘ folder from a previous initialization, Git misinterpreted it as a submodule, causing errors when I tried to track its files directly. This was confusing at first. Overcoming these issues taught me the importance of consistent `git add` and `git commit` practices, and how Git handles nested repositories.
- **Web Design Principles:** Creating a visually appealing and functional website required iterative design and constant refinement of CSS. I learned about the importance of the CSS Box Model, how to effectively use padding and margins for spacing, and the necessity of basic responsive design to ensure the site looks good on both desktop and mobile devices.
- **Project Structuring:** Organizing files and folders effectively within the repository to maintain clarity and scalability was also a key learning point. I learned to keep related files together (e.g., all CSS in ‘style.css‘, all HTML in ‘index.html‘) and to properly link them.
- **Documentation:** Preparing comprehensive documentation like this report and the ‘README.md‘ file reinforced the importance of clear communication in software projects. It taught me to explain my code and decisions clearly, which is a crucial skill for future collaboration.

Looking back, this whole project taught me a lot. It really boosted my hands-on skills in making software and keeping track of projects.

5 Conclusion

This software engineering portfolio stands as a testament to my progress as a first-year student. It successfully integrates a static website, a functional Python application, and professional documentation, all managed with industry-standard tools like Git and LaTeX. This whole experience wasn't just about using what I learned in class; it also really helped me get over real problems. It made me even more excited about software engineering and got me ready for what's next in my studies and career.

References

- Python Software Foundation. (n.d.). *The Python Tutorial*. Retrieved from `https://docs.python.org/3/tutorial/`

- Git SCM. (n.d.). *Pro Git Book*. Retrieved from `https://git-scm.com/book/en/v2`

- Overleaf. (n.d.). *Learn LaTeX in 30 minutes*. Retrieved from `https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes`

- W3Schools.com. (n.d.). *HTML Tutorial* and *CSS Tutorial*. Retrieved from `https://www.w3schools.com/`

- Gisma University. (2024). *Computer Science Lab Course Materials*.