Creating an Online Coding Assignment Auto-Assessment Platform

Will Murphy - 2766915M

December 12, 2024

1 Status report

1.1 Proposal

1.1.1 Motivation

In the early days of computers, proficient coding ability was a rare talent reserved mostly for cutting-edge software development and research purposes. However, today the landscape is vastly different. In 2022, it was reported that, in terms of salary, satisfaction, and availability, 10 of the top 20 'best' jobs in the UK on *GlassDoor* were related to computing or coding. With more and more industries leveraging digitalisation to optimise business processes, it is no surprise that the demand for coding skills has been steadily on the rise for a long time.

This increased demand for coding and therefore coding education has highlighted the need for more efficient ways to deliver and assess programming assignments. Current methods, which often rely on manual grading, are time consuming and inconsistent, especially when it comes to larger courses with more students. For teachers and lecturers, this manual approach can often prove too lengthy; this has led to the neglect of various vital parts of the learning process, such as personalised feedback.

This has not gone unnoticed by students, who have reported increased levels of stress surrounding their coding assignments due to the perceived lack of adequate support from staff. Through this, it is clear that there is a problem with the current approach of the coding education system and it is herein that the motivations for this project lie.

1.1.2 Aims

In light of the above motivations, this project proposes *AutoAssign*, an online platform to automate the creation, grading and feedback processes of assignments. The website aims to standardise, simplify and streamline all 3 of the aforementioned stages as follows:

Creation
Lecturers and teachers on the platform will be presented with a clear and straightforward interface to create assignments. This will include specifying a title and problem description, as well as optional extras such as enforcing a language to be used or providing a skeleton code template. This process will not only be easy for the creators of the assignments, but assignees will also benefit in that all their assignments will be presented in the same format.

Grading
Various grading tools will be provided to the creators of the assignments. In its simplest form, the platform will be able to match a submission's output to a predefined expected value. Optionally, grading can be taken further if a file containing unit tests is provided. Additionally, lecturers / teachers can opt to specify execution time and/or memory usage requirements. This flexibility ensures that submissions can be accurately assessed in a way that truly reflects the effectiveness of the solution.

Feedback
The platform will provide the functionality for lecturers/teachers to manually add feedback to submissions as they usually would. However, additional tools will also be available to automate the process. For example, if unit tests were provided, assignment creators can define feedback for individual unit tests that will be used should they fail. In addition, the platform will also facilitate the use of *ChatGPT* to generate personalised feedback. This tooling will significantly alleviate the pressure on lecturers/teachers of large classes while still providing students with sufficient feedback to help them learn.

In addition to streamlining these 3 standard stages of coding assignments, this platform also aims to introduce a new element: competition. Through the monitoring of average execution time and memory consumption, a leaderboard will be generated for each assignment. After an assignment's deadline has passed, all submissions will be ranked and visible to all assignees. Authors of submissions on the leaderboard will be able to opt to remain anonymous if they wish. The goal of this functionality is to inspire students to seek not just the most basic solution to a problem, but a more optimised one. This is a vital part of coding in the real world and should therefore be encouraged throughout all stages of the learning process.

1.2 Progress

- Researched and decided on technologies/frameworks to use for the project
- Initial project setup making a *GitHub* repository with all relevant project files
- Planned out the core functionality of the site (what pages it will have and what a typical user experience would involve)
- Outlined and implemented an initial design for the platform
- Backend integration with user account creation, signing in and out, creating and editing assignments, creating and editing groups
- Completed pages for: home, sign in, sign out, assignments, groups, profile
- Started work on pages for: specific assignments and groups
- Finished first draft for motivation and aims sections of the dissertation
- Finished first draft of related product review in the background section of the dissertation

1.3 Problems and risks

1.3.1 Problems

At the start, my main problem was defining the bounds of my project and where I wanted to go with it. Coming from the team project course last year, I was not expecting to have so much freedom to choose the direction. I addressed this issue by first focussing on the motivation and aims sections of the dissertation, which helped me to understand what kind of a tool might be of value and worthwhile to computer science students and educators alike.

My main blocker/problem that has held up progress later on in my project was the coursework and revision for my other courses. I underestimated the workload and this led to a string of weeks where this project saw no significant progress. Luckily I made a great start when the semester began, meaning that I am not left in a terrible position at the moment.

1.3.2 Risks

The only factor I currently consider to be a risk is the fact that I have not yet begun to implement the actual automated unit testing functionality into the platform. As this is the primary function of the site, it worries me as I have not done anything like it before. That said, before deciding on this direction for the project, I did some preliminary research on how such a feature works and it seemed well documented. Because of this, I am confident that it is something I will be able to do I am just not sure how long it will take.

1.4 Plan

As described above, I am slightly unsure how long certain parts of the project will take me but my rough plan can be outlined as follows:

- Christmas break: Finish first draft of the background chapter and begin implementation of automated unit testing
- January: Finish implementation of automated unit testing, implement manual submission feedback systems and finish first drafts of requirements analysis and design sections of dissertation
- February: Implement rankings of submissions and AI-assisted automatic feedback, finish first drafts of implementation and evaluation chapters of dissertation
- March: Revisit all first draft sections of the dissertation and write a conclusion as well as potentially implementing some of the feedback from the evaluation into the platform to improve it