Mines CSCI Autograder 4.0
Student-facing Feature Update

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++ or Python 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). The current system can be seen (and experienced) from on campus or through the VPN at https://flowers.mines.edu/.

The current system (Autograder 3.0) is the result of multiple years of work and two prior field sessions. In that time, while the Autograder has functioned well, instructors and students have noted certain areas that could be improved. The goals of the Auto-grader **4.0** project are to add a number of new features. These features fall into two categories: student-facing features and instructor-facing features. The student-facing goals include:

- Primary goals:
  - Eliminating separate test, submit, and save functions in favor of a combined submit function
  - Add capability to store a submission history, with an interface to revert to any prior submission
  - Store scores associated with each submission, and improve the display of scores and grades to students
  - Autosave code on edit (similar to e.g., Google docs) – students should never lose edits when closing the browser or changing pages
- Secondary goals:
  - Various minor improvements in the user interface
- Stretch goals (in collaboration with project team working on instructor-facing features)
  - Port to latest versions of Ruby, g++, and possibly Rails
  - Improve Python support

In addition to the above, the team will be asked to implement a continuous integration / deployment process for the Autograder project. If the instructor-facing features are also being worked (by a separate project team), this process will need to be developed in coordination with the other team, and the teams will need to coordinate on git branch management.

Technologies:	**Team size:**	**Location:**
Ruby	2-4	Mines campus
Rails		
HTML/CSS/Javascript		
SQL		