Team B - Kattis Problem Practice Tool

Brendan Carr, Brooks Langley, Casey Stamper, Will Flanagan

Platform Selection

The Kattis Problem Practice Tool will be built on the UNC CS department server ada.cs.unc.edu. Because of this, we were very limited as a team in determining our platform selection. The beginnings of this project were set up by Professor John Majikes, who handed off his initial workings and ideas for us to build out.

The predetermined client-server architecture we will be following is NginX as the web server and uWSGI as the hosting server. NginX handles the basic HTTP features of the Kattis Problem Practice Tool and uWSGI connects the web server to the application. This architecture will be implemented through and through with Python.

HTTP client ↔ NginX ↔ uWSGI ↔ Python application

We had more freedom, however, when it came to implementing the functionality of the Kattis Problem Practice Tool. As the tool will have a front-end facing feature usable on the web, we have decided raw HTML and CSS will provide us the simplest and most flexible solution when implementing the front-end. We considered React to complete this task. However, it seemed we would not be able to leverage the real benefits of using a framework like React when the front-end we seeked was extremely minimal. Next, we looked into how we would host the plethora of Kattis coding problems and their automated tests. Per Professor Majikes suggestion, Docker proved to be a leading contender to accomplish such a task. Docker allows us to keep the bulky functionality of Kattis specific feature requirements away from the core client-server architecture. Plus, we all found interest as developers in learning a popular and powerful tool like Docker. Kubernetes came up in discussion as a possible alternative to Docker, but seemed more intensive to spin up and manage than Docker. As a result, we moved forward with Docker.