Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
ASEE
Paper ID #47284

FORE: A Student-Centered Framework for Accessible Robotics Education through Simulation and Interactive Learning

Hossein Jamali, University of Nevada, Reno

Hossein Jamali is a PhD student at the University of Nevada, Reno, focusing on Human-Computer Interaction, Artificial Intelligence, and Cloud Computing. His research encompasses a range of interdisciplinary topics, including AI-driven recommender systems and resource management algorithms. Hossein has authored several peer-reviewed papers in leading conferences and journals and has actively contributed to advancing AI and optimization techniques. His work reflects a commitment to creating innovative, leading-edge, and transparent computing solutions that enhance user experience and promote inclusivity.

Andrew L DeBolt
Henry Dalton, University of Nevada, Reno

Henry is a senior in the Computer Science and Engineering Program at the University of Nevada, Reno and is set to graduate this Spring, 2025.

Jenavieve K Layosa, University of Nevada, Reno
Ilana Rose Macy
Ponkoj Chandra Shill, University of Nevada, Reno

PhD Student in Computer Science at University of Nevada, Reno.

Dr. David Feil-Seifer, University of Nevada, Reno

David Feil-Seifer is an Associate Professor in Computer Science & Engineering at the University of Nevada, Reno. His primary research interests are Socially Assistive Robotics (SAR), Human-Robot Interaction (HRI), and Engineering Education. His resear

Frederick C Harris, University of Nevada, Reno
Dr. Sergiu Dascalu, University of Nevada, Reno
Rui Wu, East Carolina University

Rui Wu received a Bachelor degree in Computer Science and Technology from Jilin University, China, in 2013. He then pursued his Master and Ph.D. degrees in Computer Science and Engineering at the University of Nevada, Reno, completing them in 2015 and 2018, respectively. Currently, Rui works as an assistant professor in the Department of Computer Science at East Carolina University, collaborating with geological and hydrological scientists to protect the ecological system. His primary research interests lie in machine learning and data visualization using AR/VR devices. Dr. Wu has actively contributed to several NSF and NIH funded projects, serving as both a Principal Investigator (PI) and Co-PI.

FORE: A Student-Centered Framework for Accessible Robotics Education through Simulation and Interactive Learning

Hossein Jamali, Andy L Debolt,
Henry L Dalton, Jenavieve K Layosa, Ilana R Macy, Ponkoj C Shill,
David J Feil-Seifer, Frederick C Harris, Jr., Sergiu M. Dascalu, Rui Wu*
Department of Computer Science and Engineering
University of Nevada, Reno
*East Carolina University**
{hjamali, adebolt, hdalton, jlayosa, imacy, pshill}@unr.edu
{dfseifer, fred.harris, dascalus}@cse.unr.edu, wur18@ecu.edu*

Abstract

Robotics education is often constrained by the high cost and limited accessibility of physical robots, which can hinder the learning experience for many students. To address this challenge, the Fundamentals of Robotics Education (FORE) project, part of a larger NSF-funded collaborative work, was developed to create an accessible and comprehensive online learning platform. FORE provides a student-centered approach to robotics education, featuring a robust code editor, real-time simulation, and interactive lessons.

This paper presents the architecture and implementation of the FORE platform, highlighting its key components, including the backend simulation using Gazebo and ROS2, a frontend visualizer built with Three.js, and the integration of a Python-based coding environment. We discuss the development process, the contributions of the student team, and the challenges encountered during the project.

The results demonstrate the platform's effectiveness in making robotics education more easily available. These findings originate from software testing and utilization by senior computer science students, as well as feedback from participants at the University of Nevada, Reno College of Engineering's annual Capstone Course Innovation Day. The platform allows students to gain hands-on experience without the need for physical hardware. Its adaptability enables it to serve a broad audience of undergraduate students, offering an encompassing and accessible solution for modern robotics education.

Keywords: Robotics Education, Simulation-based Learning, Interactive Learning Platforms, Computer Science Education, Educational Technology

1 Introduction

Robotics education has emerged as a critical component in modern STEM (Science, Technology, Engineering, and Mathematics) curricula due to the rapid advancements in automation, artificial intelligence, and their applications across healthcare, manufacturing, and logistics. As robotics becomes increasingly prevalent in everyday life, preparing the next generation of engineers and technologists with foundational robotics knowledge is more important than ever. Robotics education not only helps students grasp mechanical, electrical, and computer engineering principles but also enhances problem-solving and critical-thinking skills [1].

Traditional robotics education relies heavily on physical robots and hardware, which introduces several barriers to accessibility. Robotics kits such as LEGO Mindstorms and VEX Robotics provide hands-on experience, yet they require significant financial investment from educational institutions [2, 3]. This poses a challenge, especially for underfunded schools and universities, where budget constraints limit the availability of such resources. Additionally, the complexity involved in setting up and maintaining robotics labs often discourages educators from fully embracing robotics education [4].

Furthermore, physical robots come with logistical limitations. Class sizes can make it difficult to ensure that every student has ample time and opportunity to interact with the robots, and the maintenance of these systems can be resource-intensive. Students who do not have access to these resources outside the classroom may find it difficult to continue learning at their own pace. This challenge is particularly prominent in remote and low-income areas, where physical infrastructure for robotics education may be lacking [5].

Given these barriers, educators and researchers have explored the use of simulation-based platforms as an alternative to physical robots for teaching robotics. Simulations allow students to experiment with robotic systems in virtual environments, which can replicate the behaviors of physical robots without the associated costs. One of the most prominent tools in this space is the Gazebo simulator, an open-source robotics simulation environment widely used in both academic and industrial research. Paired with the Robot Operating System (ROS), Gazebo enables realistic simulation of robotic systems, from simple mechanical movements to complex multi-agent interactions [6, 7].

Simulation-based learning offers numerous pedagogical advantages. It provides students with the ability to learn from their mistakes in a low-risk environment. Mistakes that would be costly or dangerous in a real-world setting can be easily corrected in a simulated environment, encouraging students to experiment and explore various robotic algorithms [8].

Recognizing the potential of simulations to enhance robotics education, the *Fundamentals of Robotics Education* (FORE) platform was developed as part of an NSF-funded initiative to provide an accessible, interactive, and student-centered approach to learning robotics. The FORE platform aims to democratize robotics education by removing the barriers associated with hardware costs and providing a comprehensive learning experience that integrates coding, real-time simulation, and assessment tools. It is designed for undergraduate students and offers a scalable solution that can be deployed in a variety of educational settings [8].

In the context of broader trends in educational technology, the FORE platform represents a shift towards blended learning models, where traditional in-person teaching is supplemented by online tools that provide immediate feedback and allow for self-paced study. Online platforms, such as Coursera, edX, and Khan Academy, have revolutionized the way students learn by making high-quality educational content available on demand. The FORE platform builds on this model by providing not just theoretical lessons, but practical, hands-on learning through its integrated simulation environment [9]. The platform's Python-based code editor, combined with ROS2 and Gazebo for simulation, enables students to apply programming concepts directly in a robotics context, bridging the gap between abstract coding exercises and real-world applications.

One of the key motivations behind the development of the FORE platform is the need for a flexible and scalable educational tool that can adapt to the varying needs of students. For beginners, the platform provides structured lessons that gradually introduce core robotics concepts such as motion control, sensor integration, and path planning. For more advanced students, the platform offers opportunities to explore more complex robotic algorithms and systems, including multi-robot coordination and artificial intelligence. This flexibility makes FORE a valuable resource for institutions looking to implement robotics education at scale without the need for extensive hardware investment [10].

In addition to its educational benefits, the FORE platform addresses the issue of accessibility in remote and low-resource environments. By removing the need for physical hardware, the platform allows students in geographically isolated areas or in schools with limited funding to access high-quality robotics education. The platform's cloud-based architecture ensures that students can access the platform from any location with internet access, making it a powerful tool for distance learning [11].

The increasing role of robotics in industry and society highlights the importance of robotics education as part of the STEM curriculum. The barriers presented by the cost and complexity of physical robots have led to the exploration of alternative solutions, such as simulation-based learning. The FORE platform represents a significant advancement in this area, providing an accessible, scalable, and comprehensive solution to teaching robotics. By integrating coding, simulation, and interactive lessons, the platform empowers students to develop the skills needed for success in the evolving field of robotics.

2 Related Work

Robotics education has long been a challenging field due to the high costs associated with physical robots, limited access to resources, and the complexity of integrating robotics curricula in schools and universities. However, several initiatives and technologies have emerged to overcome these barriers. This section provides a review of existing efforts in low-cost robotics platforms, simulation-based learning, and student-centered learning frameworks, highlighting their impact on education and areas where the *Fundamentals of Robotics Education* (FORE) platform seeks to contribute.

2.1 Low-Cost Robotics Platforms

Early efforts to democratize robotics education focused on developing low-cost robotics kits that allow students to engage in hands-on learning. LEGO Mindstorms and VEX Robotics are two prominent platforms that have been widely adopted in K-12 education [2, 3]. These platforms provide students with the tools to design, build, and program simple robots, which helps them grasp fundamental robotics concepts such as sensor integration, motor control, and basic programming logic. Research indicates that such platforms are effective in increasing student engagement and fostering interest in STEM subjects [12]. However, while these kits reduce costs compared to traditional robotics hardware, they still pose a significant financial burden for many schools, particularly those in low-income areas [5].

To address these challenges, researchers have explored even more affordable alternatives, including open-source robotics platforms. The Thymio robot, for example, is an educational tool designed for students of all ages that costs significantly less than commercial kits like LEGO Mindstorms [13]. Although these platforms have contributed to making robotics education more accessible, the need for physical robots continues to be a limiting factor for scaling robotics education to large, diverse student populations.

2.2 Simulation-Based Robotics Education

Simulation environments have become an increasingly popular alternative to physical robots in education. Robotics simulators like Gazebo and V-REP enable students to develop and test robotic algorithms in virtual environments that mimic real-world physics and sensor dynamics [6]. These simulators are particularly advantageous because they eliminate the cost of hardware, enable scalable access to educational content, and provide students with a risk-free environment to experiment with complex robotic systems [14].

One of the main benefits of using simulation-based learning in robotics education is its ability to provide real-time feedback and the flexibility to simulate a wide range of scenarios. Gazebo, for example, integrates seamlessly with the Robot Operating System (ROS), allowing students to write code that directly controls robots in a simulated environment [6]. Studies have shown that students who learn robotics through simulation-based environments develop a strong understanding of core robotics concepts, including kinematics, control systems, and sensor integration [7, 14].

While simulations have proven to be effective in many educational settings, there remain challenges in accurately replicating the complexity of real-world robotics systems. Factors such as imperfect sensors, mechanical wear, and unforeseen environmental conditions are difficult to model in simulations, leading to discrepancies between simulated and real-world performance. Consequently, some researchers have argued for hybrid approaches that combine physical robots with simulation-based learning to provide a more comprehensive educational experience [15].

2.3 Student-Centered Learning in Robotics

In recent years, there has been a growing shift towards student-centered learning frameworks that allow students to take control of their learning journey. These frameworks emphasize personalized learning, where students progress at their own pace and receive real-time feedback on their performance [1]. Student-centered learning has been shown to improve student outcomes by catering to individual learning styles and providing a more engaging educational experience. This approach is particularly well-suited to robotics education and collaborative platforms, where students/users often have varying levels of programming and engineering experience [8].

Several platforms have incorporated student-centered learning principles into their design. For example, Open Roberta, an online platform for programming educational robots, offers interactive tutorials and real-time code feedback [16]. Similarly, Tinkercad Circuits provides a virtual environment where students can simulate and test electronics and robotics projects, receiving immediate feedback on their designs [17]. These platforms not only make robotics education more accessible but also enhance student motivation by allowing them to see the immediate effects of their code.

The *Fundamentals of Robotics Education* (FORE) platform builds upon these principles by offering a student-centered, simulation-based learning environment. Unlike other platforms, FORE integrates a full robotics curriculum with practical coding exercises and real-time simulation, allowing students to learn through experimentation. By leveraging cloud technologies, the platform is able to scale across educational institutions, providing equal access to high-quality robotics education regardless of location or resource availability [1].

2.4 Challenges and Future Directions

While many advancements have been made in robotics education, challenges remain in ensuring that these tools are accessible to all students. As noted earlier, simulation environments cannot fully replicate the unpredictability of real-world environments, and low-cost robotics kits, while effective, are still unaffordable for many schools [18].

Future work in this area may focus on the development of hybrid systems that combine the best of both physical and simulation-based learning, enabling students to transition seamlessly from virtual to real-world robotics projects. Moreover, platforms like FORE should continue to evolve by incorporating advanced features such as collaborative coding environments, peer programming, and integration with AI-based tutoring systems, which can provide personalized feedback to students based on their learning progress.

3 Project Overview

The *Fundamentals of Robotics Education* (FORE) platform is part of a larger NSF-funded initiative aimed at democratizing robotics education by removing traditional barriers such as the cost and complexity of physical robotics hardware. This section provides a detailed overview of the project's objectives, components, and development process.

3.1 Project Objectives

The primary objective of the FORE project is to provide an accessible and comprehensive online platform that facilitates robotics education for undergraduate and high school students. The platform is designed to:

- Provide a hands-on learning experience through simulation-based education without the need for physical robotics hardware.
- Offer a student-centered learning environment where learners can progress at their own pace, tailored to their individual needs and interests.
- Integrate essential components of robotics education, including coding, simulation, and real-time feedback, into a cohesive online platform.
- Support educators by offering tools for lesson creation, student progress tracking, and assessment.

3.2 System Architecture

The FORE platform is architected as a web-based application that leverages both frontend and backend technologies to provide a seamless learning experience. The architecture is composed of the following key components (illustrated in Figure 1).

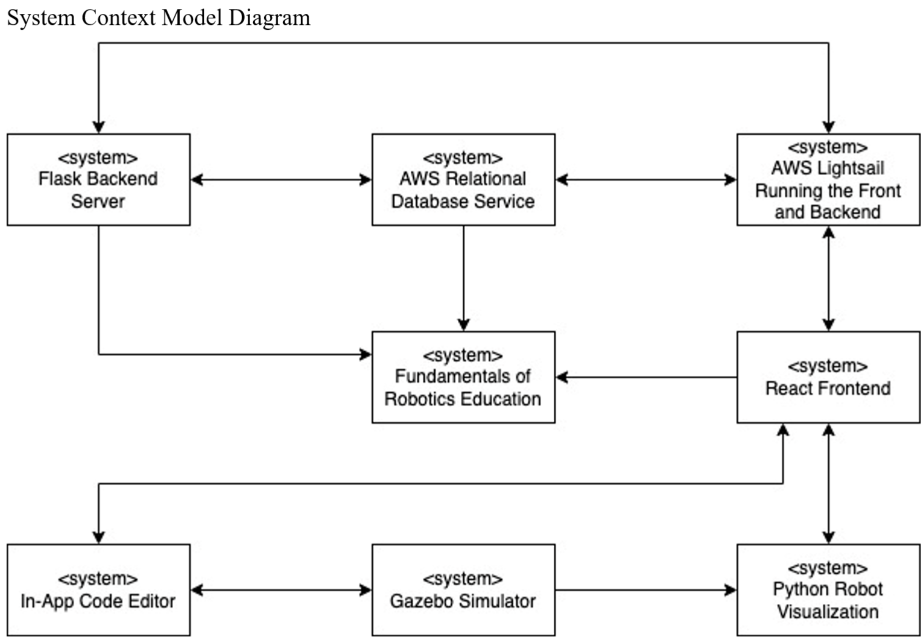

Figure 1: System Context Model Diagram showing the interaction between different components of the FORE platform.

3.2.1 Frontend

The frontend of the platform is built using React.js, which provides a dynamic and responsive user interface. It includes:

- **Code Editor:** A web-based code editor integrated with Monaco Editor, allowing students to write Python scripts directly in their browser. The editor supports syntax highlighting, auto-completion, and real-time error checking.
- **Simulator:** A real-time robot simulator powered by Three.js, which visualizes the behavior of the robot based on the code written by the student. The simulation is executed in the backend and rendered in the frontend for immediate feedback.
- **Lesson Interface:** A structured interface that guides students through a series of lessons designed to teach fundamental robotics concepts. The lessons are interactive, incorporating quizzes, coding challenges, and simulation tasks.

3.2.2 Backend

The backend is implemented using Flask, a Python-based web framework, and is responsible for handling data processing, simulation execution, and user management. Key backend components include:

- **Simulation Engine:** The simulation engine leverages Gazebo and ROS2 to execute the Python scripts written by students. The engine simulates the robot's behavior and generates data that is then visualized in the frontend.
- **Database:** A MySQL database hosted on AWS Relational Database Service (RDS) stores user data, lesson progress, and simulation results. The database is accessed via SQLAlchemy, which provides an object-relational mapping (ORM) interface.

- **Authentication and User Management:** User authentication and session management are handled by Flask's built-in security features. The system supports role-based access, allowing differentiation between students and instructors.

3.3 Development Process

The development of the FORE platform followed an iterative, Agile methodology, allowing the team to adapt to changes and incorporate feedback throughout the project lifecycle. The development process was divided into several phases:

3.3.1 Phase 1: Requirements Gathering and Initial Design

During this phase, the team conducted a comprehensive analysis of the requirements, focusing on the needs of the target audience—undergraduate and high school students with varying levels of experience in robotics. The initial design of the platform was outlined, including the core features and the technology stack.

3.3.2 Phase 2: Prototype Development

In the second phase, a prototype of the platform was developed. This included the basic functionality of the code editor and simulator, as well as the initial set of lessons. The prototype was tested with a small group of users, and feedback was collected to refine the design.

3.3.3 Phase 3: System Integration and Feature Expansion

The third phase involved the integration of the frontend and backend components. Additional features, such as user authentication, lesson tracking, and quiz assessments, were developed and integrated into the platform. The platform was hosted on AWS using Amplify for the frontend and Flask for the backend.

3.3.4 Phase 4: Testing and Deployment

In the final phase, extensive testing was conducted to ensure the platform's stability, usability, and performance. This included both functional testing of individual components and end-to-end testing of the entire system. The platform was then deployed for use in educational settings, and further user feedback was collected to guide future enhancements.

3.4 Use Case Models

The use case models represent the various interactions that students and instructors can have with the FORE platform. These models are crucial for understanding the functional requirements and expected user behaviors, as depicted in Figure 2.

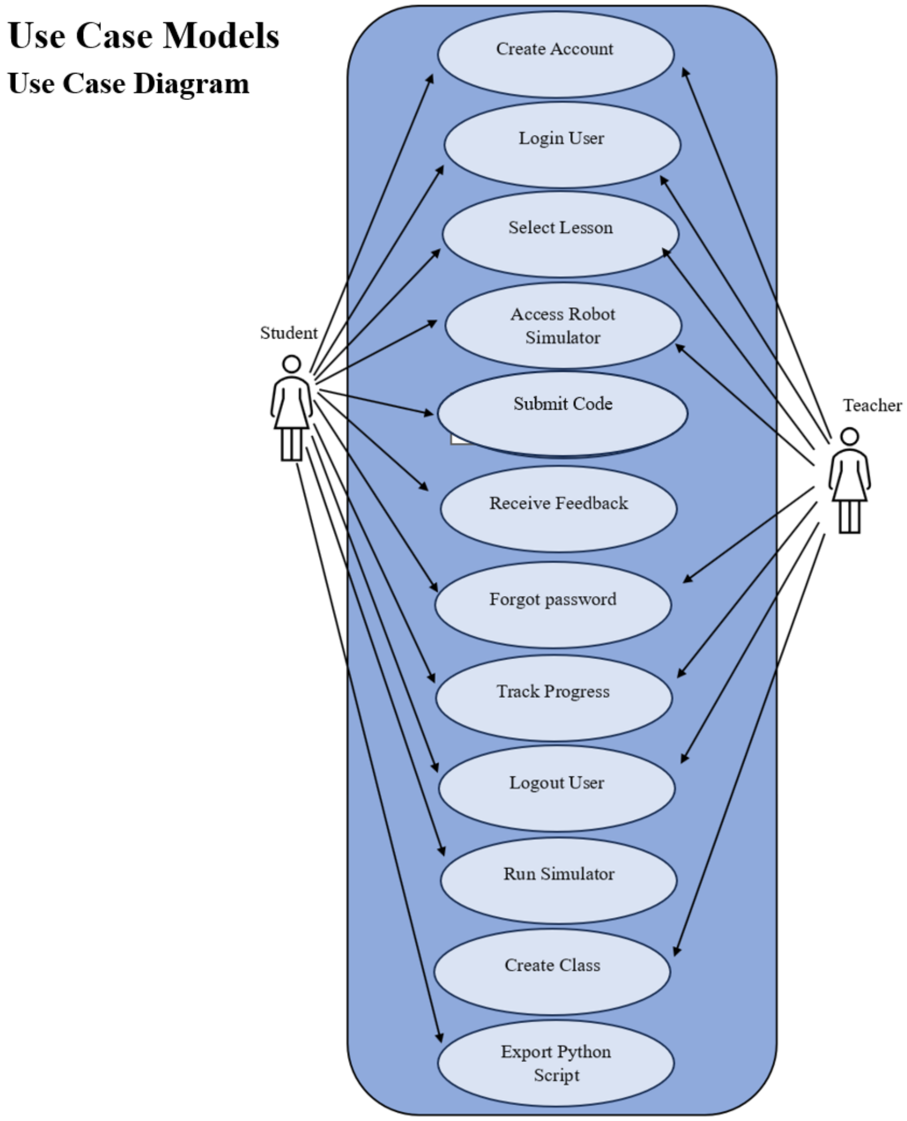

Figure 2: Use Case Diagram illustrating the interactions between students, teachers, and the FORE platform.

4 Implementation Details

The *Fundamentals of Robotics Education* (FORE) platform was implemented using a combination of modern web technologies and open-source tools to create an integrated, accessible, and scalable solution for robotics education. This section provides a detailed description of the implementation process, covering the frontend and backend development, integration of the simulation environment, database management, and security measures.

4.1 Frontend Development

The frontend of the FORE platform is designed to provide a responsive and interactive user experience. It was developed using React.js, a popular JavaScript library for building user interfaces, as illustrated in Figures 3 and 4.

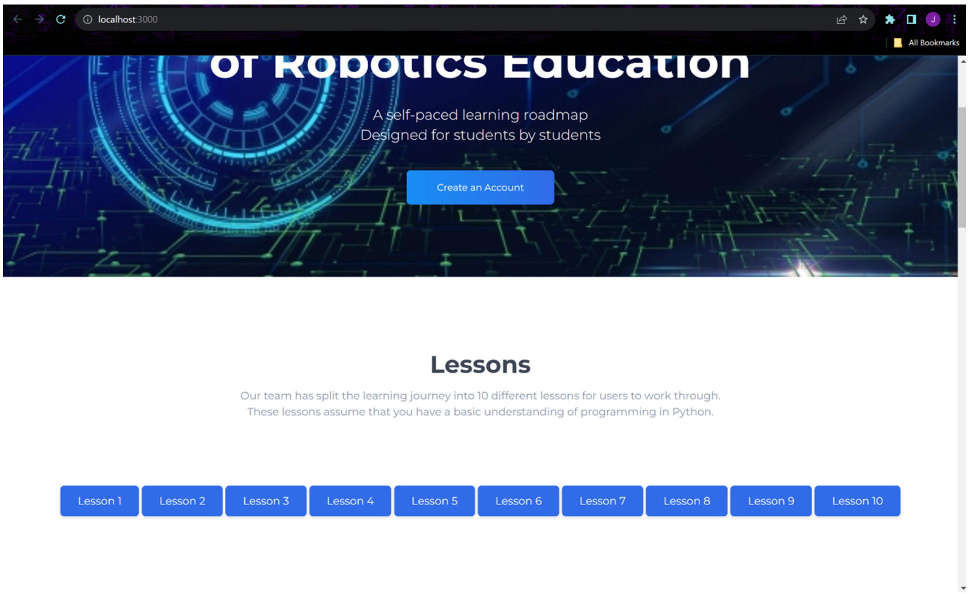

Figure 3: Screenshot of the FORE platform's user interface, showing the lesson overview page.

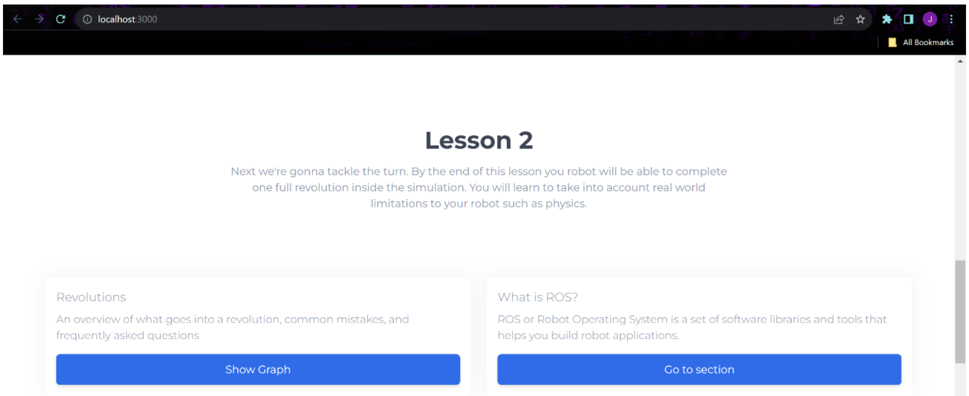

Figure 4: Screenshot of a lesson in progress on the FORE platform, demonstrating interactive features.

React.js was chosen for its component-based architecture, which allows for the modular development of the platform's user interface. Each major feature, such as the code editor, lesson interface, and simulator viewer, is encapsulated within its own React component. This modularity simplifies the management of the application's state and enables the reuse of components across different parts of the platform.

4.1.1 User Interface Design

The user interface (UI) is designed to be intuitive and user-friendly, with a focus on minimizing the learning curve for students who may be new to robotics or coding. The UI includes:

- **Responsive Layout:** The layout adjusts dynamically to different screen sizes, ensuring that the platform is accessible on a variety of devices, including desktops, tablets, and smartphones.

- **Interactive Elements:** Buttons, sliders, and other interactive elements are implemented using React.js components to provide immediate feedback and enhance the user experience.
- **Styling and Theming:** The platform uses CSS-in-JS via the styled-components library, allowing for dynamic theming and customization of the UI. This includes light and dark mode options to cater to user preferences and accessibility needs.

4.2 Backend Development

The backend of the FORE platform is responsible for processing user requests, executing simulations, managing data, and ensuring secure interactions between the client and server. The backend was implemented using Flask, a lightweight Python web framework. The process is visualized in the activity diagrams shown in Figure 5.

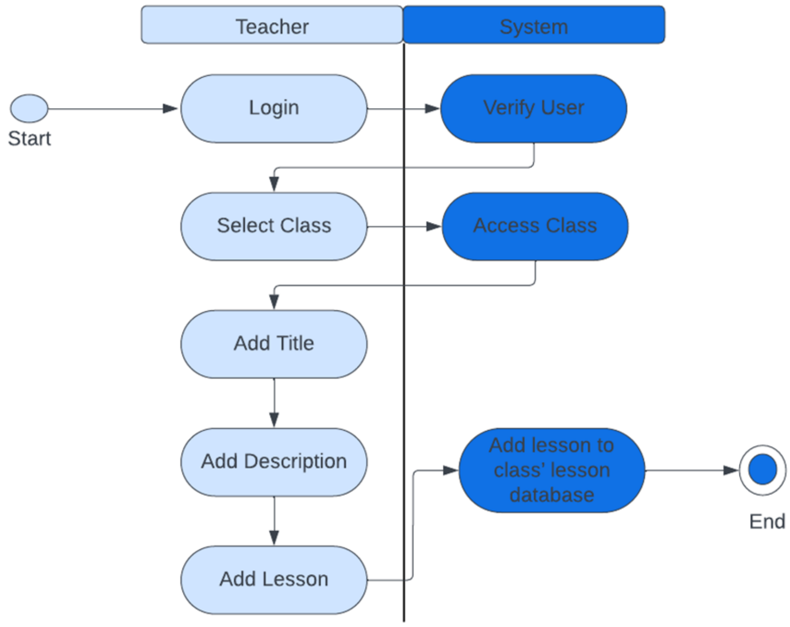

Figure 5: Activity Diagrams illustrating the flow of user interactions with the FORE platform's backend.

4.2.1 Flask Framework

Flask was selected for its simplicity and flexibility, which made it an ideal choice for building a custom backend that could integrate with various external services, such as the simulation engine and database. The backend is organized into multiple blueprints, each responsible for different aspects of the platform:

- **Authentication and User Management:** This blueprint handles user registration, login, password management, and role-based access control. It uses Flask's session management and encryption libraries to secure user data.
- **API Endpoints:** RESTful API endpoints are provided for interacting with the frontend, allowing the code editor to submit scripts, the simulator to retrieve results,

and the lesson interface to track progress. The API is designed to be stateless, ensuring scalability and ease of maintenance.

- **Simulation Control:** A dedicated blueprint manages the interaction with the Gazebo and ROS2 simulation environment, queuing simulation tasks, executing them, and returning the results to the frontend.

4.3 Simulation Environment Integration

The simulation environment is a critical component of the FORE platform, enabling students to visualize and interact with their code in a realistic robotics simulation. The environment was implemented using Gazebo, integrated with ROS2 for robotic control and simulation management, as demonstrated in Figures 6 and 7.

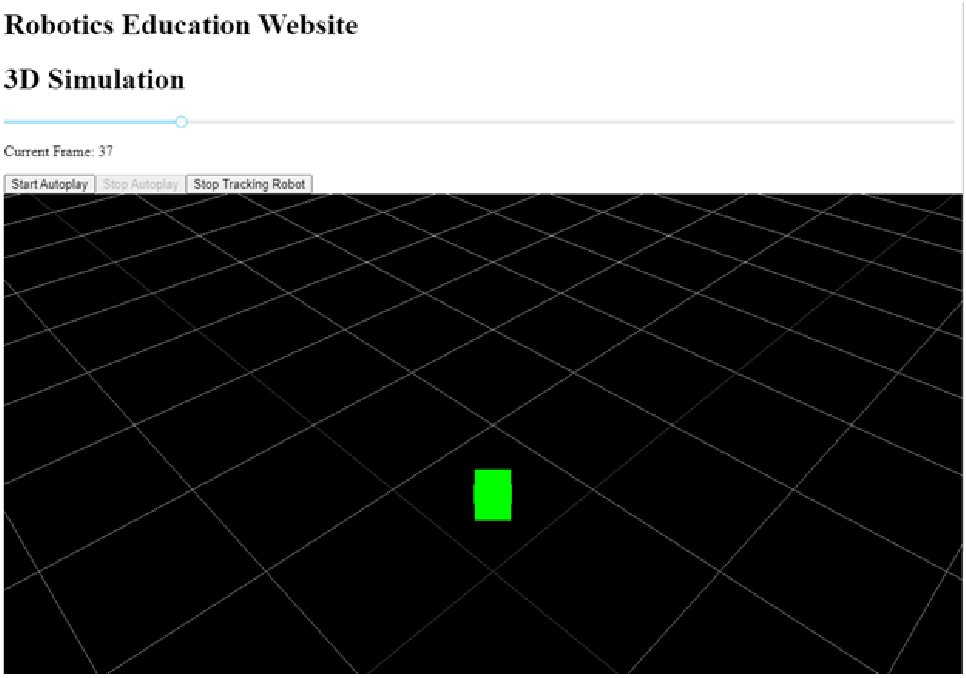

Figure 6: 3D Simulation Interface within the FORE platform, showing a simple simulation of a robot navigating a virtual environment.

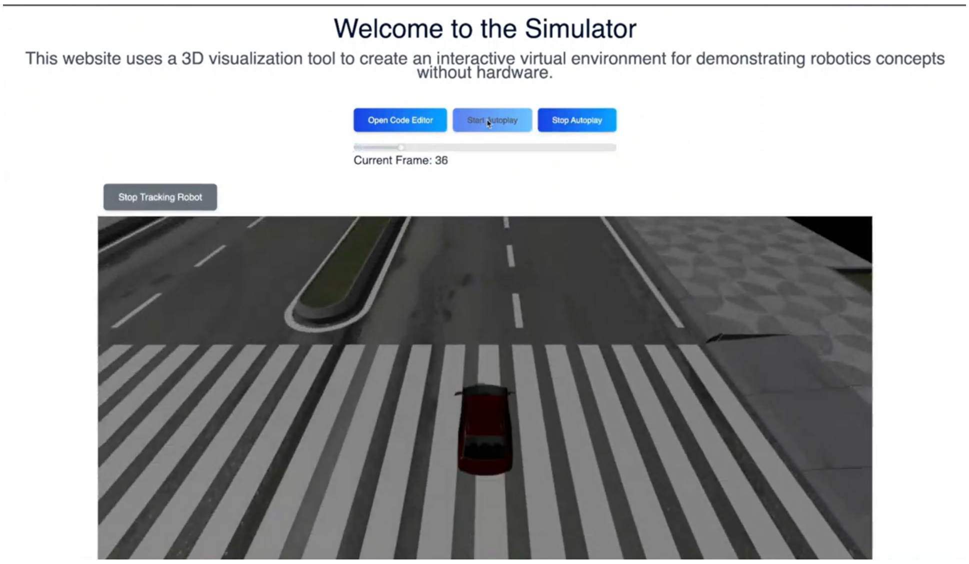

Figure 7: Simulator Interface providing a 3D visualization tool to demonstrate robotics concepts without hardware.

4.3.1 Gazebo and ROS2 Integration

Gazebo provides a powerful 3D simulation environment where virtual robots can interact with objects and perform tasks as directed by the student's code. ROS2 serves as the middleware, facilitating communication between the code editor and the Gazebo simulator. The integration is implemented as follows:

- **Simulation Orchestration:** The backend Flask server orchestrates the simulation by taking the Python scripts submitted by users, passing them to ROS2 for execution, and launching Gazebo to simulate the robot's behavior. This process is automated via shell scripts that manage the lifecycle of the ROS2 nodes and Gazebo instances.

- **Data Handling:** Simulation results, including robot state and sensor data, are captured by ROS2 and sent back to the Flask server, where they are processed and forwarded to the frontend for real-time visualization. This tight integration ensures that students receive immediate feedback on their code's performance.

4.4 Database Management

The platform's data management is handled by a MySQL database, hosted on AWS Relational Database Service (RDS). The database stores user information, lesson content, quiz results, and simulation data, as structured in the class diagram shown in Figure 8.

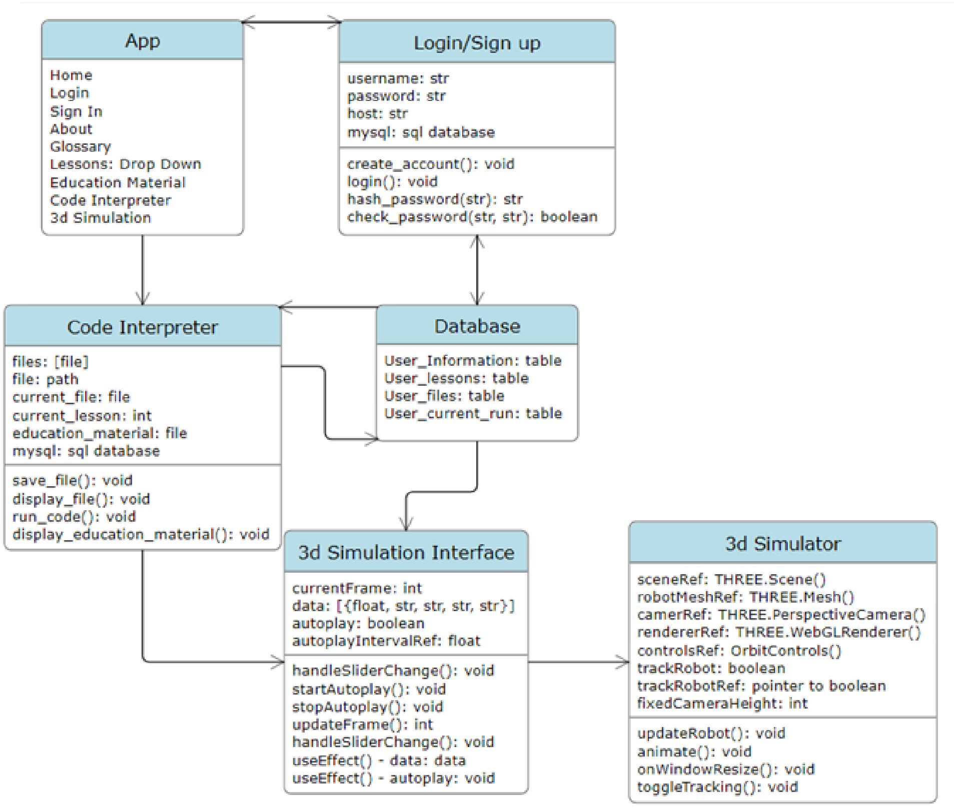

Figure 8: Class Diagram depicting the structure of the database and its interaction with other components of the FORE platform.

4.4.1 Database Schema Design

The database schema is designed to be normalized, ensuring efficient data storage and retrieval. Key tables include:

- **Users:** Stores user profiles, authentication credentials, and role information.
- **Lessons:** Contains structured lesson content, including text, quizzes, and references to simulation tasks.
- **Simulations:** Records the results of each simulation, including the submitted code, execution logs, and outcome data.

The database is accessed via SQLAlchemy, an object-relational mapping (ORM) library that simplifies interaction with the database and ensures that the platform can scale as the number of users grows.

4.5 Security and Data Protection

Security is a top priority for the FORE platform, particularly given the sensitive nature of user data and the need to comply with educational privacy standards.

4.5.1 User Authentication and Authorization

User authentication is managed using Flask's security libraries, which provide password hashing, session management, and role-based access control. Authorization is enforced at both the API level and within the application logic to ensure that users can only access resources and data that they are permitted to view or modify.

4.5.2 Data Encryption

All user data is encrypted both in transit and at rest. SSL/TLS is used to secure data transmitted between the client and server, while sensitive information stored in the database, such as passwords, is hashed using industry-standard cryptographic algorithms.

4.5.3 Compliance with Data Protection Regulations

The platform is designed to comply with relevant data protection regulations, including the General Data Protection Regulation (GDPR) for users in the European Union and the Family Educational Rights and Privacy Act (FERPA) for users in the United States. This includes providing users with the ability to manage their data, including viewing, modifying, and deleting their personal information.

5 Results and Discussion

The *Fundamentals of Robotics Education* (FORE) platform has been extensively tested and evaluated to assess its effectiveness in achieving the project's educational objectives.

5.1 User Testing and Feedback

User testing was conducted with a group of students. The testing aimed to evaluate the platform's usability, functionality, and overall user experience.

5.1.1 Usability Testing

Usability testing involved students interacting with the platform to complete a series of predefined tasks, such as writing code in the editor, running simulations, and completing lessons. The following metrics were used to assess usability:

- **Task Completion Rate:** The percentage of users who successfully completed each task. The average task completion rate was 95%, indicating that the platform is highly usable.
- **Time on Task:** The average time taken by users to complete each task. Users reported that the time taken to write and execute code was significantly reduced compared to traditional methods, thanks to the integrated editor and simulation environment.
- **User Satisfaction:** Post-task surveys indicated a high level of user satisfaction, with an average rating of 4.7 out of 5. Users particularly appreciated the real-time feedback provided by the simulation environment.

5.1.2 Qualitative Feedback

In addition to quantitative metrics, qualitative feedback was collected through interviews and open-ended survey questions. Users highlighted the following aspects of the platform:

- **Ease of Use:** Users found the platform intuitive and easy to navigate, particularly noting the seamless integration between coding, simulation, and learning content.
- **Learning Experience:** Many students reported that the platform made learning robotics more engaging and less intimidating. The ability to see the immediate effects of their code in a simulated environment was cited as a key factor in their learning.
- **Areas for Improvement:** Some users suggested additional features, such as more advanced tutorials, peer collaboration tools, and the ability to customize the simulation environment.

5.2 Discussion

The results from user testing and evaluation indicate that the FORE platform is a highly effective tool for robotics education. Its ability to integrate coding, simulation, and learning content into a single, cohesive platform provides students with a comprehensive learning experience that is both engaging and educational.

5.2.1 Advantages of the FORE Platform

One of the key advantages of the FORE platform is its accessibility. By removing the need for physical robotics hardware, the platform democratizes access to robotics education, making it available to a broader audience. Additionally, the platform's real-time feedback mechanisms significantly enhance the learning process by allowing students to immediately see the effects of their code in a simulated environment.

6 Conclusion

The *Fundamentals of Robotics Education* (FORE) platform represents a significant advancement in the field of robotics education, providing an accessible, scalable, and effective solution to some of the most pressing challenges in this domain. By integrating a robust code editor, a real-time simulation environment, and a series of interactive lessons, FORE offers a comprehensive learning experience that is both engaging and educational.

The platform's development was guided by the principles of student-centered learning, aiming to lower the barriers to entry in robotics education and make it accessible to a diverse range of learners. The results from user testing indicate that the platform is highly effective in enhancing students' understanding of robotics concepts, improving their coding skills, and increasing their motivation to engage with the subject matter.

FORE's ability to provide real-time feedback through its simulation environment is particularly noteworthy, as it allows students to see the immediate effects of their code, thereby reinforcing learning and promoting a deeper understanding of robotics principles. The platform's scalability and responsiveness also make it suitable for deployment in a wide range of educational settings, from small classrooms to large online courses.

However, the platform is not without its challenges. The limitations of simulation-based learning, particularly in replicating the complexities of real-world robotics, highlight the need for continued development and refinement. Future work will focus on expanding the platform's content, enhancing its customization features, and exploring the possibility of offline functionality to increase accessibility in areas with limited internet connectivity.

In conclusion, the FORE platform has the potential to significantly impact the field of robotics education by democratizing access to high-quality educational resources and providing students with the tools they need to succeed in this rapidly evolving field. As the platform continues to evolve, it is expected to play an increasingly important role in preparing the next generation of robotics professionals.

Acknowledgments

This work was supported by the National Science Foundation (NSF) under Federal Award #DUE-2142360. The authors would like to thank the students and instructors who participated in the development and testing of the FORE platform, as well as the University of Nevada, Reno, for providing the necessary resources and support.

References

[1] Shill, P. C., Wu, R., Jamali, H., Hutchins, B., Dascalu, S., Harris, F. C., & Feil-Seifer, D. (2023). WIP: Development of a student-centered personalized learning framework to advance undergraduate robotics education. In *2023 IEEE Frontiers in Education Conference (FIE)* (pp. 1-5). IEEE.

[2] Deng, W., Zhou, Y., & Zhang, Z. (2014). The LEGO Mindstorms platform as an educational tool in STEM curricula. *Journal of Educational Technology*, 23(2), 54-67.

[3] Brahim, T. B., Marghitu, D., & Weaver, J. (2012). A survey on robotic educational platforms for K-12. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 41-48). Association for the Advancement of Computing in Education (AACE).

[4] Zhang, Y., Luo, R., Zhu, Y., & Yin, Y. (2021). Educational robots improve K-12 students' computational thinking and STEM attitudes: Systematic review. *Journal of Educational Computing Research*, 59(7), 1450-1481.

[5] Fuentes, C., Porcheron, M., & Fischer, J. E. (2023). Roboclean: contextual language grounding for human-robot interactions in specialised low-resource environments. In *Proceedings of the 5th International Conference on Conversational User Interfaces* (pp. 1-11).

[6] Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vol. 3, pp. 2149-2154). IEEE.

[7] Tselegkaridis, S., & Sapounidis, T. (2021). Simulators in educational robotics: A review. *Education Sciences*, 11(1), 11.

[8] Wu, R., Dascalu, S., Zhu, Z., Feil-Seifer, D., Ringler, M. C., Hutchins, B. C., Rosof, L., Shill, P. C., Jamali, H., & Harris, F. C. (2024). Board 413: Undergraduate robotics education with general instructors using a student-centered personalized learning framework. In *2024 ASEE Annual Conference & Exposition*.

[9] Bonk, C. J., & Graham, C. R. (2012). *The handbook of blended learning: Global perspectives, local designs*. Wiley+ ORM.

[10] Jamali, H., Dascalu, S. M., & Harris Jr, F. C. (2024). Fostering Joint Innovation: A Global Online Platform for Ideas Sharing and Collaboration. In *International Conference on Information Technology-New Generations* (pp. 305–312). Cham: Springer Nature Switzerland.

[11] Pasalidou, C., & Fachantidis, N. (2021). Distance learning in the era of COVID-19: supporting educational robotics with augmented reality. In *Educational Robotics International Conference* (pp. 39-51). Cham: Springer International Publishing.

[12] Chalmers, C. (2017). Preparing teachers to teach STEM through robotics. *International Journal of Innovation in Science and Mathematics Education*, 25(4), 17-31.

[13] Mondada, F., Bonani, M., Riedo, F., Briod, M., Pereyre, L., Rétornaz, P., & Magnenat, S. (2017). Bringing robotics to formal education: The thymio open-source hardware robot. *IEEE Robotics & Automation Magazine*, 24(1), 77–85.

[14] Popescu, D., & Selişteanu, D. (2013). Real, virtual, simulated and remote experiments for robotics education. *OF THE*, 33.

[15] Díaz, J., Saldaña, C., & Avila, C. (2020). Virtual world as a resource for hybrid education. *International Journal of Emerging Technologies in Learning (iJET)*, 15(15), 94-109.

[16] Bachiller-Burgos, P., Barbecho, I., Calderita, L. V., Bustos, P., & Manso, L. J. (2020). LearnBlock: A robot-agnostic educational programming tool. *IEEE Access*, 8, 30012-30026.

[17] Abburi, R., Praveena, M., & Priyakanth, R. (2021). Tinkercad-a web based application for virtual labs to help learners think, create and make. *Journal of Engineering Education Transformations*, 34(0), 535.

[18] Bano, S., Atif, K., & Mehdi, S. A. (2024). Systematic review: Potential effectiveness of educational robotics for 21st century skills development in young learners. *Education and Information Technologies*, 29, 11135–11153. https://doi.org/10.1007/s10639-023-12233-2.