The automated system Steach.Space designed for Programming teaching

A Kizyanov¹, D Luchaninov¹, R Bazhenov¹, A Pronin², N Tabachuk³ and A Samorukov⁴

¹ Sholom-Aleichem Priamursky State University, 70a Shirokaya str., Birobidzhan, 679015, Russia
² Ekaterinburg Academy of Contemporary Art, 3 Kultury str., Ekaterinburg, 620012, Russia
³ Pacific National University, 136 Tihookeanskaya str., Khabarovsk, 680000, Russia
⁴ Orenburg State University, 13 Prospect Pobedy, Orenburg, 460018, Russia

E-mail: r-i-bazhenov@yandex.ru

Abstract. The research aims at designing and verifying an automated system for online learning in Programming languages. Such systems were designed to provide students’ independent theoretical training and exercising within in-person learning and are able to operate on their own. Nginx, Docker, Gunicorn, Django, Python, PostgreSQL, HTML5, and CSS Framework were applied. Testing was held at Sholom-Aleichem Priamursky State University and Pacific National University in the 2019-2020 academic year. Training in C++ and Python was arranged for 30 students for the research efficiency checking. The researchers found out the system used could make Programming learning more efficient. The results showed that it was improved by 30% in an experimental group and by 17% in a control group. The survey data used defined that the participants of the experiment (students, lecturers, experts) appreciated the software solution. The designed system Steach.space has a user-friendly interface, contains modules for developing integrated/built-in courses, providing teaching aids and tests, computer-assisted verification of assignments in Programming, and observing statistics/history on student learning. The software can be used for teaching programming languages, as well as other subject domains.

1. Introduction

Every year, more and more areas gradually adopt electronic platforms. Education is also not an exception. It was especially noticeable in 2020 because of the pandemic. All educational institutions had to use a distance learning format which caused a huge demand for distance learning software.

There are systems for distance learning in primary schools [1] and in higher education institutions [2]. The adaptability of such training is actively discussed in the professional environment [3], research is conducted in computer-aided education [4], and video is regarded as a form of distance learning [5]. Researchers apply the online system incorporated in individual studies [6], in corporate environments [7], in teaching private lessons [8]. There are programs of search and development in 3D-oriented training software [9, 10].

The purpose of the research is to develop and test an automated web-based system for teaching Programming.
2. Methods
After monitoring the market for automated programming training software, such as Udemy (https://www.udemy.com/), Smotriuchis (https://smotriuchis.ru/), Stepik (https://stepik.org/), Hexlet (https://en.hexlet.io/), Geekbrains (https://geekbrains.ru/), Coursera (https://www.coursera.org/), Codecademy (https://www.codecademy.com/), the researchers set the requirements. They include a convenient and simple interface, free certificates available, the Russian language support, the ability to run one’s course, and automatic homework check, preparation and automated verification of Programming tasks, collecting and presenting statistics/history on student learning.

The authors used Nginx, Docker, Gunicorn, Django, Python 3.8, PostgreSQL 10, HTML5, and CSS Framework technologies to start the designed software.

Piloting and assessment of the system were carried out during the 2019-2020 academic year at Sholom-Aleichem Priamursky State University and Pacific National University. The study involved 60 first-and second-year students majoring in 09.03.02 Information systems and technologies. The participants were divided into two groups (experimental and control) of 30 people each. When selecting students, the developers took into account the level of student skills. Those students who ever participated in various Programming academic competitions were not invited to that study for reasons of possible peak ratios. Programming courses in C++ and Python were selected for the experiment. The students in the experimental group were taught online using an automated solution verification system.

The students in the control group were involved in online training with a supervised solution check. Knowledge and skills were tested before and after training, and a one-hundred-point scale was applied for assessment.

To test availability to other subject areas, the system was used in Yekaterinburg Academy of Contemporary Art, Orenburg State University, where the courses were developed not in Programming. The experiment involved 17 and 19 students, respectively.

The convenient and qualitative operation of the system was assessed by a survey of 66 students and 10 lecturers.

3. Results and discussion
The review of the task with developing Steach.space system allowed the scholars to differentiate three function modules.

The system combines several user roles of an admin, a trainer, and a student.

- Authorization module/unit (registration, user authorization, and completing a personal profile).
- Course management module/unit (designing and editing courses, editing sections, tasks, tests).
- Performance module/unit (checking tasks submitted by students, calculating student assessment statistics).

To store the required information, the authors designed a relational database consisting of the following tables:

- User - information about registered users.
- UserGroup - groups of students.
- Course - courses and groups that can attend these courses.
- Section - sections that are included in the course.
- Exercise-tasks that are components of the section.
- Try - all attempts to solve problems.
- Result - task checking results.
- Tests-models of final tests at the end of the course.
- Test Body-test specification.
- Options - test choice points.
• Test Result - test performance.

The authors selected appropriate technologies for programming the platform.

The authors developed a web-based Steach.space system (http://steach.space/) using programming languages and technologies such as Django, HTML, CSS, JavaScript, and PostgreSQL. The instruments listed are free of charge.

One of the problems was code isolation. To ensure proper operation, it is required the code to be executed inside and checked with tests and not to harm the basic system. The solution is possible using Docker containers. They isolate a single process in the system and prevent it from interacting with the system core/heart. It starts the necessary operations and completes the process. Docker containers also have an advantage over virtual machines in speed, because while the operating system in the virtual environment starts and takes a long time, the process starts in few seconds and starts performing tasks immediately. Finally, the authors developed several Docker containers for each language, since there are compiled languages, interpreted languages, and what is more, some of them need special libraries. List of supported languages is composed of Python, C#, Go, Java, PascalABC.NET, C++. Actually, it is a standard set of popular languages, except PascalABC.NET, which is added because it is taught at Russian secondary school curriculum.

The system interface is aimed at Russian-speaking users and is friendly to use, so even unadvanced computer users do not seem to have difficulty in dealing with it (figure 1).

![Figure 1. Steach.space System Interface.](image)

The trainers can create their own courses; fill them with teaching aids, assignments with guidelines and tests for checking (figures 2, 3, 4). Such a user has a personal profile that represents a list of his courses and groups.

![Figure 2. Test Form filled out.](image)
According to the completed tasks of users enrolled in the course, the trainer generates statistics for the group, which renders the performance of each student (figure 5).

Each completed task sending is stored in the system, which allows one to realize what is causing the problem (figure 6).
The designed system was tested in the 2019-2020 academic year among 60 students participating in the experiment. They studied Programming courses in C++ and Python. When selecting students, their skills level was considered. The participants of various Programming academic contests were dropped out. The members of the experimental group (30 people) used the designed automated system, while the control group (30 people) used online learning with a supervisor checking solutions. The one-hundred-point scale was used when assessing knowledge and skills. Students’ achievements were assessed at the beginning and the end of the course studied. According to the test results, all the students were divided into three groups according to their skill level: low (less than 33 points), medium (34-66 points), and high (more than 67 points). The results of the study are shown in figure 6.

The results prove that the improvement in the experimental group was 30 per cent, and 17 per cent in the control group. Besides, there should be noted a significantly reduced number of students in the Low group, which stand for a positive motivational aspect in the approach to online Programming learning in the designed system.

To test adaptability to other subject areas in the system, teachers of the Yekaterinburg Academy of Contemporary Art and Orenburg State University developed non-Programming courses. The experiment involved 17 and 19 students, respectively.

Figure 6. Feedback to student performed tasks.

After testing, 66 students from different universities were asked to take a survey. The results of it could outline the benefits and drawbacks of the developed solution. The questions were the following:

- How relevant do you think this system is for you?
- Assess the complexity of this system.
- Would you like this system to be part of your lesson?
- Would you like to use similar resources for other subjects?
- Rate the system interface on a scale of 1 to 5.

The first and second questions were rated on a scale of one to ten, the third and fourth were answered either Yes or No.

Figure 7. Study findings in the control (a) and experimental (b) groups.
The average score for the first question was 8.9 points. The average value for the second question was 5.8 points. 80% of students answered ‘Yes’, the other 20% of them answered ‘No’ to the third question. The difference of responses to the fourth question was presented as follows: 80% answered ‘Yes’, 20% answered ‘No’. The fifth question about the system interface was assessed by an average score of 4.3 points.

Analyzing the questionnaire data, one can conclude that the designed system is relevant for students and its usage in learning is not difficult.

The system was also evaluated by 10 teachers from different universities. The questionnaire asked to answer the following questions:

- Assess the system interface on a five-point scale.
- Assess the opportunity of using the system in teaching your subjects on a five-point scale.
- Rate the convenience of the developed test tasks on a five-point scale.
- Rate the convenience of the course development on a five-point scale.
- Rate the convenience of presenting statistics for students on a five-point scale.

The average values were calculated for all the answers to the questions: question #1 - 4.5, question #2 - 4.2, question #3 - 4.3, question #4 - 4.2, question #5 - 4.1.

As a result, one can say that the teachers highly appreciated the designed system. The average values turned out to be more than 4 everywhere.

4. Conclusion

The developed Steach.space system has a user-friendly interface, contains modules for developing integrated/built-in courses, providing teaching aids and tests, automated check of Programming tasks, and monitoring statistics on student learning. One can use it to teach Python, C#, Go, Java, PascalABC.NET, C++ programming languages.

The Steach.space system can be used for teaching Programming both in classroom and distance learning. It also allows one to develop education courses in other subject areas.

In the future, the authors are going to enlarge the list of supported languages, add a kind of student grade record, and gamification elements.

References

[1] Burdina G M, Krapotkina I E and Nasyrova L G 2019 Int. Journal of Instruction 12 1-16
[2] Lepellere M A, Cristea I and Gubiani D 2019 Models and Theories in Social Systems 295-309
[3] Syed M R 2020 Simulation 348
[4]UGHUR S and Kurubacak G 2019 Handbook of Research on Challenges and Opportunities in Launching a Technology-Driven International University 338-68
[5] Dellagiacoma D, Busetta P, Gabbasov A, Perini A and Susi A 2020 Authoring interactive videos for e-learning: the ELEVATE tool suite Int. Conf. in Methodologies and intelligent Systems for Technology Enhanced Learning (Cham: Springer) p 127-36
[6] Abakumova I, Bakaeva I, Grishina A and Dyakova E 2019 Int. Journal of Cognitive Research in Science, Engineering and Education 7 85
[7] Rabiman R, Nurtanto M and Khalifah N 2020 Online Submission 9 1059-63
[8] Hibbi F Z and Abdoun O 2020 Recent Advances in Mathematics and Technology 141-50
[9] Laffey J M, Sadler T D, Goggins S P, Griffin J and Babiuch R N 2019 Virtual Reality in Education: Breakthroughs in Research and Practice 623-43
[10] Hwang G J and Chen P Y 2019 Interactive Learning Environments 1-16