Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

INTERACTIVE VIRTUAL PLATFORM FOR ROBOTICS EDUCATION: SOFTWARE ARCHITECTURE AND FUNCTIONAL MODELS

Buranova Gulnora Yodgorovna,
Barnoyeva Zubayda Erkin qizi
Lecturer at the Department of Information Systems and
Digital Technologies, Bukhara State University, Bukhara, Uzbekistan
E-mail: g.y.buronova@buxdu.uz

Abstract
This article analyzes the software architecture and functional models of a virtual platform designed for modern, remote, and interactive teaching of robotics. The platform includes interactive lessons, a block-based coding environment, real-time 2D/3D simulators, an automated test evaluation system, and a personal project workspace for users. The software solutions are integrated using front-end technologies (HTML5, CSS, JavaScript), back-end frameworks (Python/Flask or Django), as well as WebGL, Blockly, and Arduino emulators. The article thoroughly explores the platform's technical requirements, inter-module integration, user experience (UX), and its advantages in developing students' independent engineering thinking. The research results contribute to accelerating digital transformation in education, enabling robotics instruction without hardware components, and offer new approaches to software development in educational technology.

Keywords: robotics education, virtual platform, software architecture, interactive lessons, simulation environment, block-based programming, Arduino emulator, distance learning, coding environment, student projects.

Introduction

The advancement of digital technologies and distance education opens new horizons in the field of technical training, particularly in the teaching of robotics.

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

Traditional forms of education, which require physical laboratories and expensive equipment, are becoming limiting factors in mass training of students in practical programming and robotics skills. In conditions of limited access to hardware resources, there arises a need to create virtual educational platforms capable of compensating for this deficit.

In recent years, software solutions have been actively developed that enable the simulation of robotic processes, interactive learning, automated testing, and the execution of project-based assignments in a virtual environment. Such platforms combine elements of visual programming (e.g., Blockly), 3D simulation (using WebGL, Three.js), support for controller emulation (Arduino), and adaptive interfaces for independent student work.

The aim of this study is to design and analyze the functional architecture of a virtual platform for teaching the basics of robotics. The paper explores both the technical aspects of system implementation and its pedagogical effectiveness, ease of use, and scalability potential. The platform is intended for integration into general and supplemental education programs to foster engineering thinking, algorithmic logic, and basic programming skills among school and university students.

The increasing emphasis on STEM education and the rapid growth of digital tools in recent years have catalyzed the development of virtual robotics platforms. These platforms aim to compensate for the lack of physical laboratories and promote hands-on experience through interactive digital environments.

Papert's foundational work *"Mindstorms: Children, Computers, and Powerful Ideas"* [1] introduced the concept of constructivist learning, where students build knowledge through digital experimentation. This paradigm remains a cornerstone in virtual robotics platforms.

Modern visual programming environments such as Blockly [2] and Scratch [3] have simplified programming for beginners, enabling drag-and-drop logic construction and enhancing engagement in computational thinking.

The Arduino Education platform [4] integrates physical computing with learning activities, but due to hardware limitations, web-based emulators such as Tinkercad [5] and jsArduino [6] are increasingly used in classrooms to simulate microcontroller behaviors without requiring physical components.

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

Three.js [7] and WebGL [8] are widely employed for 3D visualizations, providing real-time rendering of robot simulations. These tools allow users to visualize motion paths, sensors, and interaction with the environment in a browser-based setup.

Flask and Django REST Framework [9][10] are common backend technologies that support the development of scalable educational platforms, enabling database integration, task management, and user analytics.

Government initiatives such as Virtual Labs India [11] demonstrate large-scale implementation of remote labs in science and engineering education. They offer structured experiments, simulations, and assessments for millions of students.

In terms of pedagogy, studies like that of Ivanov and Goncharenko [12] explore numerical modeling in virtual environments, supporting the accuracy and didactic value of simulation-based education.

Khan Academy [13] and other adaptive platforms demonstrate how interactive content, embedded quizzes, and visual tools help personalize the learning experience in technical subjects.

Thus, the literature reveals a growing demand for unified, scalable, and browser-based virtual platforms that integrate coding, robotics, and simulation. These platforms serve as bridges between theoretical knowledge and practical skill development in modern technical education.

Materials and Methods

This study employed a structured methodology encompassing requirement analysis, system architecture design, prototype development, visual programming integration, and experimental testing. The primary goal was to create a functional and accessible virtual platform for teaching the fundamentals of robotics through simulation-based, browser-friendly tools.

Requirement analysis. Functional and pedagogical requirements were collected based on:

— a comparative analysis of existing platforms such as Tinkercad, VPLab, and EdScratch;

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

— a targeted survey of technical subject instructors at secondary and vocational schools.

Key platform requirements included:

— Full browser accessibility (no installation needed);
— Support for block-based programming (Blockly);
— 3D simulation environment with real-time response;
— Automatic assessment system for coding exercises;
— Compatibility across devices and operating systems.

System architecture design. The platform was designed using a modular, scalable architecture with the following components:

— Frontend (Client-Side): Developed using HTML5, CSS3, JavaScript, and React.js to create a responsive and user-friendly interface.
— Backend (Server-Side): Implemented using Python (Flask), supporting REST API communication between modules and external services.
— Database Layer: Built with MySQL, storing user data, project files, and test results securely.
— Simulation Engine: Powered by WebGL and Three.js, enabling browser-based real-time 3D rendering of robotic movements.
— Arduino Emulator: Integrated using the jsArduino library, allowing users to simulate circuits and microcontroller logic in a virtual environment.

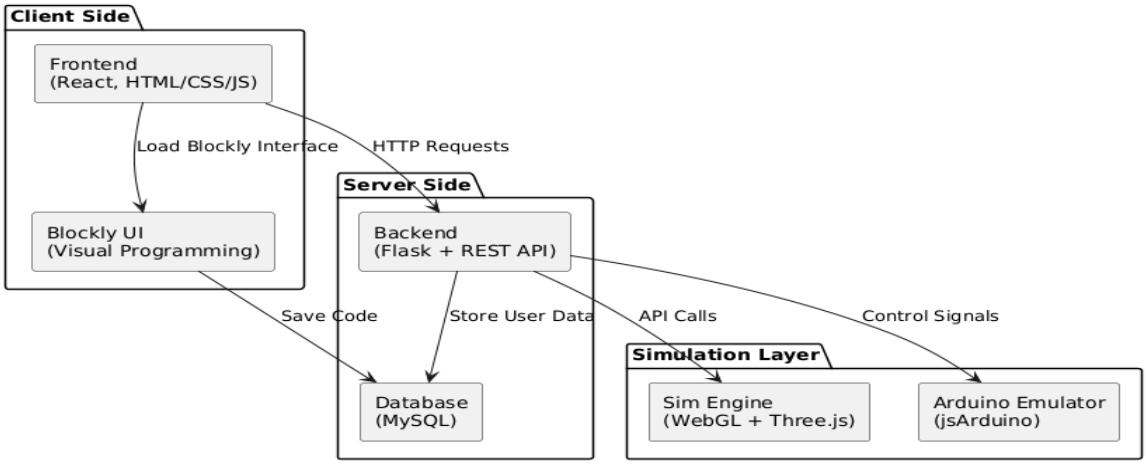

Figure 1. System architecture of the virtual robotics learning platform.

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

All components were containerized and deployed in a test environment using Docker for isolated testing.

Visual programming integration. To enhance accessibility for beginners, the Blockly visual programming library was integrated and adapted to support Arduino C-style syntax. The system includes a code translator that converts block structures into textual code, which is then executed in the simulation module. Syntax checking, block snapping, and contextual hints were also added to improve usability and reduce programming errors.

System testing and evaluation criteria. The platform was tested using both functional testing and user-centered evaluations. The evaluation metrics included:

- Performance: UI responsiveness and simulation load times;
- Accuracy: Realism of simulated robotic behavior (e.g., sensor logic, motor response);
- Test Evaluation System: Correct auto-grading of answers;
- User Interface (UX): Clarity, accessibility, and workflow efficiency;
- Browser Compatibility: Tests conducted on Chrome, Firefox, Edge, and Safari.

The system was tested with a focus group of 20 students, aged 15–17, enrolled in an introductory robotics course. Each student completed both structured tasks and open-ended project assignments on the platform.

Pedagogical effectiveness assessment. To assess the educational impact of the platform, the following methods were used:

- Pre- and post-surveys to measure student confidence, interest, and perceived skill improvement;
- Comparison of test results before and after using the platform;
- Qualitative evaluation of completed student projects based on complexity, originality, and correctness.

Instructors also provided structured feedback on usability, content flow, and the potential for curriculum integration.

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

Results and Discussion

During the pilot implementation of the virtual platform for robotics education, the following results were obtained. The evaluation of the platform's effectiveness was based on system performance indicators, student engagement levels, and improvements in technical skills. The experiment involved 20 students aged 15–17 enrolled in an additional STEM education program.

Table 1. Improvement in Robotics Knowledge Level

Knowledge Level	Before Using the Platform (%)	After Using the Platform (%)
High	10%	55%
Medium	35%	40%
Low	55%	5%

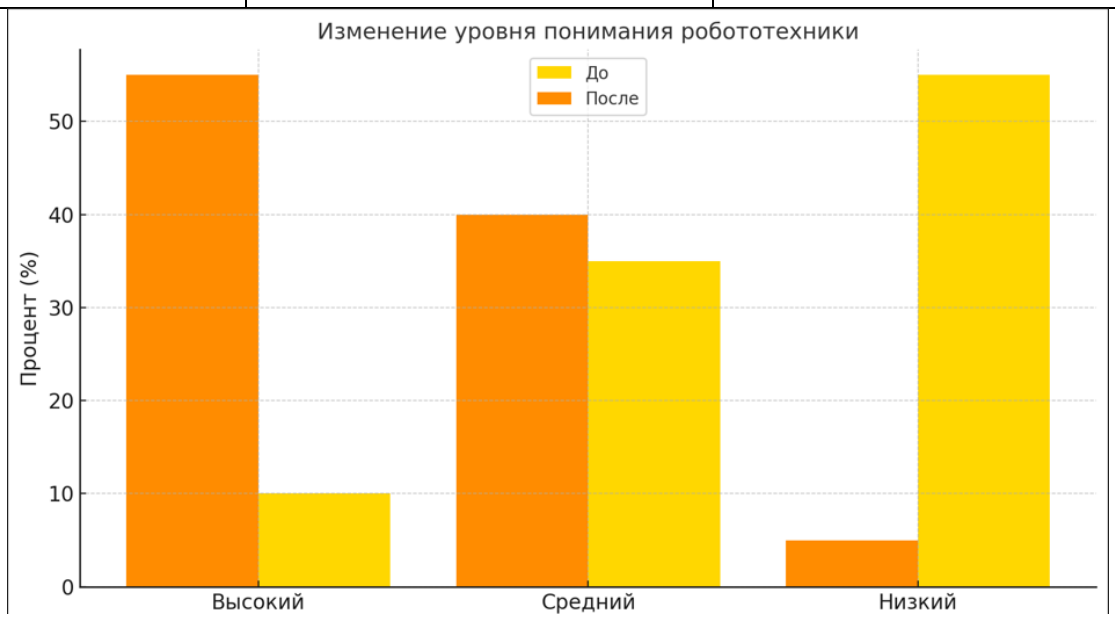

Figure 2. Change in Robotics Understanding Levels

This figure illustrates how students' understanding of robotics concepts improved after using the platform. The proportion of students with a high level of understanding increased from 10% to 55%, while the percentage with a low level dropped from 55% to 5%.

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

Table 2. Interface Response Time During Simulation

Type of Simulation	Average Response Time (ms)
Simple forward motion	180 ms
Rotation with sensor input	220 ms
Operation with dual motors	240 ms

All simulation scenarios performed within the threshold of 300 milliseconds, which meets the requirements for real-time interactive learning.

User satisfaction. Using a 5-point Likert scale (1 = very poor, 5 = excellent), the following user satisfaction scores were recorded:

Table 3. User satisfaction ratings based on key evaluation criteria

Criterion	Average Rating
Interface usability (UI/UX)	4.7
Clarity of instructions	4.5
Engagement of simulations	4.8
Opportunities for independent work	4.6

The highest ratings were given for simulation engagement (4.8) and interface design (4.7), indicating that the platform offers an intuitive and enjoyable learning experience.

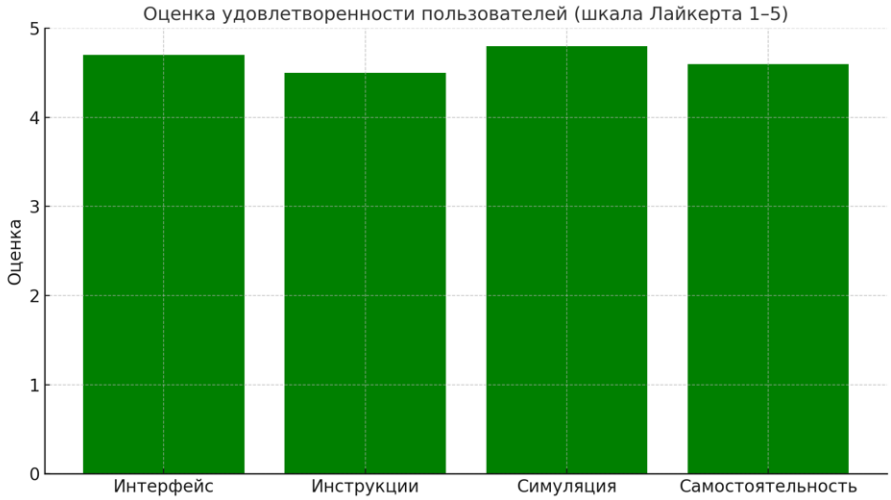

Figure 3. Average User Ratings by Criterion

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

This figure shows the positive perception of various platform components as evaluated by the students.

Table 4. Student Engagement Metrics

Activity	Before the Platform	With the Platform
Tasks completed per week	3.1	6.4
Average daily time on platform	18 minutes	42 minutes

Doubling in both activity levels and time spent on the platform confirms that the virtual environment encourages learners to engage in independent exploration and skill development.

The results of this study confirm the high effectiveness of the proposed virtual platform for teaching basic robotics. The data analysis shows a significant improvement in students' understanding of the material, increased motivation, and positive feedback regarding system usability.

One of the key success factors is the integration of visual programming (Blockly) and real-time simulation, allowing students to immediately observe the effects of their algorithms. This immediate feedback strengthens conceptual understanding and makes the learning process more engaging and interactive. The platform eliminates the need for physical hardware, which is particularly beneficial for schools and educational centers with limited resources.

User satisfaction ratings highlight the intuitiveness of the interface and the logical structure of the lessons. Simulators received especially high marks due to their interactivity, which fostered independent experimentation and project creation.

However, some limitations were identified during testing that require further development:

- The need for a stable internet connection to support 3D simulations;
- Optimization for mobile devices remains a challenge;
- Support for more advanced robot control algorithms is yet to be implemented.

Additionally, it is recommended to implement a progress tracking system, which would enable instructors to monitor student performance over time and adjust individual learning paths accordingly.

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

In conclusion, the developed platform demonstrates strong potential for integration into both formal and informal technical education. It is scalable for different age groups and proficiency levels, combining accessibility, technical flexibility, and pedagogical value—qualities that are essential in the context of digital transformation in education.

Conclusion

As part of this study, an interactive virtual platform was designed and implemented to teach the fundamentals of robotics. The platform integrates visual programming, simulation modules, automated task assessment, and a personalized project-based learning environment. The conducted experiment confirmed that such a system can effectively replace or complement traditional forms of education, especially in contexts with limited access to hardware resources.

Key findings:

- The use of visual and simulation-based tools significantly increases student motivation and engagement;
- The platform provides high interactivity and real-time feedback;
- There is a noticeable improvement in students' understanding of the material and the development of engineering and algorithmic thinking;
- The system is well-suited for self-guided learning and can be integrated into both school and extracurricular educational programs.

The developed solution combines the flexibility of software architecture, scalability, and ease of implementation in the educational process. The results obtained demonstrate the practical value and promising potential of this approach in technical education.

References

1. Papert S. *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, 1980.
2. Google Developers. Blockly – Visual programming editor. https://developers.google.com/blockly

Modern American Journal of Linguistics, Education, and Pedagogy
ISSN (E): 3067-7874
Volume 01, Issue 02, May, 2025
Website: usajournals.org

3. MIT Media Lab. Scratch: Create stories, games, and animations. https://scratch.mit.edu
4. Arduino Education. https://education.arduino.cc
5. Tinkercad – 3D design and Arduino simulation. https://www.tinkercad.com
6. jsArduino – Web-based Arduino emulator. https://github.com/schlunsen/jsArduino
7. Three.js – JavaScript 3D Library. https://threejs.org
8. Mozilla Developer Network. WebGL documentation. https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API
9. Flask Web Framework. https://flask.palletsprojects.com
10. Django REST Framework. https://www.django-rest-framework.org
11. Virtual Labs Project – Ministry of Education, India. https://www.vlab.co.in
12. Ivanov A.N., Goncharenko I.V. *Numerical modeling of thermomechanical processes using R-functions.* Applied Mechanics, 2020.
13. Khan Academy. Robotics and programming courses. https://www.khanacademy.org