Revision-Bot: A Chatbot for Studying Past Questions in Introductory Programming

Chinedu Wilfred Okonkwo *Member, IAENG,* and Abejide Ade-Ibijola *Member, IAENG*

***Abstract*—Students struggle to get good grades in Computer Programming courses. One of the most important ways to improve the pass rate of a difficult subject, such as programming, is to implement a learning support system based on Artificial Intelligence (AI) technology. This system can provide students with a customised learning experience. This paper proposed a Revision-Bot (abbreviated as called `RevBot`), an intelligent interactive system that assists students in practicing past exam questions in a Python programming course. `RevBot` was created and implemented using the Snatchbot Chatbot API. An assessment was conducted to establish the usefulness of `RevBot`, and the results show that `RevBot` can help students improve their performance in the Python programming course.**

***Index Terms*—Chatbot, Revision, Python Programming, Introductory Programming, Exam Questions, Programming Pedagogy.**

I. Introduction

EDUCATION is one of the foundations of human capability which requires great improvement in the age of the Fourth Industrial Revolution (4IR) [1]. The COVID-19 pandemic brought a lot of restrictions including reduction in human-to-human contact operations. The educational system has been severely impacted by the pandemic, and as a result, teaching and learning are now delivered via an online platform [2, 3, 4]. Students receive their lectures at their various homes or different locations. Also, tutorials, exercises, and assessments are performed through online system [2]. Consequently, the use of technology innovation in education, such as the Chatbot, will help to improve the educational activities [5]. One of the most significant ways to encourage and foster an improved individualised learning experience is through the use of a Chatbot [6, 7].

Artificial Intelligence (AI) has a wide variety of uses in education. Chatbot technology, which is utilised to assist with teaching and managerial tasks, is one of the most recent AI applications applied to Education [8]. Students' assumptions that programming courses are difficult to pass or obtain a good grade have become one of the most important problems in studying programming [9]. One key factor influencing this problem is the lack of support provided to the students [10, 11]. According to studies, a large number of students drop out of introductory programming modules [10, 12], because there are not enough appropriate learning assistants to assist them [13, 14].

Although programming is a difficult course to teach [15, 16] and students dislike it, but it is a required skill in the 4IR era [17]. Different approaches, such as pedagogy models and software tools, have been proposed to assist students' comprehension of programming subjects [13, 18], but none deals with the interactions with the past exam questions. While these teaching approaches and software tools are helpful in learning and understanding programming subjects, there is continuous need to create more tools for supporting students and engaging them in more practice with past exam questions. Studies have shown that practice improves learning ability [19, 20, 21], therefore, the problem to be addressed in this study is:

- To design and develop a learning support tool – *Chatbot system*, that can enable the practice of past exam questions for revision prior to Python programming assessments.

The new Chatbot system is called `RevBot`. The system will interact with the user to provide the required type of questions which can be solved with pen and paper and the answers are displayed at the end of the exercise. The creation of `RevBot` followed the SnatchBot development platform concept which enables the deployment of the tool as a web application. The design of `RevBot` involves four basic units including user interface unit, Information unit, Storage unit, and Integration unit. The User Interface unit enable the user to interact with the system, the Information unit consist of the question and Artificial Intelligence Markup language (AIML) interfaces, and the Storage unit stores the pre-defined questions and answers. The AIML gives the `RevBot` the ability to compare the user's input to pre-defined data and provide a corresponding response to the user. As a web application, `RevBot` can be implemented or installed to social networking platforms via the Integration unit. Figure 1 depicts SnatchBot's operation flow.

This study makes the following contributions: We have:

1) developed an intelligent system that can interact with the user (student), providing past exam questions and answers for revision in a programming course,
2) the application of this system will help to improve overall performance of the students, and
3) the results of the evaluation show that the tool is useful and improves students' learning capability in Python programming subject.

The rest of this paper is structured as follows: Section II provides background information and related works, and Section III describes the Chatbot's design. Section IV discusses implementation and results, while Section V evaluates the Chatbot. Section VI brings the study to a close and makes

Manuscript received September 21, 2021; revised March 08, 2022. This work is based on research supported by the National Research Foundation (NRF) of South Africa (Grant Number: 119041).

Chinedu Wilfred Okonkwo is a Postdoctoral Research Fellow in the Centre of Applied Data Sciences, School of Consumer Intelligence and Information Systems, at the University of Johannesburg, South Africa. e-mail: chineduo@uj.ac.za

Abejide Ade-Ibijola is a Professor of Digital Transformation at the Johannesburg Business School, University of Johannesburg, South Africa. e-mail: abejide@jbs.ac.za, website: https://www.abejide.org.

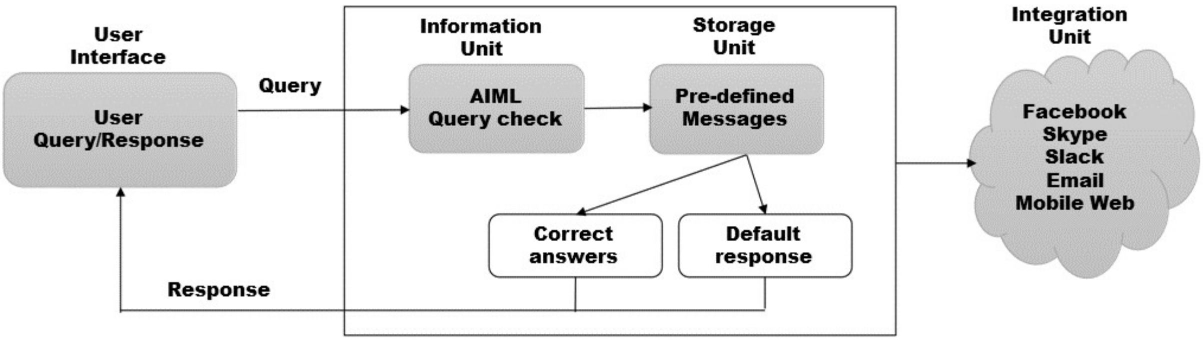


Fig. 1: Snatchbot flow diagram (Adapted from Okonkwo and Ade-Ibijola [8]). The figure is owned by the authors of this paper (Okonkwo and Ade-Ibijola).

recommendations for future work.

II. Background Information and Related Works

In this part, we present the theoretical information of the study with a focus on the applications of AI technologies in Education.

A. AI in Education

AI technology is becoming more common in our daily lives, and we are surrounded by it [22]. AI is being felt in Education, and traditional techniques are changing dramatically [23]. In this age of 4IR, the academic world is expected to become more convenient and personalized with the assistance of AI applications. These will make educational materials available through smart phones and computers, thereby altering the way individuals learn. AI in Education is expected to promote the automation of administrative duties [24], enabling organisations to minimise the time needed to finish challenging duties, and teachers to devote more time to student mentoring [25]. The use of AI systems will transform the Education system in the following ways:

Personalised Learning - AI customizes both in-class tasks and final exams to ensure that learners receive the greatest possible help. Teachers cannot be replaced, but they can do much better by providing personalised suggestions to each student [26].

Smart Content - This is a very interesting topic today. AI systems can generate quality digital content, including virtual content such as video conferencing, video presentations, and digital books [27]. AI systems use traditional curriculum to produce tailored textbooks for a variety of subjects. Consequently, textbooks are digitised, and new teaching interfaces are developed to assist learners of all ages and academic levels [28].

Global Learning - Education is unrestricted, and AI can assist in breaking down barriers. Technology enables remote or online teaching and learning of any subject anywhere in the world and at any moment. AI-powered Education provides learners with basic IT abilities [29]. With more innovations, a wider variety of lessons will be accessible online, and learners will learn from wherever they are with the assistance of AI technology [30].

Simplifying Administrative Tasks - AI has the potential to automate administrative tasks for educators and educational institutions [24]. Educators devote valuable time to marking tests, evaluating projects, and offering useful answers to learners; however, these processes can be automated with the help of technology. This implies that teachers would have sufficient time to engage in other activities such as mentoring and supervising their students [25].

New Efficiencies - AI may also be employed in complicated data modeling to enable the operations department to develop data-driven projections [31].

B. Chatbots in Education

In education, chatbot systems are a great way to change how institutions communicate with students today [13]. Students can now learn more about their subjects [8], opportunities, and extracurricular activities at school [32]. Teaching and learning methods evolve as technology becomes more prevalent in students' daily lives [33]. Educators can now provide students with a unique and engaged learning experience with the aid of AI technology [26]. Technology advances have enabled the development of a variety of educational resources to assist students in their understanding of study materials [34, 35]. Chatbot systems in education are improving student experiences and are a game changer in the field of creative educational technology [36, 37].

Chatbots, as a multipurpose AI technology, can be used by students to learn. By converting a lecture into a series of messages, it can be made to look like a chat conversation [8]. Chatbots can be used to determine and evaluate a student's level of comprehension on a regular basis, and thus change the learning process [38]. Teachers in today's world need modern-day strategies to help them handle their hectic schedules. Teachers can now use chatbots to automate their routine tasks [13]. While the teachers track the students' progress, the bots will answer any student question about the course, assignments, or deadlines. In today's Education industry, having a strong faculty, better classes, and well-equipped laboratories is no longer enough. Chatbots can be extremely useful in these situations. Students can get all the knowledge they need about their courses, modules, and faculties from chatbots that assist them during the admissions process. The bots may also serve as campus guides, assisting students upon their arrival [39].

C. Related Works

While we were unable to find a Chatbot specifically created to generate past questions and answers on a programming subjects, we did find the following related works in this area:

Answering Query Sharod *et at* [40] created a Chatbot system to respond to questions. The system will automatically respond to users' questions for knowledge purposes. Ranoliya *et al* [32] created a Chatbot based on the FAQ dataset that uses AIML and Latent Semantic Analysis to correctly respond to users' queries. Chandra and Suyanto [41] used a sequence-to-sequence model to create a Chatbot that responds to university student admission questions using pre-defined information from the university admission dataset.

Teaching Programming A chatbot called Python-Bot was created to help novice programmers understand Python's basic syntactic structures and semantics using the snatchbot development platform [8]. Gavin and Glavin [42] created an interactive and adaptive Chatbot system to help undergraduates learn Java programming. Ade-Ibijola developed a software tool that can automatically generate distinct practice programs in Python [43].

Student Assessment Benotti *et al* [38] presented Chatbot design, a software platform used to teach high school students about computer science concepts. To provide automatic assessment, the Chatbot employs finite state automata, pattern matching, and lemmatization techniques. A chatbot for self-assessment of core competencies was created to encourage students' ability to self-regulate [44].

D. Comparison of the Related Works

Table I compares some of previous research works that are related to our work. Although the previous works are various types of Chatbots systems in Education, none has attempted to address the issue of engaging the students with past exam questions for practice.

E. The Gap

According to the literature, programming is a difficult subject to teach, and many students struggle to understand it. As a result, the failure rate in programming courses has been demonstrated to be high. There are many Chatbot systems for educational purposes, but none of them deal with students interacting with past exam questions for practice. This is the gap we have addressed.

F. Motivation

Programming skills is important because the world is becoming more automated and it is necessary for humans to be able to control human-machine interaction [17]. Knowledge of Computer Programming improves mathematical and logical reasoning and can be useful in various situations, including decision-making and problem-solving [17, 45]. Thus, programming abilities are in high demand as the technological age progresses [46]. Unfortunately, students perceive Computer Programming courses to be difficult to learn, which contributes to a high failure rate in programming subjects [21]. It is important to simplify the Computer Programming learning process to inspire students' interest in programming. `RevBot` is created to present an interactive platform for improving program comprehension.

III. Design of `RevBot`

The system design of `RevBot` is discussed in this section, along with an example of a working algorithm. `RevBot` was created using the SnatchBot platform's concepts (a code-free design).

A. Overview of Snatchbot Platform

The SnatchBot framework is a full platform-as-a-service (PaaS), allowing users to design, test, and implement high-quality Chatbots without having to create their own structure. Chatbots developed on this platform can be used for virtually any communication medium, like email, text, and voice; social networking platforms like SnatchBot messaging network, slack, skype, Facebook, and Mobile Web. The Channel is another intriguing feature of SnatchBot. The SnatchBot Channel is comprised of a built-in Application Programming Interface (API) that allows Chatbots applications on mobile web and social media platforms. `RevBot` is deployed on a website via SnatchBot Channel. Students can access `RevBot` on the website using smartphones and Computers. SnatchBot includes an integrated editor for creating basic or complicated conversations with action buttons and translations. Also, it helps one to develop a variety of *interactions* in association with the Chatbot's activities. SnatchBot uses *interactions* to describe the *subjects*. *Interactions* specifies a unique activity of the Chatbot, and *Subjects* are the pre-defined contents of the activity (audio, messages, graphs, videos, etc.).

B. `RevBot` Development

`RevBot` was built using the SnatchBot API and its predefined tools, which are powered by various Natural Language Processing (NLP) algorithms. During the development of `RevBot`, we created a design comprised of various *interactions* and *subjects*. There are 33 *interactions* in `RevBot` including Introduction, Contents, questions, Answers, Support etc. Because `RevBot` is primarily concerned with past exam questions and answers, the *subjects* are mostly messages. Figure 2 depicts parts of the `RevBot` developing scheme.

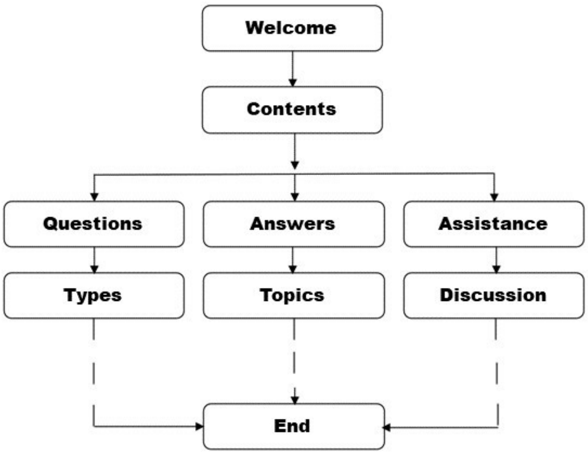


Fig. 2: `RevBot` developing scheme

Interactions are linked together locally and globally with SnatchBot's capabilities for efficient operations. Local connections are only active on a particular *interaction*, while

TABLE I: Comparison with the related works

S/N	Title	Author	Comparison
1	An Educational Chatbot for Answering Queries.	Sharod et al, 2020.	They used the K-means clustering algorithm to create a Chatbot that can respond to a variety of questions.
2	Chatbot for University Related FAQs.	Ranoliya et al, 2017.	They created a Chatbot for answering University FAQs using AIML and LSA.
3.	Indonesian chatbot of university admission using a question answering system based on sequence-to-sequence model.	Chandra and Suyanto, 2019.	Using the sequence-to-sequence model, they created a Chatbot that responds to the University admission questions.
4	CLuAI – Conversational Learning using Artificial Intelligence: An Interactive and Adaptive Chatbot Learning Framework for Teaching Programming.	Gavin and Glavin, 2020.	Created a Chatbot prototype that can teach Java programming interactively.
5	Python-Bot: A Chatbot for teaching Python Programming.	Okonkwo and Ade-Ibijola, 2020.	Using the Snatchbot development platform, they created a Chatbot for teaching Python programming.
6	A Tool for Introducing Computer Science with Automatic Formative Assessment.	Benotti et al, 2018.	They used pattern matching, cutting-edge lemmatization techniques, and finite state automata to create a Chatbot that can introduce programming concepts to students for automatic formative assessment.
7	Co-design for a Competency Self-assessment Chatbot and Survey in Science Education.	Durall and Kapros, 2020.	They created a Chatbot for student Formative assessment using the Co-design process.

global connections are active on all *interactions*. Every connection is a logical statement in form of: if x then y else z. The following algorithm adapted from [8] depicts the main action in the information decision part of the `RevBot`.

Algorithm 1: `RevBot` Information-Decision (Adapted from [8])

```
Input: Question (user queries obtain from user text
  input);
if Question Matches Storage_Bank then
  | return predefined response;
else
  | return default answer;
end
```

The algorithm takes the user's text as input (Line 1), compares it to the pre-defined response in the storage unit (Line 2), and returns the matching message (Line 3). If no matching pattern is found, Line 5 returns the default answer. Line 6 ends the process.

Figure 3 depicts the `RevBot` functional decision in response to user queries. It shows that when a student connects to `RevBot`, the bot sends him/her a welcome message in which it asks the student if he/she wants to interact with the bot. If the user responds YES, `RevBot` will direct the user to the contents. The user selects the specific operation that he or she desires. The procedure is repeated until the user is done.

IV. Implementation and Results

The system architecture of RevBot consists of four major sections including the Welcome, Questions, Answers and Support.

Welcome Section - This section introduces the Chatbot and confirms whether the student is ready to interact with `RevBot` or not.

Questions Section – It contains various types of past exam questions from the Python programming course. The user (Student) chooses the type of question he or she wants to try or practice from this section, and the bot responds accordingly.

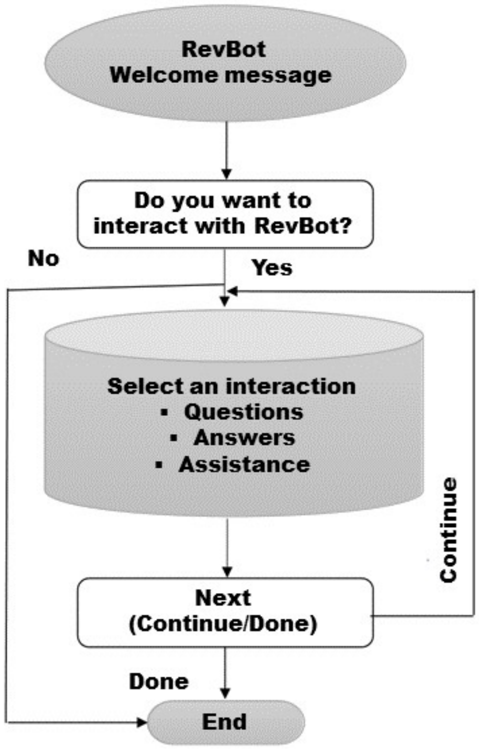


Fig. 3: `RevBot` functional decision flow
`RevBot` starts operation with an introduction, followed by the generation of questions and answers for users.

Answers Section – This section contains all the answers to the questions. When the exercises are completed, students can check the correct answers to each question as well as the assigned mark/score.

Support Section – This section assists the student during the conversation with the bot. Students may request the assistance of a Lecturer or Tutor at any time during the conversation.

The `RevBot` operation begins with an introduction to the user (student) and a question about whether or not the user is ready to converse. The Chatbot takes the student's question, arranges it, and responds by conveying pre-defined Python programming past exam questions to the student. Also, it provides answers to the questions. With a pen and paper, the student can practice a variety of questions and then request answers. Figure 4a and 4b show samples of a student conversation with the RevBot.

The SnatchBot platform includes NLP capabilities, which allow RevBot to understand the student's query and respond appropriately. This is accomplished by splitting a sentence down into two main components, such as entity and intent. The entity is the object, and the intent is the purpose of the sentence. To complete a specific task, we needed to specify the kinds of entity and intent the NLP module would identify. SnatchBot comes with a set of predefined NLP models that we used in our development. Machine Learning is used by the models to make choices on every new sentence they encounter, based on the samples given. Two NLP models were used in this study to assist students: one to determine the date (entity) and another to determine the time (intent) for a meeting appointment. During an interaction with RevBot, a user or student may seek the assistance of a Lecturer or Tutor, so a meeting may be required. These models aid in the generation of a meeting date and time. The proposed meeting will be notified to the Lecturer/Tutor via email, and the student will receive support if the Lecturer/Tutor confirms the meeting. Figures 5a and 5b show the RevBot NLP module capability of scheduling a meeting for student support.

V. Evaluation and Application of RevBot

This section presents the findings of an online poll of students' impressions of RevBot's use and potential usefulness.

A. *Evaluation of* RevBot

We created RevBot, an intelligent system that can interact with students on past exam questions and provide the corresponding answers. The primary goal is for students to practice these questions in preparation for their Python programming course exam. This operation will aid in improving student success in the Python programming course and increase the course pass rate.

An online survey was conducted at the University of Johannesburg, South Africa. to determine the students' perceptions of the usefulness of the tool. The purpose of the evaluation is to determine students':

1) perception of Computer Programming,
2) perception of the use of the tool (RevBot), and
3) whether the practice of the past exam questions will help to improve the student's performance in a Python programming course examination.

The collected quantitative data was statistically analysed to describe the frequency and response percentage of the respondents. The outcomes are detailed below.

B. *Participants' Profile*

The study comprised 176 university undergraduates, including both first-year students (90.6%) and returning or old students (9.4%). The sample population consists of male and female Computer Science and other related course students taking the introductory Python programming subject for the first time, as well as those who have previously taken it. This audience was chosen because they are potential users of this tool and are currently enrolled in a Python programming module. Students were polled on their thoughts on:

Computer Programming Course In general, Computer programming courses were investigated to determine how students felt about taking them. 81% of the respondents perceived the courses to be difficult, 12.5% consider them like every other course and easy to learn, and 6.5% were indifferent. See Figure 6a.

User-friendly The assessment looked at the students' perceptions of RevBot and how easy it is for them to use. 85.6% believed that the design of RevBot is simple and easy to use, 4.5% find it difficult to use, and 9.9% could not decide whether it is easy or difficult to use. See Figure 6b.

Preparation of Examination This section examined the questions provided by RevBot to see if they were possible exam questions that could aid students in their preparation for the Python programming course exam. Interestingly, 88.2% of students believed that the tool could help them prepare well for the Python programming exam in the hopes of getting a good grade, 1.9% disagreed and 9.9% were unsure, and no one disagreed. See Figure 6c.

Improved Performance This tested to see if the use of RevBot could help students improve their exam performance and get better grades. 87.2% agreed that using RevBot could help them revise and improve their comprehension abilities, resulting in higher exam scores, while 3.4% disagreed and 9.4% were indifferent. See Figure 6d.

According to the findings, students perceived Computer Programming courses to be difficult to understand, and they struggled to pass them. The students thought RevBot's design was simple and easy to use. Concerning the tool's ability to assist them in exam preparation, the questions provided by RevBot are likely exam questions that, if practiced and understood, will aid them in passing the exam. As a result, using RevBot can help them improve their learning abilities in the Python programming course, improve their performance abilities, and help them get a better grade in the course.

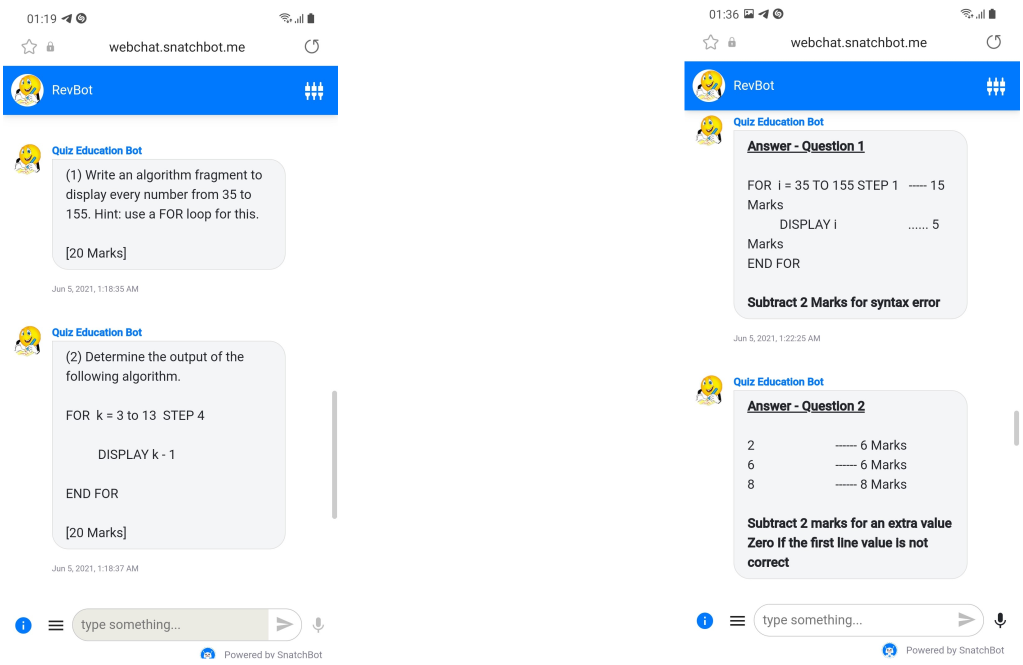
C. *Application of* RevBot

RevBot is an educational learning tool designed to assist students in preparing for Python programming course quizzes and exams. Students can use this tool to practice previous exam questions and learn about the marks/scores assigned to each type of question. The use of this tool will help to improve students' learning capabilities and, as a result, lower the failure rate in Python programming courses.

VI. Conclusions and Future Works

A. *Conclusions*

In this study, we created RevBot, an intelligent interactive system that can interact with students by presenting past exam questions to be solved with a pen and paper, as well as the answers. RevBot was created with the SnatchBot Chatbot API and can be deployed to websites and social



Quiz Education Bot

(1) Write an algorithm fragment to display every number from 35 to 155. Hint: use a FOR loop for this.

[20 Marks]

Jun 5, 2021, 1:18:35 AM

Quiz Education Bot

(2) Determine the output of the following algorithm.

```
FOR  k = 3 to 13  STEP 4

        DISPLAY k - 1

END FOR
```

[20 Marks]

Jun 5, 2021, 1:18:37 AM

Quiz Education Bot

Answer - Question 1

```
FOR  i = 35 TO 155 STEP 1   ----- 15
Marks
        DISPLAY i             ...... 5
Marks
END FOR
```

Subtract 2 Marks for syntax error

Jun 5, 2021, 1:22:25 AM

Quiz Education Bot

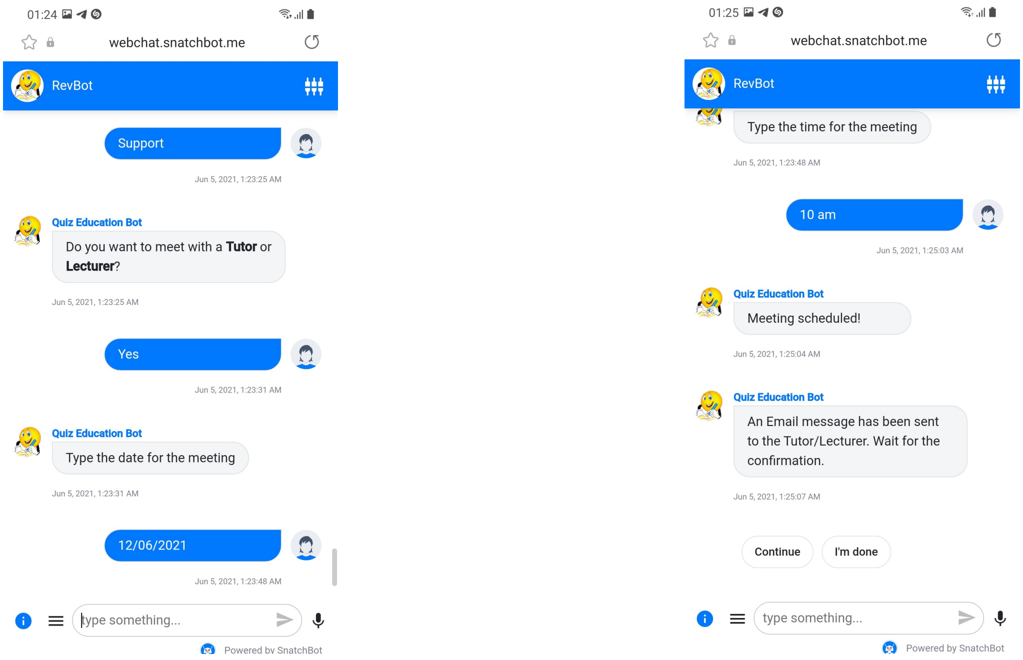
Answer - Question 2

```
2               ------ 6 Marks
6               ------ 6 Marks
8               ------ 8 Marks
```

Subtract 2 marks for an extra value Zero if the first line value is not correct

(a) Conversation with RevBot-Questions (b) Conversation with RevBot-Answers

Fig. 4: Screenshots of sample conversation with RevBot



Support

Jun 5, 2021, 1:23:25 AM

Quiz Education Bot

Do you want to meet with a **Tutor** or **Lecturer**?

Jun 5, 2021, 1:23:25 AM

Yes

Jun 5, 2021, 1:23:31 AM

Quiz Education Bot

Type the date for the meeting

Jun 5, 2021, 1:23:31 AM

12/06/2021

Jun 5, 2021, 1:23:48 AM

Type the time for the meeting

Jun 5, 2021, 1:23:48 AM

10 am

Jun 5, 2021, 1:25:03 AM

Quiz Education Bot

Meeting scheduled!

Jun 5, 2021, 1:25:04 AM

Quiz Education Bot

An Email message has been sent to the Tutor/Lecturer. Wait for the confirmation.

Jun 5, 2021, 1:25:07 AM

(a) Processing of date for a meeting (b) Processing of time for a meeting

Fig. 5: RevBot NLP models capability

media platforms. A survey was used to conduct the evaluation, and the results indicate that a large number of students taking the Python programming course agreed that RevBot is simple to use and can help them prepare for the Python programming course examination.

B. Future Works

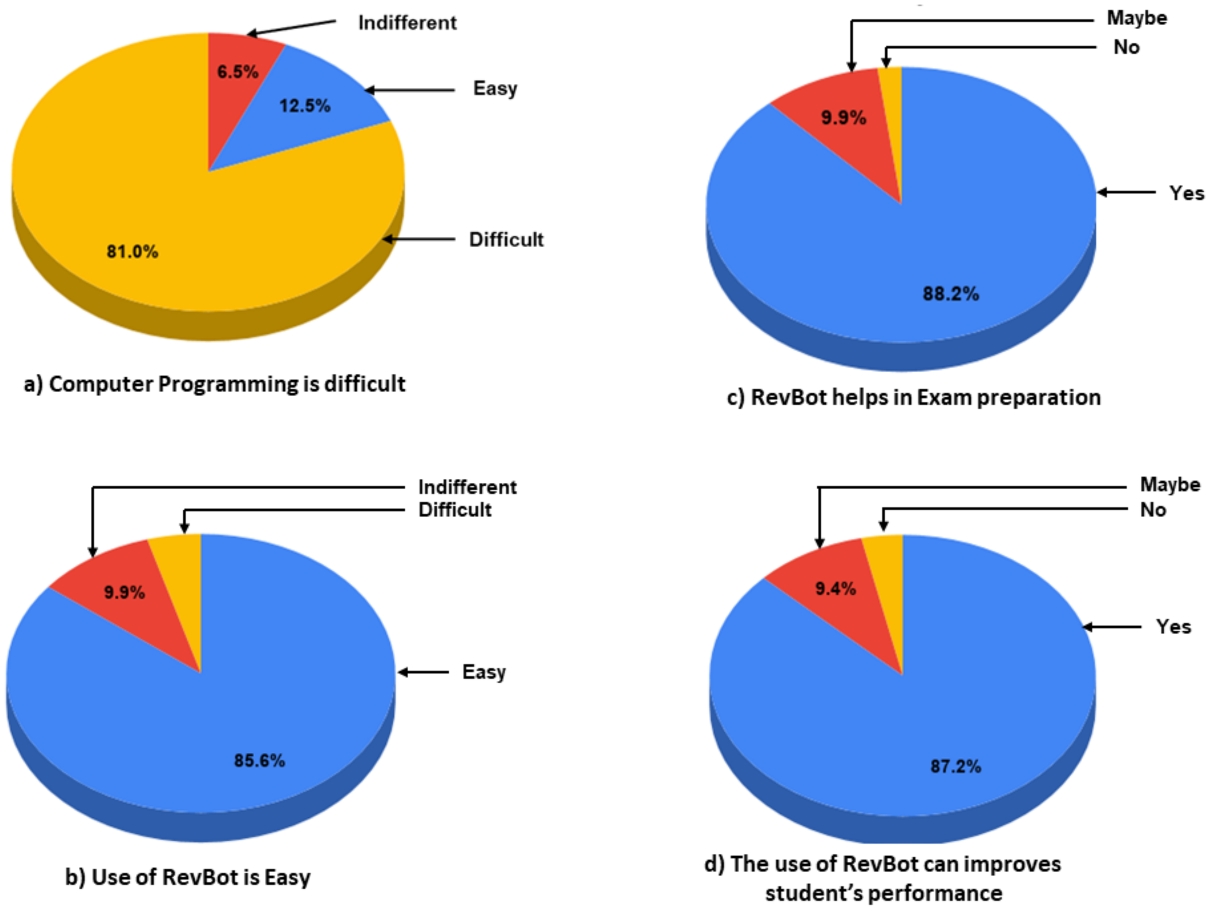
This tool only covers the Python programming course. Other computer programming courses, as well as other undergraduate courses, are expected to be added in the future to improve students' learning abilities.

Acknowledgement

Any opinion, findings and conclusions or recommendations expressed in this material are those of authors and therefore the National Research Foundation (NRF) of South Africa does not accept liability in regard thereto.

References

[1] Bryan Edward Penprase. The fourth industrial revolution and higher education. *Higher education in the era of the fourth industrial revolution*, 207, Palgrave



a) Computer Programming is difficult

c) RevBot helps in Exam preparation

b) Use of RevBot is Easy

d) The use of RevBot can improves student's performance

Fig. 6: Evaluation results of the `RevBot`

Macmillan, Singapore, 2018. https://doi.org/10.1007/978-981-13-0194-0_9.

[2] Simon Burgess and Hans Henrik Sievertsen. Schools, skills, and learning: The impact of covid-19 on education, 2020.

[3] Giorgio Marinoni, Hilligje Van't Land, and Trine Jensen. The impact of covid-19 on higher education around the world. *IAU Global Survey Report*, 2020.

[4] Giorgio Di Pietro, Federico Biagi, Patricia Costa, Zbigniew Karpiński, and Jacopo Mazza. *The likely impact of COVID-19 on education: Reflections based on the existing literature and recent international datasets*, volume 30275. Publications Office of the European Union, 2020.

[5] Ana Marta Aleixo, Susana Leal, and Ulisses Miranda Azeiteiro. Conceptualization of sustainable higher education institutions, roles, barriers, and challenges for sustainability: An exploratory study in Portugal. *Journal of cleaner production*, 172:1664–1673, 2018.

[6] Sam Cunningham-Nelson, Wageeh Boles, Luke Trouton, and Emily Margerison. A review of chatbots in education: practical steps forward. In *30th Annual Conference for the Australasian Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate*, pages 299–306. Engineers Australia, 2019.

[7] Brian Whalley, Derek France, Julian Park, Alice Mauchline, and Katharine Welsh. Developing active personal learning environments on smart mobile devices. In *Proceedings of the Future Technologies Conference*, pages 871–889. Springer, 2019.

[8] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Python-bot: A chatbot for teaching python programming. *Engineering Letters*, 29(1):pp25–34.

[9] Yizhou Qian and James Lehman. Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1):1–24, 2017.

[10] Sean B Eom and Nicholas Ashill. The determinants of students' perceived learning outcomes and satisfaction in university online education: An update. *Decision Sciences Journal of Innovative Education*, 14(2):185–215, 2016.

[11] Carmen Richardson and Punya Mishra. Learning environments that support student creativity: Developing the scale. *Thinking Skills and Creativity*, 27:45–54, 2018.

[12] Sohail Iqbal Malik. Role of ADRI model in teaching and assessing novice programmers. Technical report, Deakin University, 2016.

[13] Fabio Clarizia, Francesco Colace, Marco Lombardi, Francesco Pascale, and Domenico Santaniello. Chatbot: An education support system for student. In *International Symposium on Cyberspace Safety and Security*, pages 291–302. Springer, 2018. DOI:https://doi.org/10.1007/978-3-030-01689-0_23.

[14] M Naveen Kumar, PC Linga Chandar, A Venkatesh Prasad, and K Sumangali. Android based educational chatbot for visually impaired people. In *2016 IEEE International Conference on Computational Intelligence*

and Computing Research (ICCIC), pages 1–4. IEEE, 2016.

[15] Vishv Mohan Malhotra and Ashish Anand. Teaching a university-wide programming laboratory: Managing a c programming laboratory for a large class with diverse interests. In *Proceedings of the Twenty-First Australasian Computing Education Conference*, pages 1–10, 2019.

[16] Jesennia Cárdenas-Cobo, Amilkar Puris, Pavel Novoa-Hernández, José Angel Galindo, and David Benavides. Recommender systems and scratch: An integrated approach for enhancing computer programming learning. *IEEE Transactions on Learning Technologies*, 13(2):387–403, 2019.

[17] TV Kondratyuk. Fourth industrial revolution: What competences are necessary for employees? *Strategic decisions and risk management*, (3):66–79, 2018.

[18] Greg L Nelson, Benjamin Xie, and Andrew J Ko. Comprehension first: evaluating a novel pedagogy and tutoring system for program tracing in cs1. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, pages 2–11, 2017.

[19] Gary J Hill. Review of a problems-first approach to first year undergraduate programming. In *Software Engineering Education Going Agile*, pages 73–80. Springer, 2016.

[20] Sohail Iqbal Malik, Roy Mathew, Rim Al-Nuaimi, Abir Al-Sideiri, and Jo Coldwell-Neilson. Learning problem solving skills: Comparison of e-learning and m-learning in an introductory programming course. *Education and Information Technologies*, 24(5):2779–2796, 2019. DOI:https://doi.org/10.1007/s10639-019-09896-1.

[21] A model for teaching an introductory programming course using ADRI, author=Malik, Sohail Iqbal and Coldwell-Neilson, Jo, journal=Education and Information Technologies, volume=22, number=3, pages=1089–1120, year=2017, publisher=Springer.

[22] Ben Williamson and Rebecca Eynon. Historical threads, missing links, and future directions in ai in education, 2020. https://doi.org/10.1080/17439884.2020.1798995.

[23] Yu Yan and Qing Li. Research status, problems and trends of" ai plus education": Visual analysis based on core journals and cssci journal literature. In *2019 International Symposium on Educational Technology (ISET)*, pages 281–283. IEEE, 2019. https://doi.org/10.1109/ITME.2019.00093.

[24] Oussama H Hamid, Norris Lee Smith, and Amin Barzanji. Automation, per se, is not job elimination: How artificial intelligence forwards cooperative human-machine coexistence. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pages 899–904. IEEE, 2017. https://doi.org/10.1109/INDIN.2017.8104891.

[25] Ho Thao Hien, Pham-Nguyen Cuong, Le Nguyen Hoai Nam, Ho Le Thi Kim Nhung, and Le Dinh Thang. Intelligent assistants in higher-education environments: the fit-ebot, a chatbot for administrative and learning support. In *Proceedings of the ninth international symposium on information and communication technology*, pages 69–76, 2018. DOI: https://doi.org/10.1145/3287921.3287937.

[26] Bhoovandev Fakooa, Mehnaz Bibi Diouman Banon, and Baby Gobin-Rahimbux. A smart mobile application for learning english verbs in mauritian primary schools. In *Information Systems Design and Intelligent Applications*, pages 399–409. Springer, 2019.

[27] Bill Cope, Mary Kalantzis, and Duane Searsmith. Artificial intelligence for education: Knowledge and its assessment in ai-enabled learning ecologies. *Educational Philosophy and Theory*, pages 1–17, 2020. https://doi.org/10.1080/00131857.2020.1728732.

[28] Marco Furini. Automatic and smart content transformation of video lectures. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2021.

[29] Wenbin Gan, Yuan Sun, Shiwei Ye, Ye Fan, and Yi Sun. Ai-tutor: Generating tailored remedial questions and answers based on cognitive diagnostic assessment. In *2019 6Th international conference on behavioral, economic and socio-cultural computing (BESC)*, pages 1–6. IEEE, 2019. https://doi.org/10.1109/BESC48373.2019.8963236.

[30] V Kavitha and Resham Lohani. A critical study on the use of artificial intelligence, e-learning technology and tools to enhance the learners experience. *Cluster Computing*, 22(3):6985–6989, 2019.

[31] Dakuo Wang, Justin D Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. Human-AI collaboration in data science: Exploring data scientists' perceptions of automated ai. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24, 2019.

[32] Bhavika R Ranoliya, Nidhi Raghuwanshi, and Sanjay Singh. Chatbot for university related faqs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1525–1530. IEEE, 2017.

[33] Renée Crawford. Rethinking teaching and learning pedagogy for education in the twenty-first century: blended learning in music education. *Music Education Research*, 19(2):195–213, 2017. https://doi.org/10.1080/14613808.2016.1202223.

[34] Ildikó Horváth. Evolution of teaching roles and tasks in vr/ar-based education. In *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 000355–000360. IEEE, 2018. https://doi.org/10.1109/CogInfoCom.2018.8639907.

[35] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033, 2021.

[36] Matthew Verleger and James Pembridge. A pilot study integrating an ai-driven chatbot in an introductory programming course. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–4. IEEE, 2018.

[37] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Evaluating the ethical implications of using chatbot systems in higher education. *digiTAL 2021*, page 68, 2021.

[38] Luciana Benotti, Mara Cecilia Martnez, and Fernando Schapachnik. A tool for introducing computer science with automatic formative assessment. *IEEE transac-*

tions on learning technologies, 11(2):179–192, 2017.

[39] Katlego Mabunda and Abejide Ade-Ibijola. Pathbot: An intelligent chatbot for guiding visitors and locating venues. In *2019 6th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pages 160–168. IEEE, 2019.

[40] Sharob Sinha, Shyanka Basak, Yajushi Dey, and Anupam Mondal. An educational chatbot for answering queries. In *Emerging Technology in Modelling and Graphics*, pages 55–60. Springer, 2020.

[41] Yogi Wisesa Chandra and Suyanto Suyanto. Indonesian chatbot of university admission using a question answering system based on sequence-to-sequence model. *Procedia Computer Science*, 157:367–374, 2019.

[42] Michael Gavin and Frank G Glavin. Cluai–conversational learning using artificial intelligence: An interactive and adaptive chatbot learning framework for teaching programming. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, pages 556–556, 2020.

[43] Abejide Ade-Ibijola. Syntactic generation of practice novice programs in python. *Communications in Computer and Information Science (CCIS)*, pages 158–172, 2018. Doi:https://DOI.org/10.1007/978-3-030-05813-5_11.

[44] Eva Durall and Evangelos Kapros. Co-design for a competency self-assessment chatbot and survey in science education. In *International Conference on Human-Computer Interaction*, pages 13–24. Springer, 2020.

[45] Sarantos Psycharis and Maria Kallia. The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5):583–602, 2017. https://doi.org/10.1007/s11251-017-9421-5.

[46] Tien-Chi Huang, Yu Shu, Shu-Hsuan Chang, Yan-Zhang Huang, Sung-Lin Lee, Yong-Ming Huang, and Chien-Hung Liu. Developing a self-regulated oriented online programming teaching and learning system. In *2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pages 115–120. IEEE, 2014.