Project Narrative
CMS 340 – Software Engineering
Noah Harvey, Jenny Goldsher, Deandra Martin, Hiro Sato
11/1/2020

Virtual Programming Tutor Application

Students in CMS 120 have come to tutoring sessions with consistent issues. For instance, students write algorithms with incorrect data types of the variables they use. This type of issue comes from how Python is formatted (not making programmers declare data types on assignment/declaration), and omitting details from programmers. Although the format of Python is oriented towards making programming easier, beginners do not inherently learn necessary concepts because the language covers up many aspects of programming complexity.

Because CMS 120 is an introductory course, students are not expected to have prior knowledge of these concepts, however, it is good practice to teach students to code defensively and understand how the code they have written works. Since CMS 120 students are coding in python as beginners, allowing them to program with more freedom than many other high-level languages, they may face more difficulty when they get to upper level CS classes if they lack a proper understanding of key programming concepts. This emphasizes the importance of having beginner students learn effective coding practices early on.

To combat this problem, there is a need for a web application that is able to read through student's code and both give feedback and ask questions. The program won't format the code for students but it will ask related questions and challenge them to further their understanding of python. This would be a comprehensive way for CMS 120 students to reflect their understanding from class while they are programming.

Requirements Elicitation:
To formulate these requirements we gathered information from CS tutors and reviewed other related applications (Slack, CodingBat, LeetCode).

- The system should be accessible to all students via the internet (Web application)
- The system should allow users to enter a snippet of Python code
- It should be clear to user where they should insert their code snippet
- The application should be in the form of a chatbot where user can easily interact
- The system should generate answers but not show the answer until students at least try to answer those questions on their own first.
- The system should generate set of questions based on the Python code that user feeds in
- The types of questions that the system(chatbot) creates can be classified into;
    - data type questions,
    - basic syntax, about functions
- The system does not necessarily have to display all the possible questions, the number of questions can be selected or randomly generated (Default is 7)
- The system is a parser, not a compiler. This program should not execute the Python file and provide its output.
- The system is aimed to review student's understanding of Python in an interactive way, so this system should not change the original python file at all.
- The system should provide a link to a page that discusses common error messages and their meaning

Flow of program from User's perspective:

1. Open Web application
2. Insert code snippet
3. System parses the file and comes up with the set of questions and generate answer for each question
4. Display the questions to user
5. User types in their answers
6. Display whether they got it or not, if they got it wrong, - or their answer versus the computer's answer
7. Proceed to the next question
8. Repeat the 4,5,6,7 for the number of questions.

Use cases are:

- Process Python Code
- Generate Questions
- Validate user's answer