A

SYNOPSIS REPORT

On

"Implement A System To Enhance The Quality Of Education In Emerging Technology(Online Compiler)"

Submitted to

Autonomous Institute,

Affiliated to The Rashtrasant Tukadoji Maharaj Nagpur University

Department of Emerging Technologies

Bachelor of Technology (B. Tech)

Submitted By

Aboli Wankhede	(AM21004)
Vaishnavi Rahamatkar	(AM21014)
Tanushree Sarode	(AM21032)
Sakshi Mantri	(AM21038)

Guided By

Prof. Ravi Asati

S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT AND RESEARCH, NAGPUR

2023 – 2024

INDEX

ABSTRACT

In the dynamic landscape of Emerging Technologies, rapid prototyping and hands-on coding experiences are pivotal for students' learning journeys. To bolster this educational approach, we introduce an Online Compiler, a web-based platform designed to empower learners and educators alike in exploring, practicing, and mastering programming languages and concepts.

The Online Compiler serves as a virtual coding environment, providing a seamless and accessible space for users to write, compile, and execute code directly from their web browsers. With a user-friendly interface, it caters to diverse learning styles, from beginners taking their first steps in coding to advanced users refining their algorithms.

Key features include real-time syntax highlighting, error detection, and instant feedback on code execution, fostering a conducive learning environment. Additionally, the platform supports a wide array of programming languages, ensuring versatility in curriculum integration and accommodating various emerging technology disciplines.

Incorporating collaborative tools, the Online Compiler enables students to work on projects together, enhancing teamwork and problem-solving skills crucial in the tech industry. Furthermore, educators can create customized coding exercises, quizzes, and assignments, tailoring the learning experience to meet specific educational objectives.

Through this Online Compiler, we aim to democratize access to quality programming education, transcending geographical barriers and resource limitations. By providing a robust platform for hands-on learning, we envision a future where students, regardless of their backgrounds, can confidently navigate the complexities of Emerging Technologies and contribute meaningfully to the digital era.

INTRODUCTION

The rapid evolution of Emerging Technologies has ushered in a new era of possibilities, promising groundbreaking innovations across various industries. From artificial intelligence to blockchain, the landscape is vibrant with opportunities, calling for a workforce equipped with advanced programming skills and a deep understanding of these cutting-edge concepts.In the realm of education, keeping pace with this technological surge poses a significant challenge. Traditional classroom settings often struggle to provide the hands-on, real-time coding experiences essential for mastering these complex technologies. Recognizing this gap, we embark on a mission to revolutionize the educational journey through the development of an Online Compiler.The Online Compiler represents more than just a virtual coding environment; it stands as a gateway to immersive learning in Emerging Technologies. This platform aims to empower students and educators alike by offering a dynamic, interactive space where theoretical knowledge seamlessly transitions into practical application.With the Online Compiler, learners can delve into a diverse array of programming languages, experiment with algorithms, and witness the immediate impact of their code execution. Real-time syntax highlighting and error detection features provide invaluable guidance, transforming every line of code into a learning opportunity.Educators, too, find a powerful ally in the Online Compiler. Through this platform, they can craft tailored coding exercises, quizzes, and assignments that align with the curriculum's emerging technology focus. Moreover, collaborative tools foster teamwork among students, nurturing the collaborative problem-solving skills crucial for success in the tech-driven world.The significance of this Online Compiler extends beyond its technical capabilities. It represents a democratization of access to quality programming education, breaking down geographical barriers and leveling the playing field for aspiring technologists worldwide. Whether in bustling urban centers or remote rural communities, learners now have a passport to the frontier of Emerging Technologies.In this introduction, we set the stage for a transformative journey through the realms of education and technology. The Online Compiler emerges not just as a tool but as a catalyst for change, propelling us towards a future where every learner is empowered to innovate, create, and excel in the ever-evolving landscape of Emerging Technologies.

AIM

Ideate and implement a system to enhance the quality of education in Emerging Technology.

OBJECTIVES

Develop and implement an online education system.

1. Develop an online compiler for students and faculty to enhance practical coding skills in emerging technologies.
2. Create a user-friendly website with distinct roles for admin and users.
3. Enable seamless interaction between students and faculty through the online compiler platform.
4. Facilitate real-time code compilation, execution, and debugging for a variety of programming languages.
5. Provide a collaborative learning environment by allowing users to share code snippets and solutions.

LITERATURE REVIEW

1. ONLINE JAVA COMPILER WITH SECURITY EDITOR (Feb-2017)

Author: Shubham Chourasiya
Sneha Gadhave
Renuka Kulthe
Tushar Bhatt
Prof. Sunita Patil

Shubham Chourasiya and his team researched to develop the online java compiler with security editor. As the world is marching to stand on the internet, the security issue will be a bigger factor to focus on. They have developed the online java compiler with security editor which takes the user input file and compile it. The most aim of this project we will simply to put in writing a java program and compile it and rectify in online. The shopper machine doesn't having java Development Kit. The Shopper machine solely connected to the server. The server having java Compiler.

2. AN OPTIMIZING COMPILER FOR JAVA(JUNE-1999)

Author: Robert Fitzgerald
Todd B. Knoblock
Erik Ruf
Bjarne Steensgaard
David Tarditi

Here Authors have described the implementation of Marmot: a native-code compiler, runtime system, and library for Java, and evaluated the performance of a set of Marmot-compiled benchmarks. Because Marmot is intended primarily as a high quality research platform, Authors initially chose to concentrate on the extension of known, successful imperative and object-oriented language implementation techniques to Java. Similarly, they focused

more on ease of implementation and modification than on compilation speed, compiler storage usage, debugging support, library completeness, or other requirements of production systems. The remainder of this section summarizes what we have learned from implementing the Marmot system and from examining the performance of programs compiled by Marmot. They found Java bytecode to be an inconvenient input language, in that it obscures much of the information present in the Java source code (e.g., type information and the structure of high level operations such as try-finally). To generate good code, Marmot is forced to reconstruct missing types and recognize and optimize bytecode idioms. They were able to apply a large number of conventional scalar and object-oriented optimizations, most of which required extensions to support new Java language features. The features inducing the most significant changes were the precise exception model,Current limitations on code motion also hinder instruction scheduling. Overall, Marmot's techniques worked well, yielding application performance at least competitive to that of other Java systems, and approaching that of C++. This suggests that a production-quality compiler for Java could produce code competitive with that produced by production-quality compilers for C++. Marmot optimizations reduced the cost of safety checks to a quite modest level: a median of 4.1% for our benchmarks. Even with an efficient lock implementation, simple synchronization elimination reduces execution time of our 26 larger benchmarks by a median of 30%. Storage management added significant runtime costs, both inside and outside the garbage collector. Stack allocation reduced program execution time by as much as 11%.

3. ONLINE JAVA COMPILER(MAY-2019)

Authors: Shamali Kokare

Divya Chauhan

Jyoti Mishra

Prof. Manisha Singh

Shamali Kokare and her mates used Data mining techniques and Security editor techniques to develop a Online Java Compiler. Here the user barely connected to the server. Though he won't have any java development kit, still he can compile it and execute the java program by connecting to the server. Here the server is having the inbuilt java compiler and the compiled file is encrypted with the help of security editor.

4. ONLINE CODE EDITOR

Authors: Sonali R. Gujarkar
Samprada D.Nimrad
Shital Meshram

This paper presents a server-based code editor for java code. in other code editors they provide facility to run languages such as HTML, CSS ,JavaScript. This languages are web-based languages those code editors are enable to run the programming languages that's why programmes face many difficulties to run programming languages such as JAVA, PHP etc. but in our project we add programming language which is java .by using server based code editor programmer can run programming language ie. JAVA .it is a good tool to run multiple programming languages on same platform. it can help programmer to develop system faster and more accurate. this paper proposed the server-based code editor that was created for programmers or developers who want to write programmes without any platform requirements or without any specific physical computers. the server-based code editor is able to isolate programming languages by highlighting syntax of programme. We create the private cloud to store the software installation of the that languages which we want to run.

PURPOSED WORK

1. **Purpose of Work: Enhancing Education in Emerging Technologies through an Online Compiler**
    i. The primary purpose of developing an Online Compiler for Emerging Technologies is to revolutionize the way students and educators interact with and learn about cutting-edge fields such as artificial intelligence, blockchain, data science, and more. The system aims to bridge the gap between theoretical knowledge and practical application by providing a robust, user-friendly platform for coding, testing, and experimenting with code in real-time.

2. **Key Objectives:**

I. **Hands-On Learning Experience:**
   a. Offer a virtual coding environment that allows students to write, compile, execute, and debug code directly from their web browsers.
   b. Enable learners to gain practical experience in programming languages and concepts related to Emerging Technologies.

II. **Instant Feedback and Guidance:**
   a. Provide real-time syntax highlighting to aid in code readability and error detection.
   b. Offer immediate feedback on code execution, helping students identify and correct mistakes efficiently.

III. **Versatility in Language Support:**
   a. Support a wide array of programming languages commonly used in emerging technology domains, ensuring relevance and versatility.
   b. Allow students to explore and experiment with languages such as Python, JavaScript, C++, Java, and more.

IV. **Customized Learning Paths:**

a. Empower educators to create and share coding exercises, assignments, and quizzes tailored to specific emerging technology topics.
b. Enable the adaptation of curriculum content to match the rapidly evolving landscape of Emerging Technologies.

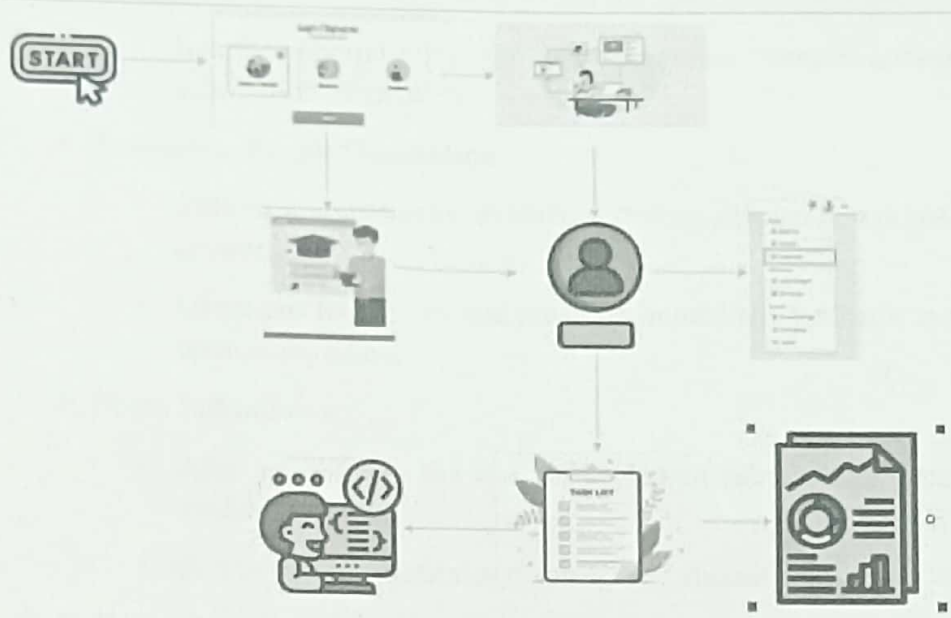
V. **Collaborative Learning and Project Work:**

a. Facilitate collaboration among students through shared coding environments and project spaces.
b. Foster teamwork and peer learning, essential skills for success in the tech industry.

VI. **Accessibility and Inclusivity:**

a. Democratize access to quality programming education by providing a platform that is accessible anytime, anywhere with an internet connection.
b. Remove geographical barriers and ensure that learners from diverse backgrounds have equal opportunities to excel in Emerging Technologies.

RESEARCH METHODOLOGY



- **Login Page**:
  - Faculty and students log in using their respective credentials.
  - Separate login interfaces for faculty and students.
- **Faculty Dashboard**:
  - Faculty can create new tests, manage existing tests, and view test results.
  - Interface for assigning tests to specific students or groups of students.
- **Test Creation**:
  - Faculty can create tests by specifying questions, options, and correct answers.
  - Ability to set time limits, assign test to specific classes or groups, and customize test parameters.
- **Student Dashboard**:
  - Students can view assigned tests and access them for solving.
  - Clear interface displaying test instructions and questions.

- **Test Solving:**
  - Students can attempt tests within the specified time limit.
  - Interface should allow for easy navigation between questions and submission of answers.
- **Automatic Result Generation:**
  - System automatically evaluates student responses against correct answers.
  - Generates test scores and provides immediate feedback to students upon completion.
- **Score Submission:**
  - After completing the test, students can submit their answers for evaluation.
  - Scores are automatically recorded and submitted to the faculty.
- **Result Viewing:**
  - Faculty can view test results for each student, including scores and detailed breakdown of answers.
  - Ability to export results for record-keeping or further analysis.

CONCLUSION

In conclusion, the implementation of our system to enhance the quality of education in emerging technology holds significant promise for bridging the gap between traditional education and the rapidly evolving demands of the digital age. By leveraging interactive online platforms, personalized learning experiences, real-world applications, and ongoing professional development for educators, we aim to empower students with the necessary skills and knowledge to thrive in a dynamic technological landscape. Through continuous evaluation and refinement, we aspire to adapt our system to meet the evolving needs of learners and educators alike, ensuring that our educational endeavors remain at the forefront of innovation and excellence. With a steadfast commitment to collaboration, adaptability, and student-centered learning, we believe our system will serve as a catalyst for transformative change in education, empowering individuals and communities to unlock their full potential in the digital era.

REFERENCES

1. Aho, A. V., & Ullman, J. D. (1977). Principles of Compiler Design. Addison-Wesley.
2. Appel, A. W. (1998). Modern Compiler Implementation in Java. Cambridge University Press.
3. Arnold, K. S., & Gosling, J. (1996). The Java programming language. Addison-Wesley.
4. Baumann, R. (2014). Developing web-based IDEs. Journal of Web Engineering, 13(2&3), 91-113.
5. Bodik, R., & Gupta, R. (2003). Web-based program analysis tools. In Proceedings of the 2003 ACM SIGPLAN workshop on Partial evaluation and semantics-based program manipulation (pp. 3-9).
6. Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ... & Gruber, R. E. (2006). Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 26(2), 4.
7. Cooper, K. D., & Torczon, L. (2011). Engineering a compiler. Elsevier.
8. Deitel, P., Deitel, H., & Deitel, A. (2017). Java: How to Program (Early Objects). Pearson.
9. Ferrante, J., Ottenstein, K. J., & Warren, J. D. (1987). The program dependence graph and its use in optimization. ACM Transactions on Programming Languages and Systems (TOPLAS), 9(3), 319-349.
10. Fisher, D., Grabski, T., & Walia, G. (2015). Online code editors: how do they affect coding behaviors and learning outcomes? Journal of Information Systems Education, 26(4), 287-296.
11. Fraser, C. W. (1992). A Retargetable C Compiler: Design and Implementation. Addison-Wesley.
12. Henglein, F., & Mossin, C. (1997). Web-based programming in a non-trivial domain. In Proceedings of the 1997 ACM SIGPLAN workshop on Partial evaluation and semantics-based program manipulation (pp. 88-97).
13. Lam, M. S., & Wilson, R. P. (1995). Limits of control flow on parallelism. ACM SIGPLAN Notices, 30(6), 46-57.
14. Palsberg, J., & Schwartzbach, M. I. (1991). Object-oriented type inference. ACM Transactions on Programming Languages and Systems (TOPLAS), 13(2), 237-268.
15. Rauschmayer, A. (2014). Exploring ES6: Upgrade to the next version of JavaScript. " O'Reilly Media, Inc.".
16. Seidl, M., & Zuleger, F. (2001). Web-based programming exercises: A didactic perspective. In Proceedings Seventh Annual IEEE International

Conference and Workshop on the Engineering of Computer Based Systems (ECBS'00) (pp. 122-128).
17. Seidl, M., & Zuleger, F. (2003). Web-based exercises in the compiler design course. In Proceedings of the 8th annual SIGCSE conference on Innovation and technology in computer science education (pp. 14-18).
18. Seidl, M., & Zuleger, F. (2004). Web-based compiler design exercises. In International Workshop on Web-Based Education (pp. 121-131). Springer, Berlin, Heidelberg.
19. Shao, Z., & Li, X. (2012). Development of a cloud-based compiler for distributed computing. In 2012 12th International Conference on Quality Software (QSIC) (pp. 216-221). IEEE.
20. Smith, M. D., & Ramsey, N. (2006). A practical framework for type inference error diagnosis. ACM Transactions on Programming Languages and Systems (TOPLAS), 28(5), 825-876.
21. Stroustrup, B. (2013). The C++ programming language. Addison-Wesley.
22. Subramanian, A., & Rajeswari, A. (2013). Development of a web-based compiler for multiple languages. In 2013 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5). IEEE.
23. Taha, W. (1999). A gentle introduction to multi-stage programming. In Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (pp. 14-23).
24. van Deursen, A., Klint, P., & Visser, J. (2000). Domain-specific language design requires feature descriptions. Journal of Computing and Information Technology, 8(4), 313-331.
25. Waite, W. M. (1996). Compiler Construction: Principles and Practice. PWS Publishing Company.

BIBLIOGRAPHY

1. BOOKS:

1. Training and Development: Enhancing Communication and Leadership Skills, by Steven A. Beebe, Timothy P. Mottet and K. David Roach, 2012
2. Training and Development: Theories and Applications: Theory and Applications by Dipak Kumar Bhattacharyya
3. Employee Training and Development (SIE) | 7th Edition by Raymond A. Noe, Amitabh Deo Kodwani

2. WEBSITE:

1. www.google.com
2. www.wikipedia.com
3. slideshare
4. Shodhganga.com