



# Developing an Automatic Assignment Grader for Computer Science Classes

Christian Sumares and Alexandre Stoykov (mentor)  
Gavilan College, Department of Computer Science, Gilroy, CA 95020



## INTRODUCTION

Beginner level computer science classes often have students create simple programs which produce very specific output. However, because students must wait for individual feedback on these assignments, the process could be greatly expedited if feedback or corrections were given immediately. The goal of this project is to develop a web application for Gavilan College that automatically grades and provides feedback for student-created programs written in the C++ language by testing them against a series of inputs and comparing their outputs with those expected by the instructor of the class.

## METHODS

Another student had worked on this project before me, using the XAMPP web server solution stack. I tried to use this at first, however some of the more complex parts of the project would have made this difficult. Instead, I switched to using the Django web framework, written in Python, and several other tools on an Ubuntu server. Using a framework allows better handling of more complex features such as user sessions and a template system, in addition to easier database access. I took advantage of Django's "Model-View-Template" structure to dynamically display feedback information to students without having to store any unnecessary files on the server. The grading itself is fairly straightforward. Each character in the program's output is compared with the equivalent character in the expected output for the input it was given. If they do not match, the index of that character is noted in the feedback.



## RESULTS

After nine weeks of work, the application is functionally complete. Students can log in using their Gavilan ID number and submit a program for any assignment open to them, which is tested and graded accordingly. The instructor can perform administrative tasks such as managing courses or assignments and creating student accounts. The screenshots below show the site in its current form, along with snippets of its code.

An example of the jQuery plugin "DataTables" which I used to easily list courses, assignments, etc.

Course	Assignments
<a href="#">C++ Programming I</a>	5
<a href="#">CSIS 1</a>	0
<a href="#">CSIS 26</a>	1
<a href="#">CSIS 27</a>	0

### Correct Output vs. Incorrect Output

```
Test 1:
INPUT:
200 650

CORRECT OUTPUT:
Please enter num1:
Please enter num2:
The sum of 200+650 = 850

ACTUAL OUTPUT:
Please enter num1:
Please enter num2:
The sum of 200+650 = 850
```

```
Test 1:
INPUT:
200 650

CORRECT OUTPUT:
Please enter num1:
Please enter num2:
The sum of 200+650 = 850

ACTUAL OUTPUT:
Please enter num1:
Please enter num2:
The sum of 179+650 = 829
```

### The main function responsible for testing students' programs

```
232 def test_student_program(exec_path, assg):
233     from os import remove
234     tests = assg.assgtestcase_set.all()
235     ret = []
236     for test in tests:
237         program = run(exec_path, stdout=PIPE, input=test.inputstr, encoding='ascii')
238         feedback = dev_util_feedback(test.inputstr, program.stdout, test.correct_output)
239         for i in range(0, len(feedback.actual)):
240             if i >= len(feedback.correct) or feedback.actual[i] != feedback.correct[i]:
241                 feedback.addIndex(i)
242         feedback.group()
243         feedback.correct = correctAsHTML(feedback.correct)
244         ret.append(feedback)
245     remove(exec_path)
246     return ret
```

## CONCLUSIONS

In summary, this application eliminates the need to manually check that every program works. It alerts students of potential mistakes they may have made, and saves time for both students and their instructor as a result. I believe it will be an invaluable asset for computer science students at Gavilan.

I created this web application with no prior experience with Django, Ubuntu, or web development of any kind, only knowing Python, so it was very much a learning process for me. I had to work through many issues to get it to this point. After the first stable version is online, I plan to add many improvements such as a better user interface, tighter security, support for other programming languages, and more types of assignments.

## REFERENCES

Server-side web frameworks. (2019, May 7). Retrieved from [https://developer.mozilla.org/en-US/docs/Learn/Server-side/First\\_steps/Web\\_frameworks](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks)

Django Documentation. (n.d.). Retrieved from <https://docs.djangoproject.com/en/2.2/>

Many other sources were used as well, including the online documentation for Ubuntu, Django, etc. as well as countless answers from Stack Overflow.

## ACKNOWLEDGEMENTS

Thank you to Gavilan College and its affiliation with the Science Technology Engineering and Mathematics (STEM) program for funding this internship. Additional thanks goes to my mentor, Alex Stoykov, for selecting me for this internship.