Development and Evaluation of an AI-Enhanced Python Programming Education System

Eric Zabala and Husnu S. Narman
Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, USA
Email: zabala@marshall.edu, narman@marshall.edu

***Abstract*—The integration of Artificial Intelligence (AI) in education has shown promising potential to enhance learning experiences and provide personalized assistance to students. However, existing AI-based educational tools often exhibit limitations, including inconsistent feedback, limited adaptability to diverse learning needs, and difficulties in delivering real-time and accurate assessments. These limitations restrict the full effectiveness of AI in supporting students' educational journeys.**

In this paper, we present the development of an AI-based Python programming education system that integrates a Chatbot for student assistance, an automated grading system for feedback, and an entrance exam feature that suggests chapters and sections for review. The Chatbot and grading system employs GPT-3.5 Turbo, leveraging its extensive knowledge base, cost-effectiveness, time efficiency, and adaptability to various programming queries, which enhances student engagement. Although the system underwent interactive testing and continuous improvements, the development process encountered difficulties in maintaining consistent AI feedback and enhancing real-time performance. Users can take quizzes, receive grades, obtain personalized feedback, and get course recommendations. The grading system achieved 28/30 consistency with its output while the course recommendation system achieved 26/30 consistency with its outputs. The results indicate that while the AI-based system aids in learning programming by providing instant feedback and recommendations for improvement, its effectiveness is limited. This project underscores the potential of AI to enhance educational tools and sets the stage for further advancements in AI-driven education systems.

***Keywords*—Artificial Intelligence, Python, Education, Advanced Learning Technologies**

I. Introduction

Artificial Intelligence (AI) is a frequent topic of discussion in modern times. Some of the Large Language AI Models (LLMs) such as ChatGPT have shown their astounding capability by passing the German State Examination in Medicine, passing the Bar Exam, and achieving a passing score on a computer science exam [1]–[3]. These feats demonstrate LLMs' versatility and potential in various fields, including education. Building on these capabilities, this paper presents the development and evaluation of an automated Python education system that we created, leveraging OpenAI's APIs to enhance learning through AI-driven feedback and personalized assistance. There have been many other programs created to teach Python to beginners using games and similar interactive ways [4]–[6]. Some of the education websites such as Tynker [4] have been specifically designed for K-12 students. There are interactive websites such as Codecademy [5] that have compilers built in so that the learners can have an opportunity for practice. There are also other educational websites such as Khan Academy [6] that prepare students for entrance exams and help guide students in a direction of learning. Currently, none of these have quizzes where students will get personalized feedback from, or have a way to ask questions related to the course they are taking before they registered for courses.

In this paper, we aim to develop a Python learning platform suitable for all ages. The platform will interact with users before they begin courses to accurately assess their skill level and recommend chapters based on their experience.

The key *contributions* of this paper are: (i) implement an AI-driven feedback system that interacts with users in real-time, providing personalized suggestions and improvements based on their input, (ii) develop a monitoring system to evaluate the AI API's performance over time, focusing on consistency and reliability, and (iii) create a feedback loop that continuously refines the AI's algorithms based on user interactions and feedback. The results indicate that although the AI-based system helps in learning programming by offering instant feedback and suggestions, its overall effectiveness is limited.

The rest of the paper is organized as follows. In Section II, we have discuss how the system is developed. In Section III, the implementations of the application as well as demonstrating the system have been explained. Section V compares the application to other previous works that teach the same subject. Section IV discusses the results, and finally Section VI contains the conclusion and remarks for future works.

II. Development

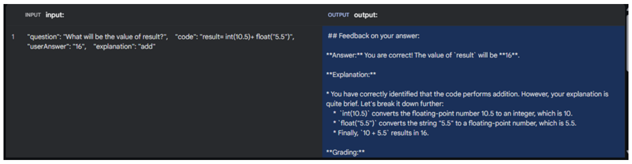
In this section, we will explain the details of the developed platform in terms of utilized tools, contents, and design.

A. Tools Utilized

To ensure the system is beginner-friendly, we decided that a web browser-based platform would be the most accessible. The system utilizes HTML5 for display, JavaScript to enhance user-friendliness, and Python for the back-end operations and AI API connections. After evaluating various options, we concluded that integrating the OpenAI API would be the most suitable approach for the project. Fig. 1 and Fig. 2 shows a test that is conducted on both OpenAI's API

979-8-3315-4090-6/24/$31.00 ©2024 IEEE

and Google's API using the same prompt. As demonstrated in Fig. 1 and Fig. 2, OpenAI responses are more appropriate for our purpose. For hosting the system, Amazon Web Services (AWS) is decided to be used as this aligns with current industry standards.



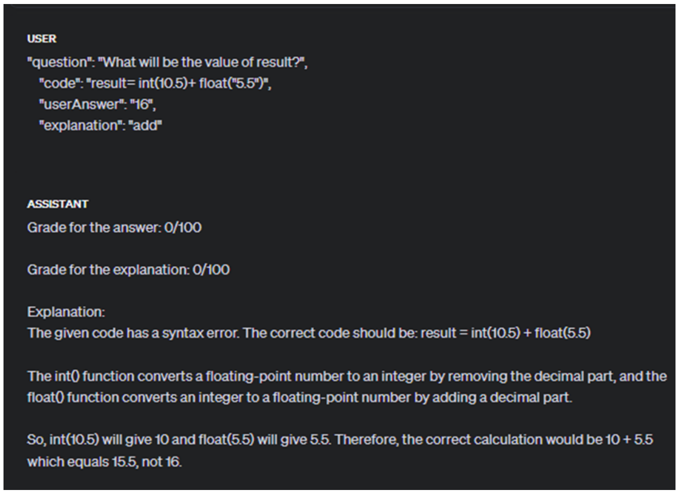
INPUT input:

```
"question": "What will be the value of result?",    "code": "result= int(10.5)+ float("5.5")",
"userAnswer": "16",    "explanation": "add"
```

OUTPUT output:

```
## Feedback on your answer:

**Answer:** You are correct! The value of `result` will be **16**.

**Explanation:**

* You have correctly identified that the code performs addition. However, your explanation is
quite brief. Let's break it down further:
  * `int(10.5)` converts the floating-point number 10.5 to an integer, which is 10.
  * `float("5.5")` converts the string "5.5" to a floating-point number, which is 5.5.
  * Finally, `10 + 5.5` results in 16.

**Grading:**
```

Fig. 1: Example of a test done for API suitability on Google's API.



```
USER
"question": "What will be the value of result?",
  "code": "result= int(10.5)+ float("5.5")",
  "userAnswer": "16",
  "explanation": "add"


ASSISTANT
Grade for the answer: 0/100

Grade for the explanation: 0/100

Explanation:
The given code has a syntax error. The correct code should be: result = int(10.5) + float(5.5)

The int() function converts a floating-point number to an integer by removing the decimal part, and the
float() function converts an integer to a floating-point number by adding a decimal part.

So, int(10.5) will give 10 and float(5.5) will give 5.5. Therefore, the correct calculation would be 10 + 5.5
which equals 15.5, not 16.
```

Fig. 2: A suitability test on OpenAI's API.

B. Content Overview

The application is structured into six chapters, with chapters one and two fully implemented. Each chapter is subdivided into sections to enhance the course's navigability and assist students in comprehending difficult concepts.

In chapter one, the first section details the installation of Python and an Integrated Development Environment (IDE) on a Windows system, as our resources were limited to Windows operating systems. Following this, two sections introduce the fundamental aspects of Python. Chapter two explores the basic types of variables. The initial section provides an overview of each type, while the subsequent sections delve deeper into their specifics. This chapter also includes a quiz designed to assess the learner's understanding of the material covered in chapters one and two.

Moreover, the system incorporates an entrance exam where users can submit answers and explanations. We permit blank responses, acknowledging that it is acceptable to lack knowledge in certain areas. The exam encompasses a range of topics extending to Chapter 6, which addresses lists, list operations, and list functions. Although chapters 3 through 6 have not been added due to time constraints, the entrance exam still references these chapters and recommends them for review to ensure a comprehensive evaluation.

C. Design Challenges

The primary objective of the system is to implement a hands-off approach to teaching, allowing it to autonomously grade, correct misconceptions, answer questions, and help users identify their knowledge gaps. This approach aimed to prevent the unnecessary reteaching of concepts that students already understood. To achieve this, we needed to identify a system capable of performing these tasks efficiently. After thorough testing, we evaluated both OpenAI's and Google's APIs. Ultimately, we choose OpenAI's APIs due to their ability to consistently format responses, which is a critical requirement for our system (see Section II-A for prompt testing.).

One of the initial challenges we encounter is ensuring that the chapter quizzes produced standardized grades. To address this, we collected data from users, graded their responses, and provided personalized feedback. Additionally, we established a grading scale for each question, which helped in correcting any inconsistencies in the grading process. Another significant challenge is achieving uniform outputs from the Entrance Exam. This required numerous iterations and adjustments to the prompt. Through persistent refinement, we are able to achieve reliable and consistent results from the Entrance Exam, ensuring that it accurately assessed the users' knowledge and understanding.

III. Implementation

In this section, we provide a detailed explanation of each component of the application [7] and outline specific source code specifications [8].

A. Home Page

Fig. 3 shows the home page of the application. Users can take the entrance exam by logging with their codes, select the chapter/section want to review, or take the chapter quiz from drop-down menu of each chapter.

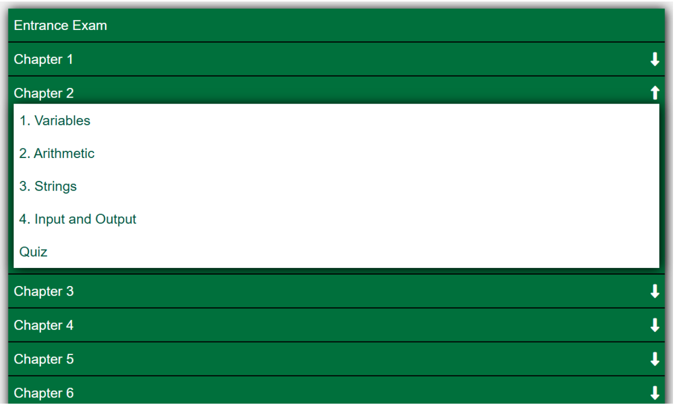
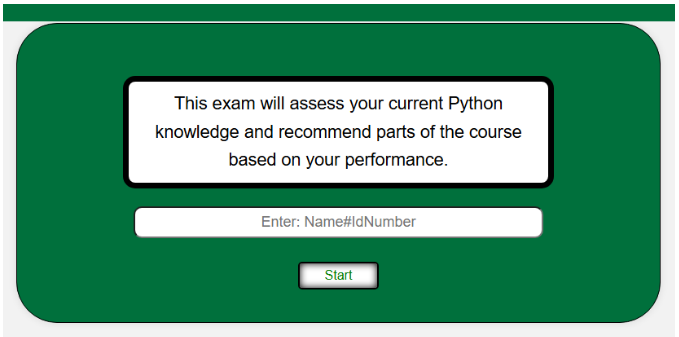


Fig. 3: Home page for the application.

B. Entrance Exam

Fig. 4 shows the login screen for the Entrance Exam. This is where the user types in their ID an example of an ID would be John#1234. This is so the system can file each user's

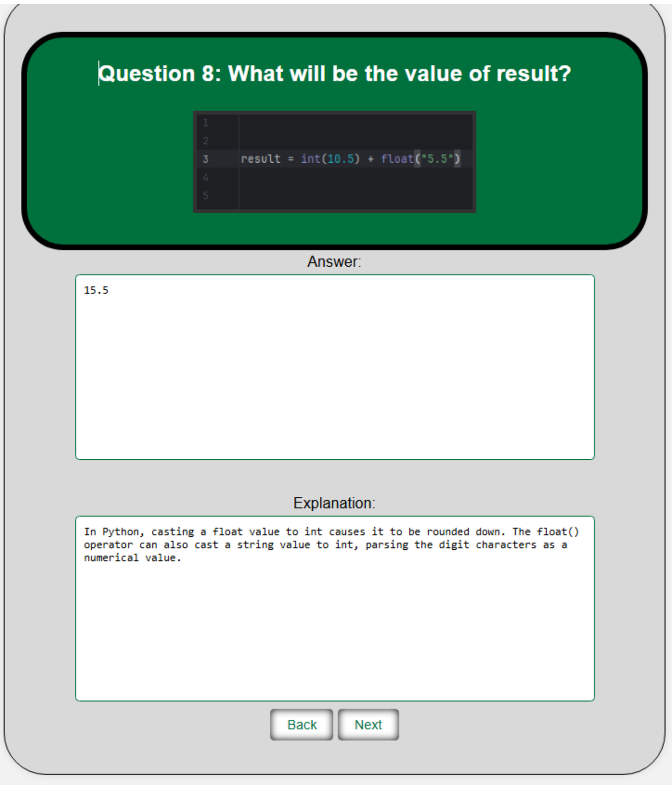


This exam will assess your current Python knowledge and recommend parts of the course based on your performance.

Fig. 4: Log-in screen for the Entrance Exam.

inputs into a JSON file. Then, the JSON file is then given to the API for grading and course review recommendations.

Fig. 5 shows one of the questions from the exam that is used to test the students' knowledge of Python. The system requires the user to give an answer and an explanation to get the question correct. This helps prevent guessing and a false belief of understanding [9].



Question 8: What will be the value of result?

```
result = int(10.5) + float("5.5")
```

Answer:

15.5

Explanation:

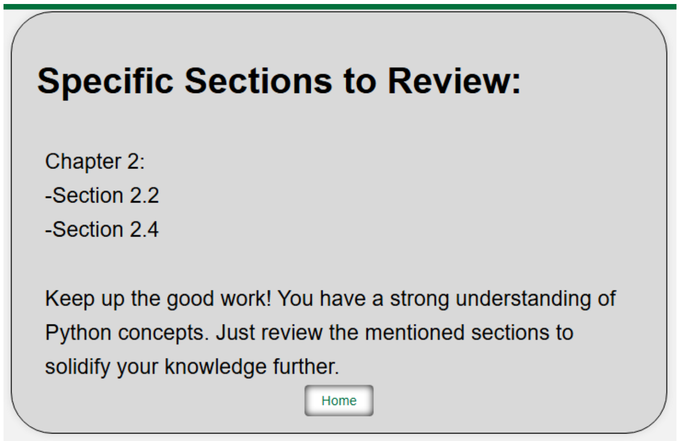
In Python, casting a float value to int causes it to be rounded down. The float() operator can also cast a string value to int, parsing the digit characters as a numerical value.

Fig. 5: An example of a question within the entrance exam.

Lastly, Fig. 6 displays a review recommendation for a student who got most of the answers and explanations correct on the exam.

C. Course Content

Fig. 7 shows how one of the lessons looks like. The system features many aspects to assist with engagement. Student engagement is critical for successful learning [10]. This includes an embedded compiler from Trinket.io using Python [11]. This allows the student to see how the code they are learning about works as well as gives the student the opportunity to test other ideas on the compiler without needing to download and install anything.

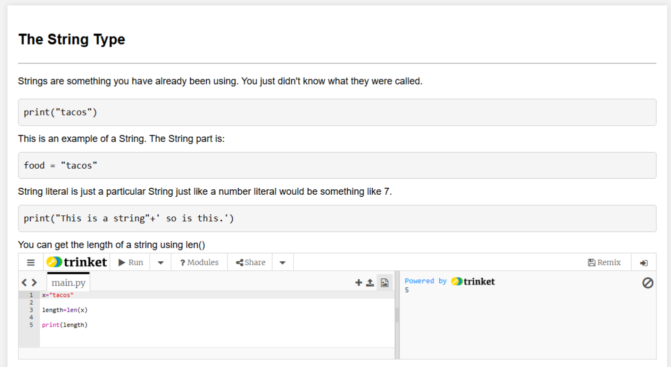


Specific Sections to Review:

Chapter 2:
-Section 2.2
-Section 2.4

Keep up the good work! You have a strong understanding of Python concepts. Just review the mentioned sections to solidify your knowledge further.

Fig. 6: Output from Entrance Exam, showing the chapter and section recommendations.



The String Type

Strings are something you have already been using. You just didn't know what they were called.

```
print("tacos")
```

This is an example of a String. The String part is:

```
food = "tacos"
```

String literal is just a particular String just like a number literal would be something like 7.

```
print("This is a string"+' so is this.')
```

You can get the length of a string using len()

```
x="tacos"

length=len(x)

print(length)
```

```
5
```

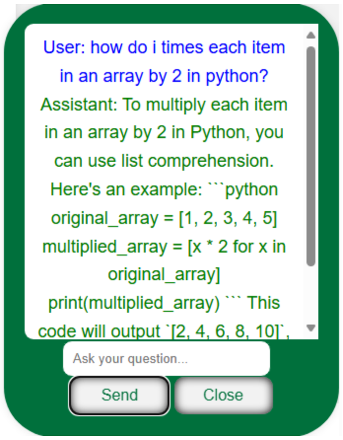
Fig. 7: An example of a common layout of a lesson within the course.

D. Chatbot

Fig. 8 shows how the chatbot is integrated into the course content for students who have Python-related questions. It can provide explanations and guidance within the scope of the course. The learners or newcomers struggling to formulate questions could potentially receive helpful information by simply expressing their confusion or area of difficulty to the chatbot. The system does not answer non-Python-related questions, instead, the system will reply with "Sorry, That is outside the scope of this class."

E. Chapter Quiz

Quizzes can be a powerful tool to assist students learn [12]. Feedback also has a powerful influence on learning as long as the student can understand the feedback [13]. Therefore, in the quizzes, we aim to provide consistent feedback for each

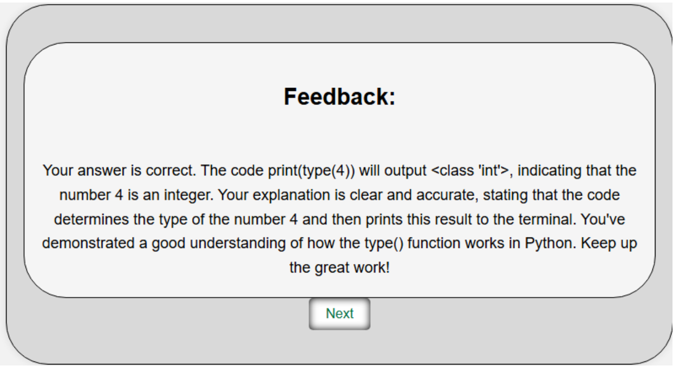


User: how do i times each item in an array by 2 in python?

Assistant: To multiply each item in an array by 2 in Python, you can use list comprehension. Here's an example: ```python original_array = [1, 2, 3, 4, 5] multiplied_array = [x * 2 for x in original_array] print(multiplied_array) ``` This code will output `[2, 4, 6, 8, 10]`,

Fig. 8: The Chatbot answering a question about an array in Python.

question according to users' answers and explanations. One of the main issues we ran into was having the system have a consistent formatted output. We tried having the system take in the questions, answers, and explanations giving them a grade and feedback. The system gives constructive feedback even when a student gives a correct answer as shown in Fig. 9.



Feedback:

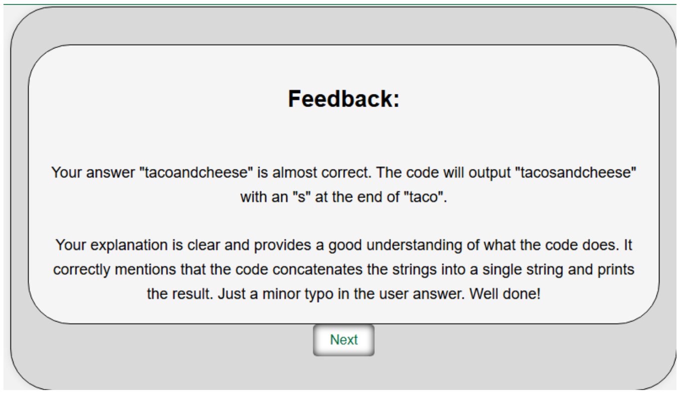
Your answer is correct. The code print(type(4)) will output <class 'int'>, indicating that the number 4 is an integer. Your explanation is clear and accurate, stating that the code determines the type of the number 4 and then prints this result to the terminal. You've demonstrated a good understanding of how the type() function works in Python. Keep up the great work!

Fig. 9: Feedback to a student for a correct answer.

Fig. 10 shows an example of a student getting an answer mostly correct. The system can also understand when a student gives an answer that answers what the code does but does not answer the question correctly (see Fig. 11). The feedback helps students get a deeper understanding of the information they are trying to learn [14]. The feedback demonstrates that the application assesses the degree of correctness in student answers, rather than simply a binary correct/incorrect evaluation.

F. User Feedback

The system includes a comprehensive feedback section where students can provide detailed input to enhance the system over time or correct any errors they encounter. This feedback mechanism is crucial as it allows for continuous improvement and refinement of the system. By gathering insights from users, the system can adapt to better meet their needs, address any issues promptly, and ensure a more effective and user-friendly experience. Encouraging students to share their thoughts not only helps in identifying areas for improvement but also fosters a collaborative environment where users feel valued and heard [14].

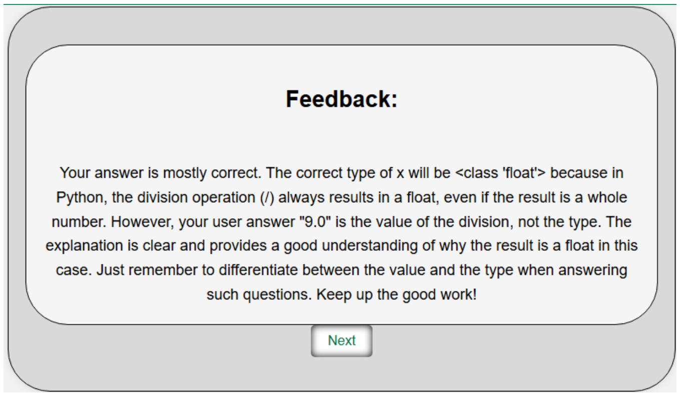


Feedback:

Your answer "tacoandcheese" is almost correct. The code will output "tacosandcheese" with an "s" at the end of "taco".

Your explanation is clear and provides a good understanding of what the code does. It correctly mentions that the code concatenates the strings into a single string and prints the result. Just a minor typo in the user answer. Well done!

Fig. 10: Feedback to a student for a mostly correct answer.



Feedback:

Your answer is mostly correct. The correct type of x will be <class 'float'> because in Python, the division operation (/) always results in a float, even if the result is a whole number. However, your user answer "9.0" is the value of the division, not the type. The explanation is clear and provides a good understanding of why the result is a float in this case. Just remember to differentiate between the value and the type when answering such questions. Keep up the good work!

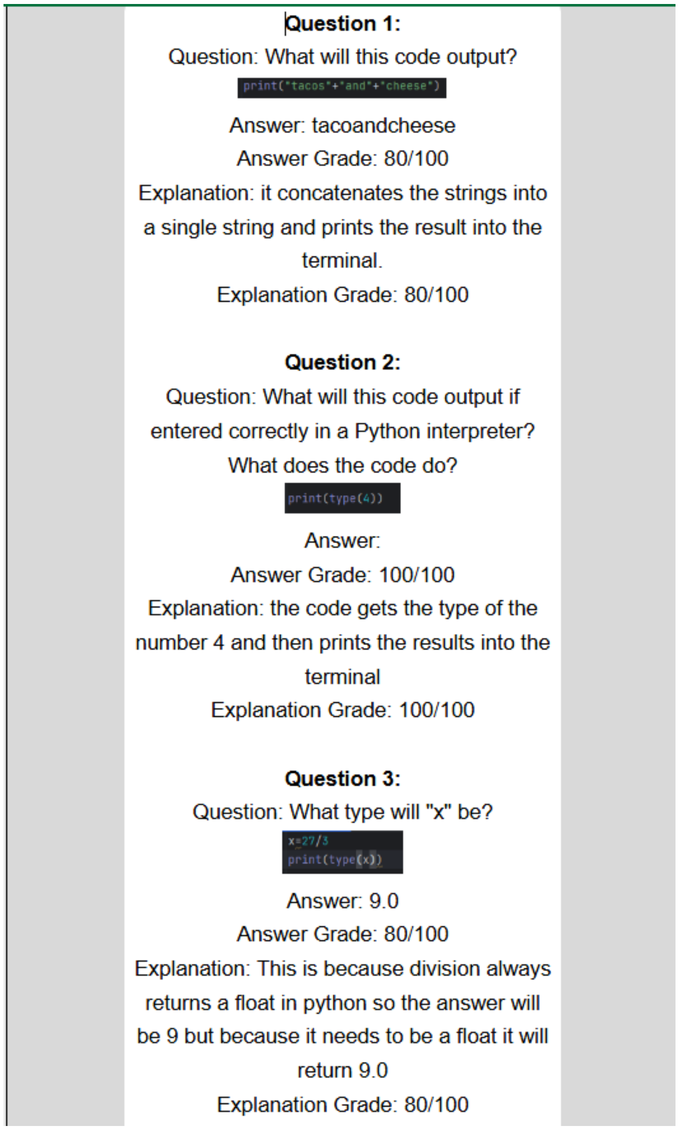
Fig. 11: Example of a student giving an incorrect answer to the question but demonstrating they still know what the code is doing.

IV. Results

The training data for the system was meticulously collected from 18 individuals aged between 20 and 68 years. This diverse group provided a comprehensive dataset consisting of 30 questions, identical to those featured in the entrance exam. Initially, the first five questions were utilized to train the system, which was subsequently integrated into the quiz module. As edge cases emerged, the system was refined to address these anomalies.

Following these refinements, the quiz underwent extensive testing. Each question was tested 30 times using uniform answers and explanations. Questions 1, 2, 4, and 5 consistently yielded the same grades, demonstrating the system's reliability. However, question 1 produced a different answer on two occasions, resulting in a consistency rate of 28 out of 30 instances. Despite these minor inconsistencies, the overall feedback remained stable, and the core meaning of the work is preserved. Figs. 12 and 13 illustrate the consistent output from the system.

For the entrance exam, the system was executed 30 times. This number was strategically chosen due to the associated



Question 1:

Question: What will this code output?

```
print("tacos"+"and"+"cheese")
```

Answer: tacoandcheese

Answer Grade: 80/100

Explanation: it concatenates the strings into a single string and prints the result into the terminal.

Explanation Grade: 80/100

Question 2:

Question: What will this code output if entered correctly in a Python interpreter? What does the code do?

```
print(type(4))
```

Answer:

Answer Grade: 100/100

Explanation: the code gets the type of the number 4 and then prints the results into the terminal

Explanation Grade: 100/100

Question 3:

Question: What type will "x" be?

```
x=27/3
print(type(x))
```

Answer: 9.0

Answer Grade: 80/100

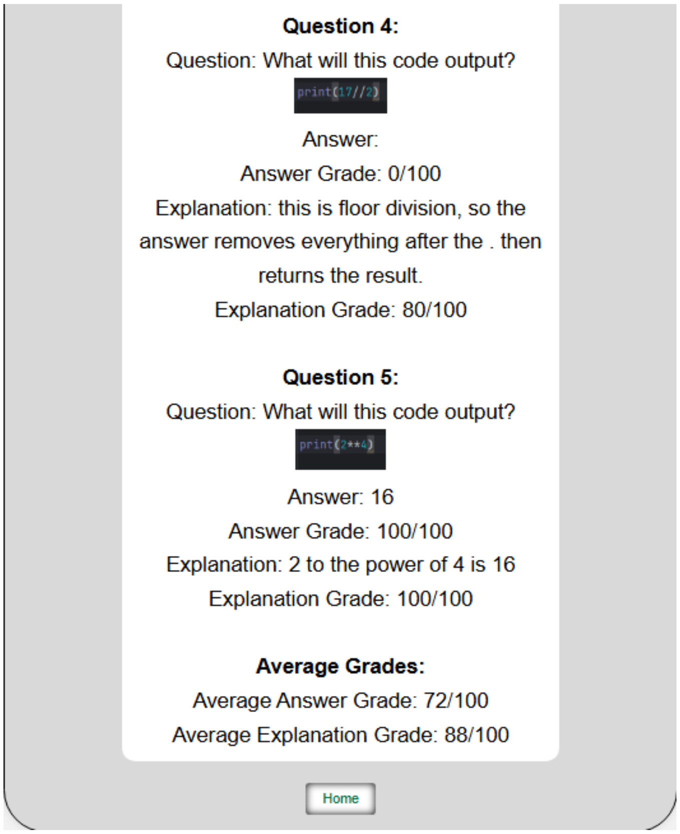
Explanation: This is because division always returns a float in python so the answer will be 9 but because it needs to be a float it will return 9.0

Explanation Grade: 80/100

Fig. 12: Shows the top portion of the grading screen after the quiz.

costs and token usage of each test. The initial results indicated that the system provided standardized results 20 out of 30 times. After implementing further improvements, the consistency rate significantly increased to 26 out of 30 instances, showcasing the system's enhanced reliability.

The system's performance is also evaluated in terms of execution time. The fastest execution was recorded at 3.95 seconds, while the slowest took 10.36 seconds, with an average execution time of 5.15 seconds per run. Notably, the system made the same mistake twice by adding section 4.6 and incorrectly marking question 29. In two other instances, no questions were marked wrong, and chapter 2.2 was recommended. The first instance removed chapter 2.4, indicating all questions were correct, while the second instance replaced chapter 2.4 with chapter 2.3 but still indicated the student mostly passed the exam.



Question 4:

Question: What will this code output?

```
print(17//2)
```

Answer:

Answer Grade: 0/100

Explanation: this is floor division, so the answer removes everything after the . then returns the result.

Explanation Grade: 80/100

Question 5:

Question: What will this code output?

```
print(2**4)
```

Answer: 16

Answer Grade: 100/100

Explanation: 2 to the power of 4 is 16

Explanation Grade: 100/100

Average Grades:

Average Answer Grade: 72/100

Average Explanation Grade: 88/100

Fig. 13: Shows the bottom portion of the grading screen after the quiz.

These results underscore the system's ability to adapt and improve over time. The feedback mechanism played a crucial role in identifying and addressing errors, leading to a more robust and reliable application. Continuous testing and refinement have ensured that the system not only meets but exceeds the expectations of its users, providing a uniform and efficient learning experience.

V. Related works and Differences

There exists a substantial body of work examining the consistency of AI responses. In [15], the authors concluded that ChatGPT exhibited inconsistencies in its outputs. Our findings support this conclusion; however, we observed that the application of APIs and prompt engineering significantly enhanced the consistency of the results.

Furthermore, a study conducted in early 2023 asserted that AI systems should be regarded solely as tools and lack the merit to replace educators [16]. Our research aligns with this perspective, and it indicates that while current AI systems are not yet capable of supplanting human educators, the continuous evolution of APIs holds considerable promise for future applications.

The integration of advanced APIs and refined prompt engineering techniques has demonstrated potential to mitigate the inconsistencies inherent in AI-generated responses. This approach not only improves the reliability of the outputs but also enhances the overall user experience. As these technologies evolve, they are likely to play an increasingly pivotal role in educational settings, augmenting traditional teaching methods rather than replacing them.

A. Khan Academy

Khan Academy offers a structured approach to learning, with courses divided into smaller sections and supplemented by instructional videos and interactive components that test student knowledge. However, while Khan Academy provides these interactive elements, it lacks integrated exams for beginner Python lessons and does not offer personalized feedback during learning. In contrast, the Python education system we developed not only includes quizzes at the end of each lesson starting from chapter two but also features embedded compilers within the lessons, allowing students to experiment with code in real-time. Additionally, unlike Khan Academy, our system incorporates a built-in chatbot that enables students to ask Python-related questions directly within the course, providing instant, tailored assistance that adapts to their individual learning needs [6].

B. Tynker

Tynker provides an engaging learning environment with interactive Python lessons and games designed for younger audiences. While it offers a subject-based structure that is easy to follow, it lacks real-time support for students who encounter difficulties during lessons. Additionally, progress in some courses may be hindered by technical issues, such as non-functional games, and the platform requires a subscription to access certain features. In contrast, our Python education system is designed to be fully accessible and user-friendly, offering real-time support through an integrated Chatbot that answers Python-related queries as they arise. Furthermore, the system includes quizzes that provide immediate feedback, helping students to understand and correct their mistakes instantly. The addition of an entrance exam in our system also allows students to identify areas where they need further review, offering a more personalized and adaptable learning experience compared to Tynker [4].

C. Codecademy

Codecademy offers a variety of interactive lessons and an AI chatbot that can assist students with their learning; however, access to these features often comes at a cost, and the number of questions users can ask the chatbot is limited without a subscription. Additionally, although Codecademy offers interactive lessons with built-in code testing, many of their Python courses and the accompanying quizzes require payment. In contrast, our Python education system is entirely free to use, providing students with unlimited access to its features, including the Chatbot, which can answer up to five questions at a time before needing a reset—without any additional costs. Our system also distinguishes itself by offering comprehensive quizzes at the end of each lesson, ensuring that students can reinforce their knowledge and receive personalized feedback without financial barriers [5].

VI. Conclusion and Future Works

In this paper, we present the development and evaluation of an AI-based Python education system designed to enhance learning through interactive quizzes, real-time feedback, and a built-in Chatbot for answering Python-related questions. The system demonstrated promising results, achieving a 28/30 consistency rate in quiz grading and a 26/30 consistency rate in entrance exam recommendations. These findings suggest that the system is effective in providing personalized learning experiences, though some inconsistencies remain.

However, the system is not yet ready for wide-scale deployment due to these identified inconsistencies and the need for further refinement. Additional work is required to improve the reliability of the system's feedback mechanisms and to expand the course content to cover all six planned chapters. Future efforts will focus on fine-tuning the AI components to ensure more standardized outputs and exploring opportunities for incorporating more advanced features that further support student learning. With these improvements, the system could eventually be deployed in an educational setting with appropriate oversight.

References

[1] L. B. Jung, J. A. Gudera, T. L. Wiegand, S. Allmendinger, K. Dimitriadis, and I. K. Koerte, "Chatgpt passes german state examination in medicine with picture questions omitted," *Deutsches Ärzteblatt International*, vol. 120, no. 21-22, p. 373, 2023.

[2] D. M. Katz, M. J. Bommarito, S. Gao, and P. Arredondo, "Gpt-4 passes the bar exam," *Philosophical Transactions of the Royal Society A*, vol. 382, no. 2270, p. 20230254, 2024.

[3] S. Bordt and U. von Luxburg, "Chatgpt participates in a computer science exam," *arXiv preprint arXiv:2303.09461*, 2023.

[4] Tynker. (2024) Intro to python course. Accessed: 2024-03-15. [Online]. Available: https://www.tynker.com/courses/python-1-jungle-run-adventure

[5] Codecademy. (2024) Learn python. Accessed: 2024-08-09. [Online]. Available: https://www.codecademy.com/catalog/language/python

[6] K. Academy. (2024) Accessed: 2024-07-08. [Online]. Available: https://www.khonacadomy.com

[7] (2024) TACo about Python. Accessed: 2024-08-10. [Online]. Available: https://www.taco-about-python.com/

[8] (2024) AI-based-education-system. Accessed: 2024-08-10. [Online]. Available: https://github.com/redportz/AI-based-education-system

[9] P. McKenna, "Multiple choice questions: answering correctly and knowing the answer," *Interactive Technology and Smart Education*, vol. 16, no. 1, pp. 59–73, 2019.

[10] J. J. Appleton, S. L. Christenson, and M. J. Furlong, "Student engagement with school: Critical conceptual and methodological issues of the construct," *Psychology in the Schools*, vol. 45, no. 5, pp. 369–386, 2008.

[11] Trinket. (2024) Embedded compiler. Accessed: 2024-08-09. [Online]. Available: https://trinket.io/

[12] K. Nguyen and M. A. McDaniel, "Using quizzing to assist student learning in the classroom: The good, the bad, and the ugly," *Teaching of psychology*, vol. 42, no. 1, pp. 87–92, 2015.

[13] M. Henderson, T. Ryan, D. Boud, P. Dawson, M. Phillips, E. Molloy, and P. Mahoney, "The usefulness of feedback," *Active Learning in Higher Education*, vol. 22, no. 3, pp. 229–243, 2021.

[14] D. C. Wynn and A. M. Maier, "Feedback systems in the design and development process," *Research in Engineering Design*, vol. 33, no. 3, pp. 273–306, 2022.

[15] M. E. Jang and T. Lukasiewicz, "Consistency analysis of chatgpt," *arXiv preprint arXiv:2303.06273*, 2023.

[16] A. M. A. Ausat, B. Massang, M. Efendi, N. Nofirman, and Y. Riady, "Can chat gpt replace the role of the teacher in the classroom: A fundamental analysis," *Journal on Education*, vol. 5, no. 4, pp. 16 100–16 106, 2023.