Enhancing Python Programming Education through Dynamic Code Execution and Assessment Methods

Nurtugan Azatbekuly[1] and Bazargul Matkerim[1]

[1]*Al-Farabi Kazakh National University, 71 Al-Farabi Avenue, Almaty, 050040, Kazakhstan*

Abstract

This study introduces a web-based educational platform with a novel approach to Python programming skill development for students. It emphasizes task-based learning, real-time code checking, and dynamic assessments, all of which are aimed at enhancing problem-solving abilities and coding practices. The primary objective of this platform is to advance education, particularly in the computer science field. It seeks to inspire a deeper understanding of Python programming, creative problem-solving, and nurture proficient programmers through hands-on learning experiences. The platform employs dynamic code execution, a global namespace, and regular expressions for real-time code checking. The platform's unique features such as real-time feedback, dynamic assessments, and visualizations contribute to a more effective and engaging learning experience. It fosters students' problem-solving skills, adherence to coding best practices, and an overall improvement in Python programming. This innovative educational tool offers a comprehensive approach to Python programming skill development, enhances students' problem-solving abilities, and encourages adherence to best coding practices. It aims to inspire a deeper understanding of Python programming, creative problem solving, and nurture proficient programmers, contributing to the advancement of education in the field of computer science. This study provides a detailed account of the platform's architecture, pedagogical foundations, and results of usage cases, showcasing its potential as an innovative educational resource.

Keywords

Programming education, Python

1. Introduction

Programming education is a dynamic field that constantly seeks innovative approaches to empower students with the skills and knowledge required to excel in software development. In this era of rapid technological advancement, Python has emerged as a cornerstone in computer science and software engineering programs owing to its simplicity and versatility. Awareness of the key design principles of educational applications is crucial. In [1] and [2], it was observed that educational applications should be engaged and stimulated in the context of the educational process. Additionally, they should provide opportunities for practice and feedback. In the development of the work done in these articles [3-5], peculiar systems were used for effective programming training. It is also worth noting this article [6], in which the system used a visual programming language that allowed students to solve programming problems by dragging blocks of code. To facilitate an engaging and effective learning environment, we introduced a novel web-based platform that transcends traditional instructional methods and offers a unique, interactive approach to Python programming education.

The appeal of Python lies in its readability, ease of use, and wide-ranging applicability. However, mastering language and its associated concepts necessitates a pedagogical framework that challenges students while providing invaluable feedback to foster their growth. Our platform is designed to address this educational imperative by affording students the opportunity to tackle Python programming tasks in a manner that not only builds their coding acumen but also instills a deeper understanding of the nuances of the language.

DTESI 2023: Proceedings of the 8th International Conference on Digital Technologies in Education, Science and Industry, December 06–07, 2023, Almaty, Kazakhstan

nurtugang17@gmail.com (N. Azatbekuly); bazargul.matkerim@gmail.com (B. Matkerim)

0009-0007-5843-8995 (N. Azatbekuly); 0000-0002-5336-687X (B. Matkerim)

This study elucidates the architectural underpinnings of our innovative platform, highlighting the pivotal role played by its dynamic code-checking mechanisms. While traditional pedagogical methods rely on static assignments, our platform generates randomized variables for each task, thus offering a distinctive, ever-evolving experience for each student. This approach encourages creative problem-solving and critical thinking, as students grapple with the multifaceted challenges presented by each task.

Central to the platform's educational ethos is its use of dynamic code assessment techniques, such as the restriction of specific keywords, enforcement of essential programming constructs, and real-time code validation. These mechanisms ensure not only code correctness, but also adherence to best coding practices, thereby promoting the cultivation of proficient and responsible programmers.

To further enhance the learning experience, the platform provides students with insightful visualizations, including charts and performance metrics, enabling them to systematically monitor their progress. Assessing user progress in learning is crucial to enable the identification of strengths and weaknesses and to guide appropriately oriented learning in educational applications, as highlighted by a referenced paper [7]. These visual aids offer a comprehensive view of their task-solving proficiency, empower them to gain insight into their performance by difficulty level and theme and motivate a continuous journey of improvement. In educational studies of these works [8-10], it was proven that the greatest effect was achieved for applications focused on specific learning skills. The conclusions of [11] suggest that the possibility of using a flexible software development process for teaching computer programming should be considered.

This study seeks to elucidate the novel approach adopted in our educational platform and its implications for Python programming education. We aim to contribute to the evolution of programming education by harnessing technology to inspire students, foster a deeper understanding of Python, encourage creative problem-solving, and ultimately shape the next generation of proficient programmers. The subsequent sections delve into the platform's architecture, pedagogical foundations, and empirical evidence of its effectiveness, thus providing a comprehensive account of this pioneering educational tool.

2. Methods and solutions

In our ongoing commitment to advancing Python programming education, we designed and developed an adaptive web application that represents a pioneering addition to the educational technology landscape. This innovative web application leverages Python’s inherent capabilities to offer a dynamic and interactive platform for users to engage with programming concepts in a hands-on and immersive manner. At its core, our web application harnesses the power of Python's built-in functions to facilitate a dynamic code execution. This feature is central to our mission of creating a learning environment in which users can not only grasp the theoretical aspects of programming but also apply this knowledge in practical scenarios. By executing the code in real-time, our platform enables learners to experiment, troubleshoot, and refine their coding skills, thereby bridging the gap between theory and practice. To gauge the effectiveness of our web application in enhancing user learning, we incorporated various assessment tools and diagrams to evaluate user progress and performance. These diagrams provide visual representations of learning trajectories, allowing educators and learners to track the evolution of their coding proficiency over time. This data-driven approach offers valuable insights into the impact of our platform on learning improvements.

The section concentrates on analyzing the problem generation system, testing it against different regular expressions, and comparing it with the correct outcome. Thus, it is crucial to present a database structure in the context of web app development.

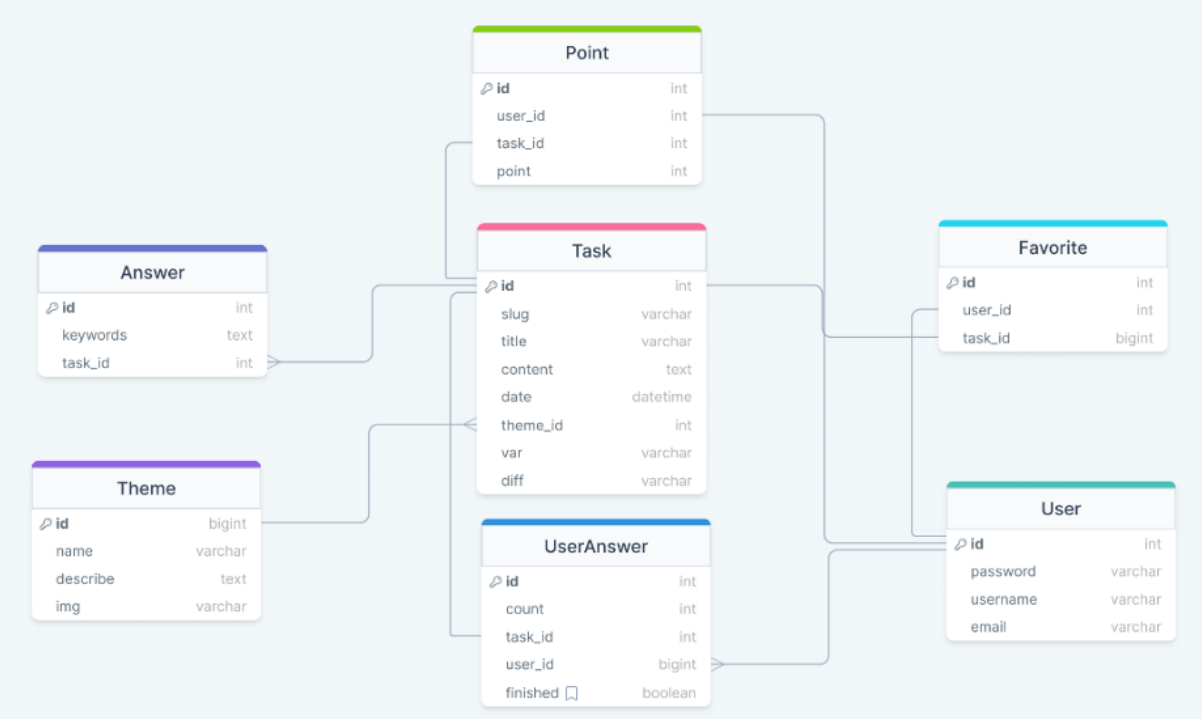

Figure 1: Class diagram of developed web-application

2.1 Task preparation

In this section, we present a detailed exploration of the task variable generation process in the context of a web application designed to facilitate educational endeavors. Systematic generation of random data plays a pivotal role in enhancing the educational experience offered by platforms. The dynamic nature of this process ensures that with each visit to the task page, users encounter a unique and tailored set of variables, specifically engineered to correspond to the characteristics and demands of individual tasks.

The linchpin of this process resides within the implemented function for generating variables, which was meticulously crafted within the Python file in the Django project. This function operates as the crux of the data-generation mechanism, orchestrating a series of conditional expressions informed by validated task IDs. This sophisticated approach ensures the creation of data that is distinct and attuned to the intricacies of each task. The versatility of this data generation method is manifested through a range of techniques, including random number generation within customizable bounds. For selected tasks, the configurability of these bounds allows precise data tailoring.

Moreover, the data-generation mechanism encompasses the acquisition of words, sentences, and lists. To achieve this, the function orchestrates a harmonious interplay of methods, such as parsing data from designated websites to generate words, harnessing the capabilities of the essential_generator library to formulate sentences, and orchestrating complex algorithms to synthesize random lists. The composite result is a comprehensive dataset that exhibits diversity and intricacy.

Integral to the operation of the platform is the creation and management of a dictionary that encapsulates variables and their corresponding values, constituting a vital prerequisite for the functioning of the `eval()` function. This dictionary structure was meticulously constructed with the names of variables meticulously extracted from the database entries associated with each task. Within the `eval()` function, these variable names assume the role of keys within the dictionary, while the corresponding values are drawn from the data previously generated using the diverse array of data generation techniques. This systematic organization is pivotal, as the

`eval()` function's `globals` parameter exclusively accommodates a dictionary format, thus necessitating this structured arrangement for the successful evaluation of user-submitted code.

In summary, the random variable generation process is a multifaceted and meticulously orchestrated component of the platform, contributing to its ability to dynamically tailor tasks and provide users with a unique and enriching educational experience. This process underscores the platform's commitment to fostering creative problem-solving and critical thinking by presenting users with an ever-evolving and diverse set of variables, thus facilitating an environment in which learning thrives through active engagement.

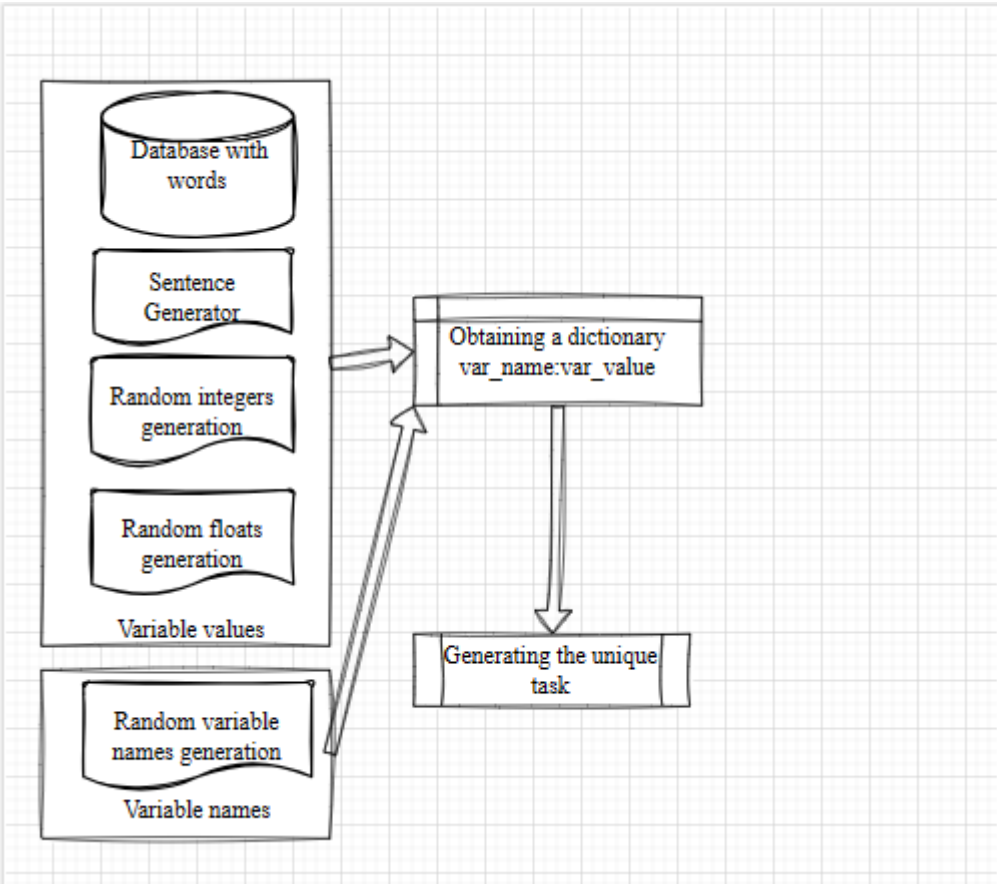

Figure 2: Generating the task with unique variables

2.2 Dynamic code execution and Runtime Environment

In the subsequent stages of our operational framework, we delve into the intricate process of code execution. Specifically, we examine the execution of two distinct types of code: code retrieved from the database, which is considered "correct" code, and user-submitted code. This phase is integral to our platform's educational objectives as it allows for a detailed comparison to determine the correctness of the user's code in relation to the predetermined standards embodied by the "correct" code.

The execution of "correct" code, sourced from our database, is a multifaceted procedure. This code undergoes a series of manipulations and evaluations that are pivotal for assessing conformity to task requirements. The process begins with the importation of essential application models, with the "TAnswer" table taking the center stage. This importation lays the groundwork for subsequent variable declarations and manipulation.

Variable declarations are a crucial aspect of this process and are managed through the utilization of the “globals(). Update the ()” method. This method integrates the variables stored within a specialized dictionary known as "vardict," meticulously constructed to encompass

variables and their associated values. Careful orchestration of these variables is central to the execution of the code.

Moreover, the code specific to each task is retrieved from the database through the "TAnswer" table. This code serves as a benchmark for the system, representing the standard of correctness against which user-generated code must be measured. The retrieval process was facilitated seamlessly, and the imported code was scrutinized to ensure compliance with predefined standards and task-specific requirements.

Concurrently, user-submitted code undergoes a different form of execution. Unlike the "correct" code, which undergoes a series of meticulous manipulations, user-generated code is executed within the confines of the “eval()” method. This approach ensures efficient execution of user-submitted code and allows for a streamlined comparison against the benchmark "correct" code.

Furthermore, our system incorporates an additional layer of scrutiny using regular expressions (regex) for keyword validation. Certain tasks require the inclusion or exclusion of specific keywords to determine the correctness of the code. This facet of our platform facilitates a granular examination of the user code to assess its adherence to predefined keyword requirements by adding an additional layer of precision to the assessment process.

In summary, the execution and comparison of codes within our platform is a multifaceted process that is meticulously designed to ensure educational integrity. This approach allows for a systematic evaluation of user-generated code against predefined benchmarks, offering users a constructive and educational experience, while maintaining the highest standards of quality and correctness.

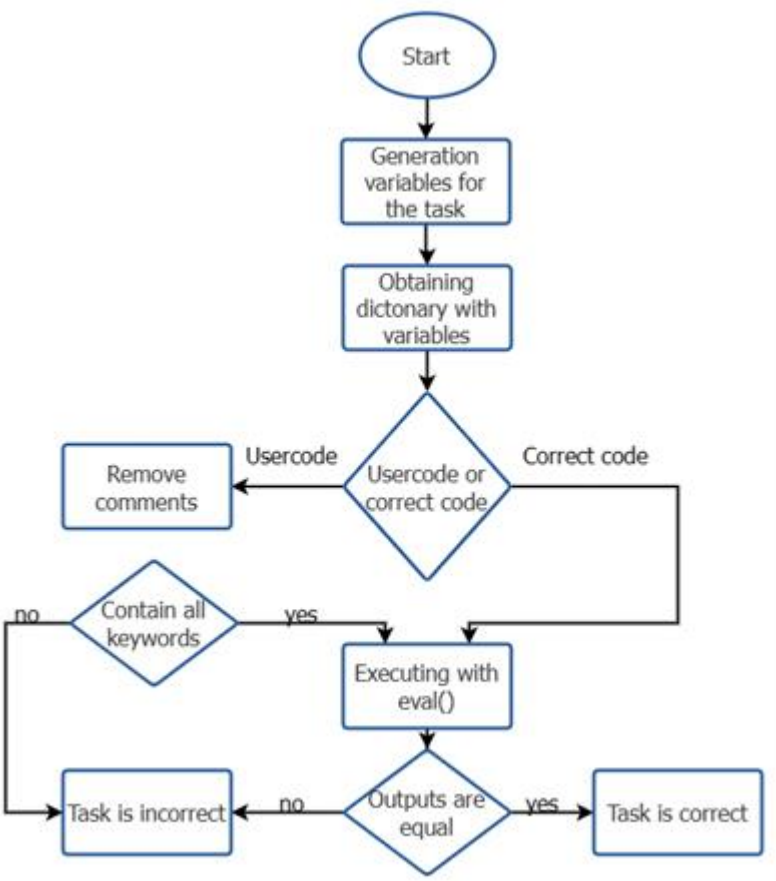

Figure 3: Block-scheme of task checking process

3. Development results

An interactive web application was developed on Django to teach programming with its own unique implementation of the Python code interpreter and with its own flexible task validation system. The creation of an interactive web application meticulously developed on the Django framework marks a significant milestone in the realm of programming education. This innovative platform boasts its own distinctive implementation of the Python code interpreter, setting it apart as a pioneering tool for learners and educators. In addition, the application features a highly adaptable and flexible task validation system, adding a layer of customization that caters to the individualized learning needs of users. This web application is a testament to our commitment to redefine the boundaries of programming education. Its unique Python code interpreter offers users a dynamic and immersive learning experience where they can experiment, iterate, and explore coding concepts in real time. The flexibility of the task validation system empowers educators to craft tasks that align with their pedagogical objectives while ensuring a rigorous and standards-based approach for assessing user progress.

In this pivotal screenshot, we unveiled a programming task that showcases the ingenuity of our web application. The task was generated using a dynamic array of random variables, each carefully configured to provide a unique and diverse set of challenges for learners.

The structure of this programming task is a testament to the versatility and adaptability of the proposed platform. The random variables injected into the task prompted users to apply their coding skills to a real-world scenario, mirroring the unpredictability of coding in professional settings.

This task is a testament to our commitment to offer learners a broad and challenging educational experience. This finding reinforces the importance of practicality and adaptability in programming education. The ability to generate dynamic tasks with random variables ensures that users are continually engaged and equipped to address the ever-evolving landscape of coding challenges. This screenshot underscores the dynamic nature of our web application and its capacity to prepare learners for real-world programming complexities.

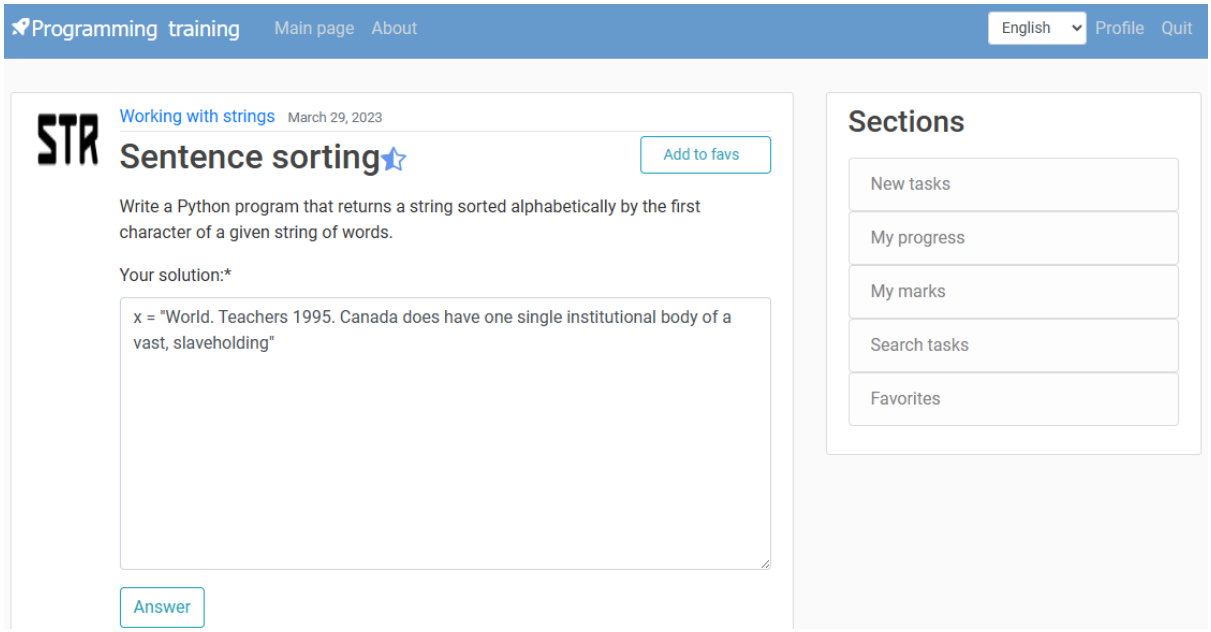

Figure 4: Programming Task with random variable

In Figure 5, we offer a glimpse of the critical aspect of our web application's assessment process. The system meticulously checks the task answers to ensure that they meet specific keyword requirements.

In this instance, the assessment mechanism identified that not all the required keywords have been used in the submitted answer. This meticulous examination goes beyond the surface and dives deeply into the completeness of the user's solution.

It is a testament to our commitment to maintaining educational standards and ensuring that learners are not only provided with accurate feedback but also guided towards a comprehensive understanding of coding concepts. The screenshot in Figure 5 underscores the precision and depth of our assessment mechanisms, allowing educators and learners to fine-tune their approach to tasks and achieve a more complete grasp of programming principles.

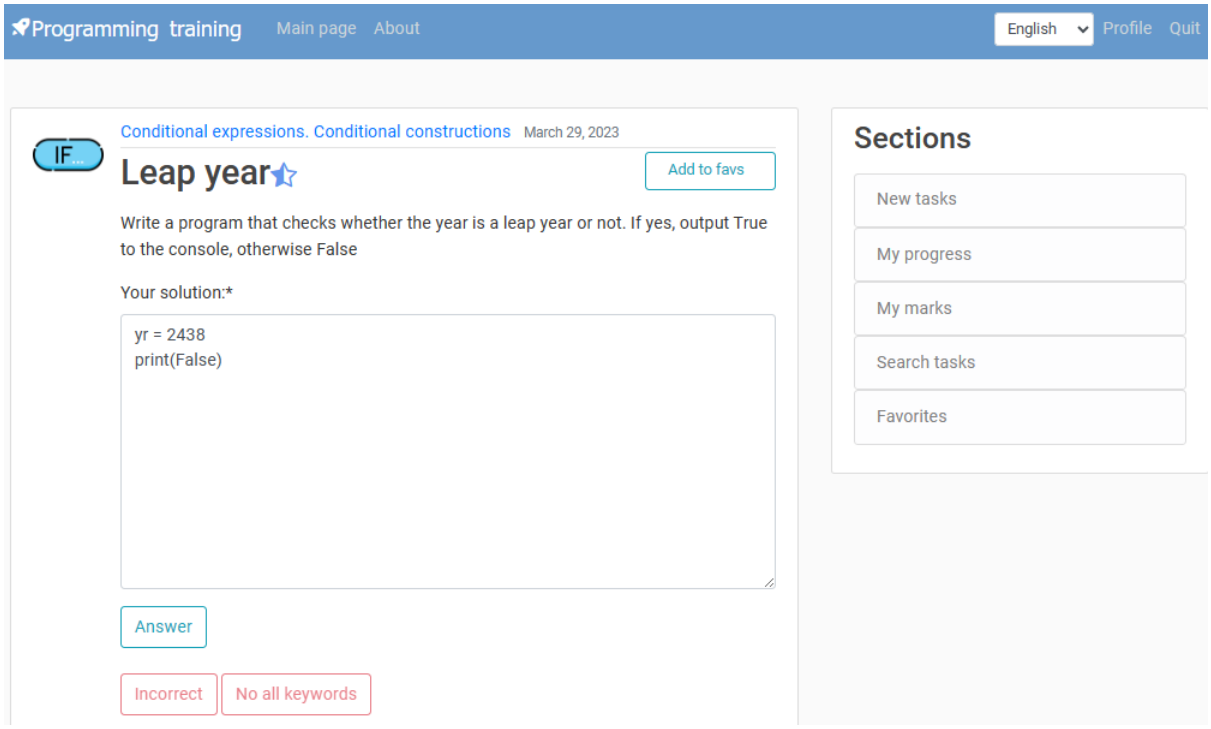

Figure 5: Assessing Keyword Completeness in Task Answers

One of the main components of our dynamic graphs is their division into programming topics. This allows students to easily assess their skills in each specific area and focus their efforts on improving their weaknesses. For example, students can see how they are produced when working with arrays compared to when working with conditional expressions.

Figure 6: Dynamics in solving programming tasks by topic

One of the key components of our methodology for improving programming learning is dynamic graphs, which provide students with the opportunity to observe and analyze their progress in solving programming problems. These graphs are divided according to various criteria, such as topics and difficulty levels, and serve as valuable tools to stimulate active learning and increase student motivation.

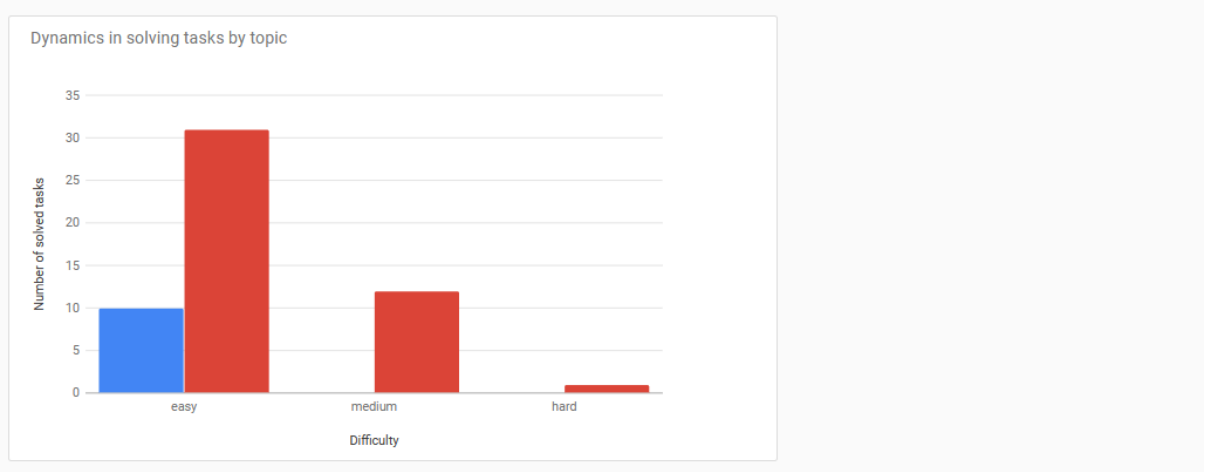

Figure 7: Dynamics in solving programming tasks by difficulty

These dynamic graphs not only provide a visual representation of student progress but also allow students to make informed decisions about where they need to improve their learning efforts. They also motivated students by providing visual confirmation that they really improved their programming skills. Thus, our methods of visualizing student progress, divided by topic and complexity, play an important role in enriching students' educational experiences and contribute to more effective programming learning.

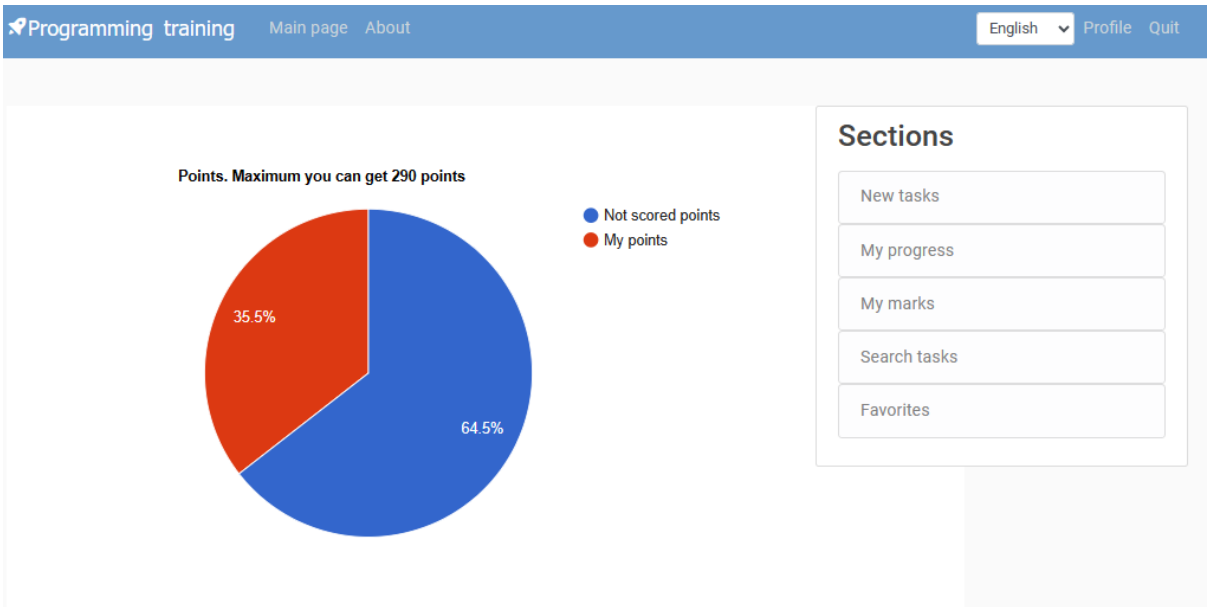

Figure 8: Visualization of a pie chart for solved tasks

The analysis of graphs and diagrams presented in this study sheds light on the importance of an open and systematic view of the results of solved tasks in the field of programming training. These visual aids enrich our understanding of the learning processes and confirm the effectiveness of the developed methods and tools.

The graphs and diagrams presented in this paper reflect the dynamics of learning and students' level of success in solving various tasks. They are an integral part of educational analytics and provide valuable insight into the learning process.

Visual representations of data allow researchers and teachers to track changes in student skills over time. Graphs allow us to observe how students improve their skills, depending on the complexity of the tasks and topics. These data make it possible to optimize the educational process and identify successful teaching methods.

In addition, graphs and diagrams emphasize the importance of an open view of the results. They provide indisputable evidence on how learning changes students' level of knowledge and skills. This visual data also serves as a feedback tool, allowing students to evaluate their progress and learn from mistakes.

Thus, the graphs and diagrams in this paper not only represent statistical data but also open an important window into the educational process. They give shape to abstract concepts and allow informed decisions to be made based on observed facts. These visual representations have become reliable companions in the field of programming education, enriching the experience of students and researchers.

4. Use cases and specific outcomes

To evaluate the effectiveness of our web application platform for learning Python, a survey was conducted among groups of participants. The group of participants is mainly represented by students of the first and second year of bachelor's degree who have little experience in programming but want to hone their practical programming skills. The purpose of the survey was to evaluate key characteristics, such as simplicity, content quality and visualization, using a percentage scale that allows participants to rate each dimension on a scale from 0 to 100. 90 people with different levels of programming experience took part in the survey. Participants were asked to evaluate these aspects based on their experience with the web application for 1-2 days. The key characteristics assessed during the survey included the next:

- Simplicity of the user interface. The user interface was evaluated for its simplicity, focusing on ease of navigation and the intuitiveness of design to ensure a user-friendly experience. The simplicity of the interface and its convenience allows the student to focus on learning and solving programming problems.
- The quality of content. The quality of practical tasks that were presented in the educational web application is evaluated, how much they reveal programming concepts in the opinion of students and help to develop in the study of programming.
- Interactivity. Interactivity in a web application is very important, there are functions for searching for tasks, saving to bookmarks, logging in to the system. The quality of all this is evaluated in this criterion.
- Visualization. Visualization is important to facilitate learning by making abstract concepts more tangible. The visually attractive interface not only creates a pleasant learning environment but also helps to concentrate on solving programming tasks.
- Feedback and assessment. The mechanisms of the web application, such as sending the written code for verification, offering the correct answer in case of a large number of attempts spent, the comments section where students can ask questions is evaluated according to this criterion.
- Quantity of content. The quantity of content is pivotal enough in providing a rich and dynamic learning experience, ensuring that the educational web application remains robust, adaptable, and capable of meeting the diverse needs of its user base.
- Performance of the platform. Participants provided feedback on the platform's overall performance, including its speed, responsiveness, and reliability in delivering content and interactive elements. The indicator of this criterion shows the effectiveness of the applied methods of code generation and dynamic code execution.

- Convenience of progress checking. The effectiveness of progress tracking and reporting features will be examined to ensure users can easily monitor their advancement, fostering motivation and a sense of accomplishment.

The survey results were analyzed to identify trends and patterns. Visual representations in the form of a diagram were used to present quantitative results. In addition, with the help of open questions, high-quality feedback was collected, allowing to receive detailed information from the participants. The average values of the survey results are shown in Figure 9.

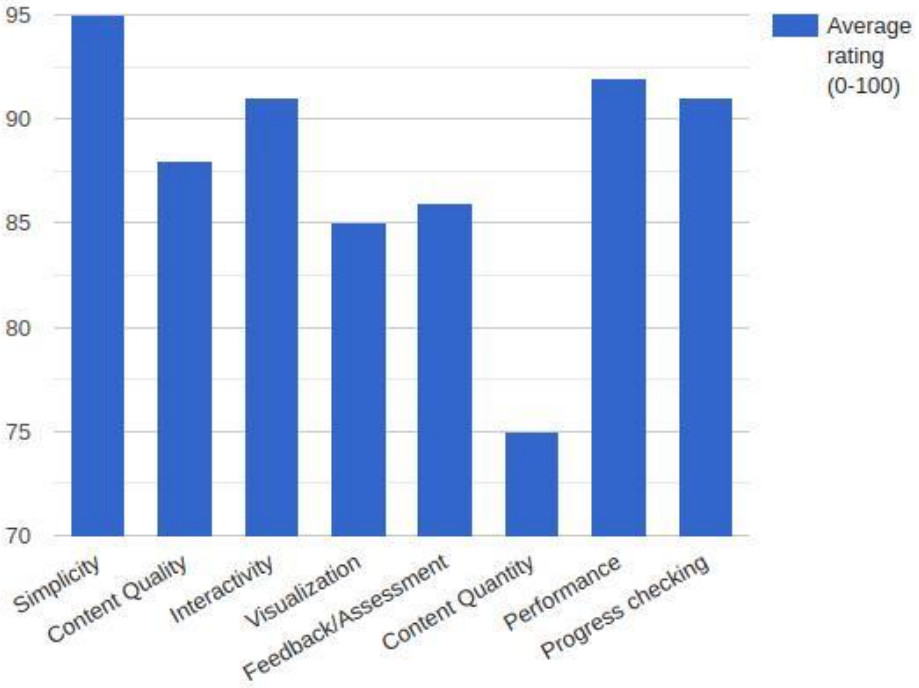

Figure 9: Average results of the survey

Specific use cases were obtained from the survey data, demonstrating cases where the platform was different, or areas in which improvements were proposed. During the survey, training results were obtained, shedding light on how users perceive the strengths and weaknesses of the platform. Correlations and patterns in the data were studied in connection with the overall goals of the educational platform. The students highly appreciated the simplicity, interactivity, speed and checking of progress, which confirms the effectiveness of the proposed methodology for developing a web application.

Survey results were reviewed, which provided valuable information for future improvements. Among them, it is worth noting the relatively low result in the amount of content. This information is very important for improving the developed platform of the educational web application, bringing user ratings in line with the goals of the educational initiative.

5. Conclusions

In this paper, we embarked on a journey to address a prevalent challenge in programming education: the need for an interactive and effective learning tool that bridges the gap between theory and practice. Our research introduces a solution in the form of an innovative web application, a digital ecosystem that not only addresses the shortcomings of existing educational apps, but also augments the learning process for programming enthusiasts.

As demonstrated in this study, our methodology for task correctness checking offers a robust mechanism for assessing the quality and accuracy of user-generated code. By implementing regular expressions (regex) to validate keywords, we ensure that the user solutions adhere to

predefined requirements. The integration of these methods provides a comprehensive evaluation system that is instrumental in maintaining the educational standards of our web applications.

The survey results provide a comprehensive view of the user experience, providing specific information about the strengths of the platform, which confirms the effectiveness of the proposed methodology and design of the web application. Also giving information about areas for development.

As we bring this paper closer, we emphasize the significance of our work in reshaping the landscape of educational technology. Our web application is a testament to the fusion of technology and pedagogy and offers a dynamic, interactive, and adaptive learning environment for coding enthusiasts. It is our hope that this contribution will not only bridge the gap in programming education but also inspire further innovation in the field of educational apps, ultimately empowering learners on their journey to programming excellence.

6. References

[1] J. Zosh, S. Lytle, R. Golinkoff, and K. Hirsh-Pasek, "Putting the Education Back in Educational Apps: How Content and Context Interact to Promote Learning," 2017, pp. 259–282. doi: 10.1007/978-3-319-45102-2_17.

[2] C. Fernández, M. A. Vicente, M. M. Galotto, M. Martinez-Rach, and A. Pomares, "Improving student engagement on programming using app development with Android devices," Comput. Appl. Eng. Educ., vol. 25, no. 5, pp. 659–668, 2017, doi: 10.1002/cae.21827.

[3] D. Zhao, C. H. Muntean, A. E. Chis, G. Rozinaj, and G.-M. Muntean, "Game-Based Learning: Enhancing Student Experience, Knowledge Gain, and Usability in Higher Education Programming Courses," IEEE Trans. Educ., vol. 65, no. 4, pp. 502–513, Nov. 2022, doi: 10.1109/TE.2021.3136914.

[4] Z. Zhan, L. He, Y. Tong, X. Liang, S. Guo, and X. Lan, "The effectiveness of gamification in programming education: Evidence from a meta-analysis," Comput. Educ. Artif. Intell., vol. 3, p. 100096, Jan. 2022, doi: 10.1016/j.caeai.2022.100096.

[5] S. Azmi, N. A. Iahad, and N. Ahmad, "Attracting students' engagement in programming courses with gamification," in 2016 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), Oct. 2016, pp. 112–115. doi: 10.1109/IC3e.2016.8009050.

[6] O. Chernikova, N. Heitzmann, M. Stadler, D. Holzberger, T. Seidel, and F. Fischer, "Simulation-Based Learning in Higher Education: A Meta-Analysis," Rev. Educ. Res., vol. 90, no. 4, pp. 499–541, Aug. 2020, doi: 10.3102/0034654320933544.

[7] A. Nguyen, T. Tuunanen, L. Gardner, and D. Sheridan, "Design principles for learning analytics information systems in higher education," Eur. J. Inf. Syst., vol. 30, no. 5, pp. 541–568, Sep. 2021, doi: 10.1080/0960085X.2020.1816144.

[8] R. Suzuki, J. Kato, and K. Yatani, "ClassCode: An Interactive Teaching and Learning Environment for Programming Education in Classrooms." arXiv, Jan. 22, 2020. doi: 10.48550/arXiv.2001.08194.

[9] J. C. Farah, S. Ingram, and D. Gillet, Supporting Developers in Creating Web Apps for Education via an App Development Framework. 2022. doi: 10.4995/HEAd22.2022.14587.

[10] S. Doi and Y. Imai, "Development of Web Application to Support Program Learning of Python and Ruby with Error Accumulation and Analysis Facility," Fourth Int. Conf. Electron. Softw. Sci. ICESS2018, Jan. 2018, Accessed: Oct. 29, 2023. [Online]. Available: https://www.academia.edu/38133220/Development_of_Web_Application_to_Support_Program_Learning_of_Python_and_Ruby_with_Error_Accumulation_and_Analysis_Facility

[11] B. Isong, "A Methodology for Teaching Computer Programming: first year students' perspective," Int. J. Mod. Educ. Comput. Sci., vol. 6, pp. 15–21, Sep. 2014, doi: 10.5815/ijmecs.2014.09.03.