

# Mines CSCI Autograder 5.0

## Rethink, redesign, rebuild



### Client:

Christopher Painter-Wakefield (cpainter@mines.edu)

### Project:

Multiple courses at Mines use an autograding system for programming assignments and exams. The core function of the autograder is a web interface that allows students to submit C++, Python, or RISC-V assembly source code solving a focused problem. The response page provides feedback on whether or not various errors occurred (e.g., compilation error, floating point exception, etc.); if no errors, the page shows a graph of green and red bars representing tests which passed or failed (similar to the output of junit and its ilk). Autograder can be seen (and experienced) from on campus or through the VPN at <https://bartik.mines.edu/>.

The current system (Autograder 4.x) is written in Ruby on Rails, and is the result of years of work by the client, multiple prior field session teams, and others. This project proposes the creation of a new system on a new platform (Python/Flask for the back end and most likely Vue.js on the front end). There are several motivating factors for this move:

- I would like to make some significant, deep changes to the data model for the system. Design decisions that made sense when Autograder was being used for one course no longer make sense with multiple courses working in multiple programming languages.
- I would like to apply lessons learned over the last few years to improve the application design in a number of dimensions: accessibility, UX/UI, security, and application deployment to name a few.
- There are existing functions in the application that have become increasingly difficult to maintain, change, or extend on the existing platform (and which need to be fixed, changed, and extended!).

While Ruby on Rails is an excellent web framework in many respects, it has a steep learning curve and provides quite a lot of implicit behavior (aka “magic”). These factors make it very difficult for a non-expert to maintain and extend the application. The goal therefore is to move to a platform with much more explicit semantics. Flask is a Python framework for back-end web development meeting this requirement. Vue.js is a popular JavaScript client framework with a relatively small learning curve compared to competing frameworks. Please apply even if you do not have experience with these technologies but are eager to learn them! I will be highly available to help you get up to speed and get past roadblocks when they occur.

It is important to state that this is not a project to simply port Autograder to Flask/Vue. This is an opportunity to fundamentally redesign the system – while we can leverage *some* elements of the existing system, I want us to produce a truly new system that will better serve students and instructors.

### Technologies:

Python/Flask  
SQL  
HTML/CSS/JavaScript  
Vue.js

### Team size:

3-5

### Location:

Mines campus