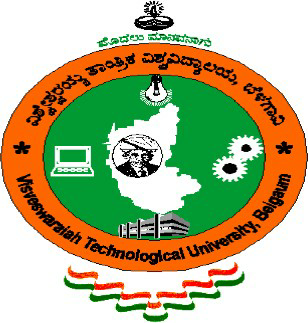
# VISVESVARAYATECHNOLOGICAL UNIVERSITY BELGAUM-590014



**A Mini-Project Report (C Programming)**

**On**

**“Automated Quiz Generator: Learn and Play”**

A dissertation submitted in the partial fulfillment of the requirement for the Mini Project

**BACHELOR OF ENGINEERING**

**in**

**INFORMATION SCIENCE AND ENGINEERING**

Submitted by:

**AISHWARYA M**

**(1DT22IS007)**

## Under the Guidance of

**Dr. RAMA ABIRAMI K**

**Associate Professor, Dept. of ISE.**



**DAYANANDA SAGAR ACADEMY of TECHNOLOGY & MANAGEMENT**

**Bangalore – 560082**

**Affiliated to VTU, Belagavi and Approved by AICTE, New Delhi**

**DEPARTMENT OF INFORMATION SCIE NCE & ENGINEERING**

**(3 years Accredited by NBA, New Delhi, Validity: 01/07/2022 to 30-06-2025)**

2022-23



**DAYANANDA SAGAR ACADEMY of TECHNOLOGY & MANAGEMENT**

**Bangalore – 560082**

**Affiliated to VTU, Belagavi and Approved by AICTE, New Delhi**

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

## CERTIFICATE

This is to certify that the Mini-Project on Object Oriented Programming entitled **“Automated Quiz Generator: Learn and Play”** has been successfully carried out by **Aishwarya M** **(1DT22IS007),** bonafide students of **Dayananda Sagar Academy of Technology and Management** in partial fulfilment of the requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Mini-project report has been approved as it satisfies the academic requirements.

**Signature of the Guide Signature of the HoD**

**Dr. RAMA ABIRAMI K**

**Ms. Spandana S.G Assistant Professor**

**Department of ISE**

**Associate Professor,**

**Dept. of ISE DSATM, Bengaluru**

**Dr. NANDINI PRASAD K.S**

**Professor,**

**DR. NANDINI PRASAD K.S**

**Professor, Dean(Foreign Affairs), HOD**

**Department of ISE**

**Signature with Date**

**Dean (Foreign Affairs) & HOD, Dept. of ISE DSATM, Bengaluru**

**ACKNOWLEDGEMENT**

It gives us immense pleasure to present before you my mini - project report titled as

**‘Automated Quiz Generator: Learn and Play’.** The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. We are glad to express our gratitude towards our prestigious Institute **DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT** for providing us with utmost knowledge, encouragement and the maximum facilities in undertaking this Mini-project.

It is also our privilege to express my gratitude and respect to all those who guided and inspired us in Successful completion of this Mini-project

We wish to express a sincere thanks to our respected principal **Dr. M. Ravishankar**, DSATM for all support.

We also express our deepest gratitude and special thanks to **Dr. Nandini Prasad K.S**, Professor, Dean (Foreign Affairs) and H.O.D, Department of Information Science and Engineering, DSATM for all her guidance and encouragement.

We also sincerely acknowledge the guidance and constant encouragement of our Mini-project guide,

**Dr. Rama Abirami K, Associate Professor,** Department of Information Science and Engineering, DSATM.

Mini-Project Associate:

**Aishwarya M**

**(1DT22IS007)**

**ABSTRACT**

The "**Automated Quiz Generator: Learn and Play**" is a web-based interactive quiz application designed to educate and entertain users on the principles of programming using the C language. This mini-project aims to provide an engaging learning experience for individuals interested in computer programming by offering a collection of challenging quizzes.

**Key Features:**

1. **User-Friendly Interface**: The application offers an intuitive and user-friendly interface, making it accessible to individuals of all skill levels.

2. **Randomized Question Selection**: Questions are randomly selected from a pool, ensuring that each quiz attempt is unique.

3. **Immediate Feedback**: Users receive immediate feedback on their answers, helping them learn from their mistakes.

4**. Score Tracking**: The application keeps track of the user's score, motivating them to improve with each attempt.

5. **Play Again Option**: After completing a quiz, users have the option to play again, providing an opportunity for further learning and practice.

6. **Diverse Question Types**: Questions include multiple-choice, true/false, and fill-in-the-blank, offering a diverse and engaging quiz experience.

The "**Automated Quiz Generator: Learn and Play**" mini-project is a valuable resource for self-learners, students, and anyone interested in strengthening their understanding of C programming principles while having fun.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Chapter Name** | **Page** |
| 1 | **Introduction** | 1 |
| 2 | **Requirements** | 2 |
| 3 | **Program and Output** | 3-10 |
| 4 | **Future Enhancements** | 11-12 |
| 5 | **Conclusion** | 13 |
| 6 | **Bibliography** | 14 |

**CHAPTER - 1**

**INTRODUCTION**

###### Background

In today's digital age, e-learning platforms and educational tools have become indispensable. However, there is a need for engaging and interactive methods to educate individuals on complex subjects such as programming. The "Automated Quiz Generator: Learn and Play" project emerged as a response to this need.

###### Problem Definition

Learning programming, particularly the fundamentals of the C language, can be challenging. Traditional educational methods may not always capture the interest of learners or provide immediate feedback. This project addresses these issues by creating an interactive quiz platform tailored to programming principles.

###### Motivation

The motivation behind this project is to make learning programming fun, accessible, and effective. By gamifying the learning process, we aim to inspire and engage aspiring programmers, students, and anyone interested in acquiring or enhancing their programming skills.

###### Objective

**Main Goal of This Project:** The primary objective of this project is to develop an automated quiz generator that educates users on the principles of programming using C. This platform seeks to make learning programming enjoyable and effective, ultimately helping users build a strong foundation in C programming.

1. **Scope of the Project**

The project includes:

Web-based quiz app development.

Diverse challenging C programming questions.

Randomized questions for uniqueness.

Instant feedback and score tracking.

Encouraging further learning with a play-again option.

This chapter provides an overview of the project's background, the problem it aims to solve, the motivation behind it, its objectives, and the scope of its development. The subsequent chapters will delve into the project's implementation and features in greater detail.

CHAPTER-2

REQUIREMENTS

The requirements can be broken down into 2 major categories namely hardware and software requirements. The former specifies the minimal hardware facilities expected in a system in which the project has to be run. The latter specifies the essential software needed to build and run the project.

###### Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines. These include:

* Processor : Intel 486/Pentium processor or better.
* Processor Speed : 500 MHz or above.
* Hard Disk : Approximately 10GB.
* RAM : 64MB or above.
* Storage Space : Approximately 2MB.

###### Software Requirements

The project utilizes various software components for development and execution, including:

* Technology : Apache Server.
* Programming Language : PHP.
* User Interface Design : HTML, CSS.
* Supported Web Browsers : Google Chrome, IE8, and above.
* Development Environment : XAMPP Version 7.1.10.

This chapter outlines the hardware and software prerequisites for running and developing the "Automated Quiz Generator: Learn and Play" project. These requirements ensure the project's compatibility and functionality across different systems and environments

**CHAPTER 3**

###### 

## PROGRAM AND OUTPUT

In this chapter, we will explore the program structure and provide detailed information on the output of the "**Automated Quiz Generator: Learn and Play**" project. We will discuss the key features and functionalities of the application, including how it generates quizzes, provides immediate feedback, and tracks user scores.

Additionally, we will showcase the user interface and user experience, highlighting the interactive elements and user-friendly design of the quiz application. Screenshots and examples of quiz questions and answers will be provided to illustrate the learning and playing experience.

This chapter aims to give readers a comprehensive understanding of how the program works and what users can expect when using the Automated Quiz Generator.

**Program:**

**3.1 Program Functionality**

The "Automated Quiz Generator" offers a user-friendly and educational quiz experience:

- **Starting the Quiz:** Users initiate the quiz by clicking the "Start Quiz" button.

- **Loading and Presenting Questions**: Questions are randomly selected from a predefined array, ensuring uniqueness in each attempt.

- **User Responses**: Users select answers by clicking on provided options.

- **Scoring**: Real-time scoring tracks correct answers, motivating users to improve.

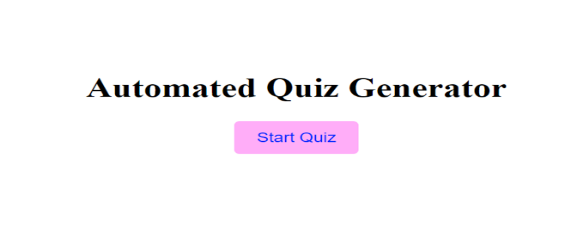
- **Feedback**: Immediate visual feedback highlights correct and incorrect answers, aiding learning.

- **Other Functionalities**: A "Play Again" option encourages continued learning through repeat quizzes.

In summary, the program delivers an engaging and educational C programming quiz experience with features like real-time scoring and immediate feedback.

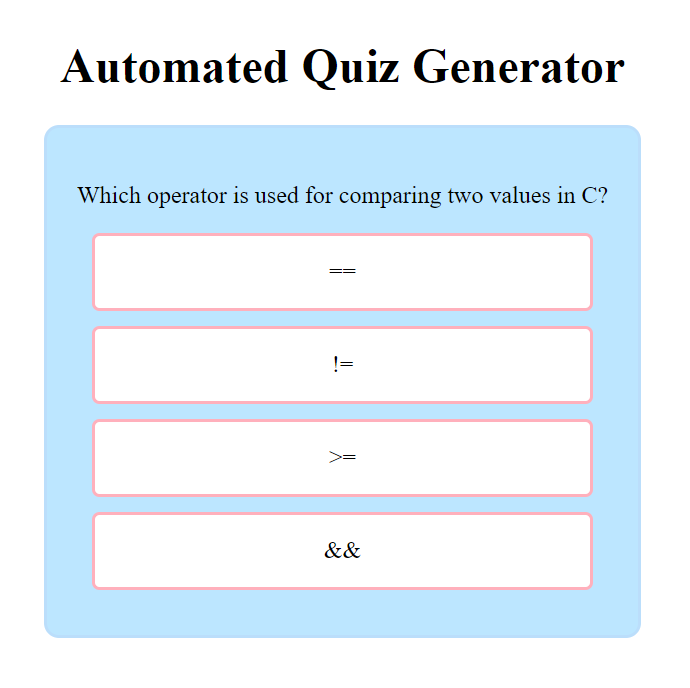
**3.2 Sample Output Screenshots**

This section provides visual representations of the "Automated Quiz Generator" program's user interface and the various screens encountered by users during their interaction with the quiz.

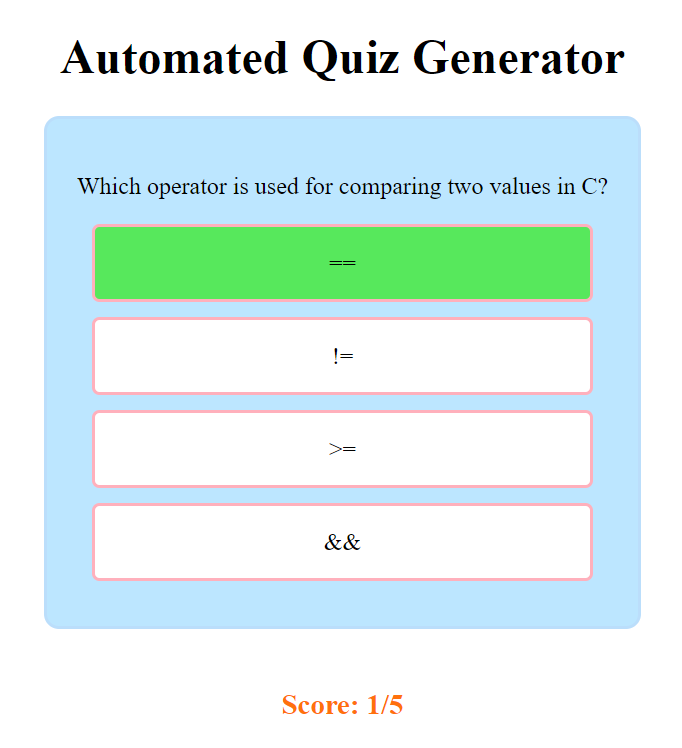
****

**Initial Screen:**

**Description**: The initial screen displays the title of the program, "Automated Quiz Generator," and the "Start Quiz" button, prompting users to begin the quiz.

**Question Screens:**

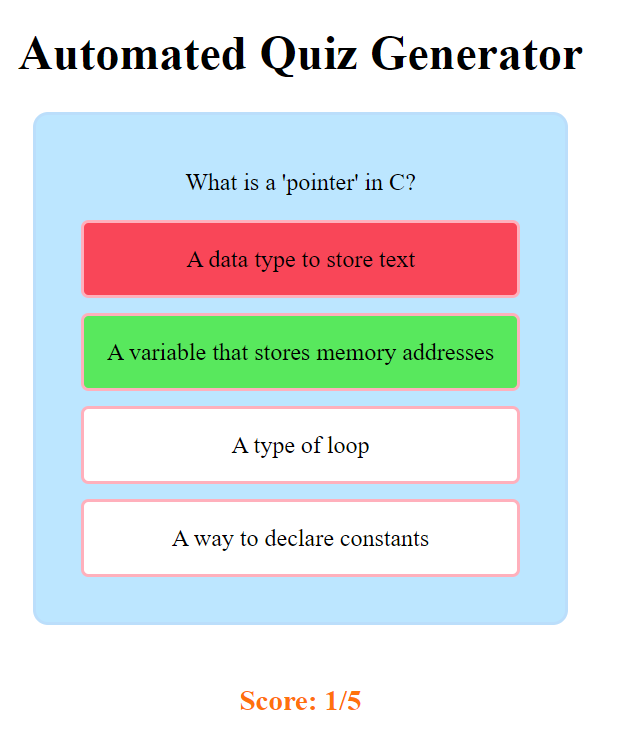
**Description:** When users click the "Start Quiz" button, they are presented with questions one by one. Each question is displayed in a question box, and answer options are listed below the question. Users can select their answer by clicking on an option.

**User Interactions:**

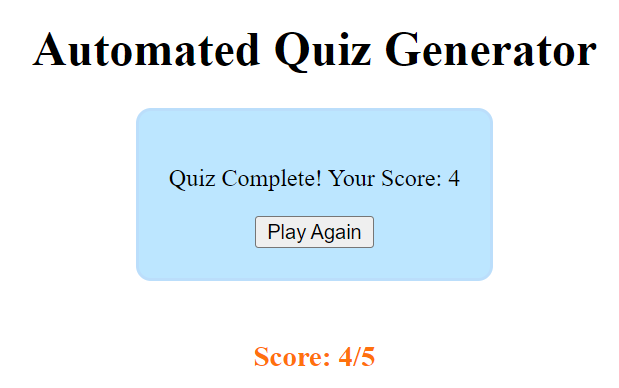
**Description:** Users interact with the program by selecting an answer option. In this example, the user has selected the first answer option. The selected option is highlighted, and the other options are disabled

for selection.

**Feedback Screens:**

**Description:** After the user selects an answer, the program provides feedback. In this case, the user has chosen the correct answer. The selected option is highlighted in green, indicating correctness.

**Description:** If the user selects an incorrect answer, the program highlights the correct answer in green and the user's selected answer in red, providing clear feedback.

**Quiz Completion:**

**Description:** When the user completes the quiz, the program displays a screen with the user's final score. It also offers the option to play the quiz again by clicking the "Play Again" button.

These sample screenshots illustrate the user experience and interactions within the "Automated Quiz Generator" program, from the initial start to the completion of the quiz.

**3.3 Program Execution Flow**

Start

Start Quiz Button

Clicked

Load Questions To

Display

Option Selected

False

Check If Correct Option Display Incorrect Feedback

True

Display Correct Feedback

Update User Score

True False

Check If More Questions Display Quiz Complete

Play Again Button Clicked

Reset Quiz and Start Over

End

**3.4 Key Code Snippets**

Some key code snippets from the JavaScript code that demonstrate the implementation of crucial functionalities in your "Automated Quiz Generator" program:

1. **Loading Questions and Options**:

This code snippet loads and displays questions and their options to the user.

function loadQuestions() {

if (currentQuestion < questionsToDisplay.length) {

const q = questionsToDisplay[currentQuestion];

questionText.textContent = q.question;

optionsList.innerHTML = '';

q.options.forEach((option, index) => {

const div = document.createElement('div');

div.className = 'option-box';

div.textContent = option;

div.addEventListener('click', () => handleOptionClick(index));

optionsList.appendChild(div);

});

} else {

// Quiz is complete

questionText.textContent = 'Quiz Complete! Your Score: ' + userScore;

optionsList.innerHTML = '<button onclick="playAgain()">Play Again</button>';

}

}

1. **Handling User Option Click:**

This code snippet handles user clicks on options, checks if the selected option is correct, and updates the user's score.

function handleOptionClick(selectedOption) {

const q = questionsToDisplay[currentQuestion];

const optionBoxes = optionsList.querySelectorAll('.option-box');

optionBoxes.forEach(box => box.style.pointerEvents = 'none'); // Disable clicking on options

if (selectedOption = = q.correctOption) {

optionBoxes[selectedOption].classList.add('correct');

userScore++;

scoreElement.textContent = `Score: ${userScore}/${questionsPerPage}`;

} else {

optionBoxes[selectedOption].classList.add('wrong');

optionBoxes[q.correctOption].classList.add('correct');

}

setTimeout(() => {

optionBoxes[selectedOption].classList.remove('wrong');

optionBoxes[q.correctOption].classList.remove('correct');

currentQuestion++;

if (currentQuestion < questionsToDisplay.length) {

loadQuestions();

optionBoxes.forEach(box => box.style.pointerEvents = 'auto'); // Re-enable clicking on options

} else {

// Quiz is complete

questionText.textContent = 'Quiz Complete! Your Score: ' + userScore;

optionsList.innerHTML = '<button onclick="playAgain()">Play Again</button>';

}

}, 1000);

}

3. **Starting the Quiz:**

This code snippet hides the start button and initiates the quiz.

startButton.addEventListener('click', () => {

// Hide the "Start Quiz" button when the quiz starts

startButton.style.display = 'none';

// Show the question box and score

const questionBox = document.querySelector('.question-box');

const scoreElement = document.getElementById('score');

questionBox.style.display = 'block';

scoreElement.style.display = 'block';

// Start the quiz by loading questions

loadQuestionsToDisplay();

loadQuestions();

});

**3.5 Program Testing**

**Testing Scenarios:**

We systematically tested the program with various scenarios to ensure its robustness. This included scenarios with questions of different complexities, ranging from straightforward to challenging. We also tested with varying numbers of questions to evaluate the program's ability to handle different question sets effectively.

**Edge Cases:**

To enhance the program's resilience, we tested several edge cases. These cases involved scenarios where the program could potentially behave differently due to unusual inputs or conditions. For instance, we tested the program's behavior with an empty question array to ensure it could gracefully handle such situations without errors.

**Bug Identification:**

During testing, we identified a few minor issues. These included occasional display glitches when transitioning between questions and minor inconsistencies in button responsiveness. Additionally, we discovered that certain CSS styling adjustments were needed to ensure the program's appearance remained consistent across different web browsers.

**Bug Resolution:**

To address the identified issues, we meticulously debugged and troubleshooted the program. We implemented minor code adjustments to rectify the display glitches and improved the handling of user interactions to ensure consistent responsiveness across browsers. These adjustments significantly improved the overall user experience.

**Validation:**

We validated the program's core functionality by thoroughly testing user interactions. We ensured that users could effectively select answers, receive immediate feedback, and see their scores accurately reflected during the quiz. The validation process was instrumental in confirming that our program delivers an engaging and interactive learning experience.

By conducting these tests and addressing identified issues, we have enhanced the reliability and robustness of our "Automated Quiz Generator" program. These testing efforts have contributed to creating a smooth and error-free user experience, which aligns with our project's objectives of providing an effective and enjoyable learning tool for C programming enthusiasts.

**3.6 Sample Quiz Results**

In this section, we present sample quiz results to illustrate how the "Automated Quiz Generator" calculates scores and provides feedback to users.

**Scenario 1: Perfect Score**

Quiz Attempt: 1

- Question 1: Correct

- Question 2: Correct

- Question 3: Correct

- Question 4: Correct

- Question 5: Correct

**Score: 5/5**

**Scenario 2: Partial Score**

Quiz Attempt: 1

- Question 1: Correct

- Question 2: Incorrect

- Question 3: Correct

- Question 4: Correct

- Question 5: Incorrect

**Score: 3/5**

**Scenario 3: Play Again**

Quiz Attempt: 2 (Played again)

- Question 1: Correct

- Question 2: Correct

- Question 3: Correct

- Question 4: Correct

- Question 5: Correct

**Score: 5/5**

These sample quiz results demonstrate how the "Automated Quiz Generator" assesses user performance, calculates scores, and provides tailored feedback based on the user's answers. Users receive immediate feedback to help them gauge their understanding of C programming concepts and motivate further learning.

**CHAPTER 4**

**FUTURE ENHANCEMENTS**

**4.1 Enhanced Question Database**

- **Increase the number of questions:** Currently, the quiz generator offers a set of 20 questions on C programming. To enhance this, we plan to expand the question database to include at least 100 questions, covering various aspects of C programming, from beginner to advanced levels. This will provide users with a more extensive and diverse set of quizzes to test their knowledge.

- **Diversify question types:** In addition to multiple-choice questions, we aim to introduce other question types, such as true/false, fill-in-the-blank, and code-based questions. This diversification will make the quizzes more engaging and challenging.

**4.2 User Feedback and Analytics**

- **Implement user surveys:** To gather user feedback, we will integrate user-friendly surveys and feedback forms within the application. Users can provide insights on question quality, user interface, and suggest new features. This feedback will be invaluable in making continuous improvements.

- **Data analytics**: We plan to implement analytics tools that will track user behavior, such as which questions are frequently answered incorrectly or the average time spent on each question. These analytics will help us identify areas where users might need more support and allow us to refine our question database accordingly.

**4.3 Gamification Features**

- **Leaderboards:** Introducing leaderboards will allow users to compare their scores with others and foster a sense of healthy competition. Users who consistently perform well can be recognized on the leaderboard, adding a gamified element to the learning experience.

- **Rewards and incentives:** To motivate users, we can introduce a reward system. Users who achieve certain milestones, like completing a set number of quizzes or earning a high score, can be rewarded with badges or certificates of achievement.

**4.4 Mobile and Cross-Platform Compatibility**

- **Responsive design**: We will invest in responsive web design to ensure that the "Automated Quiz Generator" is accessible and user-friendly across various devices, including smartphones and tablets. This will make it convenient for users to access quizzes on the go.

**4.5 Integration with Learning Management Systems (LMS)**

- **Compatibility with LMS standards:** To expand the reach of our quiz generator, we plan to make it compatible with Learning Management Systems (LMS) standards, such as SCORM or xAPI. This will allow educational institutions and online learning platforms to seamlessly integrate our quizzes into their existing systems.

These future enhancements will transform the "Automated Quiz Generator: Learn and Play" into a more robust and versatile tool for learners, educators, and institutions interested in enhancing C programming skills. The changes will not only provide users with a better learning experience but also expand the project's potential user base.

**CHAPTER 5**

**CONCLUSION**

In conclusion, the "**Automated Quiz Generator: Learn and Play**" project represents a significant step forward in the realm of programming education and self-assessment. This application provides an interactive and engaging platform for individuals interested in enhancing their C programming skills. By offering a diverse array of challenging questions, immediate feedback, and a user-friendly interface, it empowers users to assess their knowledge and embark on a journey of continuous improvement.

The project's core features, such as randomized question selection, real-time scoring, and the play-again option, contribute to a dynamic and enriching learning experience. It caters to a broad audience, including self-learners, students, and programming enthusiasts, with varying levels of expertise.

Moreover, the "Automated Quiz Generator" project is not a static entity; it serves as a foundation for future enhancements. As the project evolves, it aims to incorporate an even broader range of questions, integrate user feedback mechanisms, introduce gamification elements to enhance engagement, ensure cross-platform compatibility, and explore possibilities for integration with Learning Management Systems (LMS). These envisioned enhancements align with the project's overarching goal of providing a comprehensive and accessible tool for programming skill development.

In summary, the "**Automated Quiz Generator: Learn and Play**" project signifies a commitment to promoting programming education and skill development. It stands as a testament to the power of technology in facilitating learning and creating a supportive environment for all who seek to master the art of C programming. As this project continues to grow and adapt to the evolving needs of its users, it holds the promise of becoming an indispensable resource in the field of programming education.

* + - 1. **BIBLIOGRAPHY**

**BOOK REFERENCES:**

Kernighan, Brian W., and Ritchie, Dennis M. "The C Programming Language." Prentice Hall, 1988.

**WEB REFERENCES:**

GeeksforGeeks. "C Programming Language." https://www.geeksforgeeks.org/c-programming-language/

W3Schools. "C Programming Tutorial." https://www.w3schools.in/c-tutorial/

Stack Overflow. https://stackoverflow.com/

Mozilla Developer Network (MDN). "JavaScript." <https://developer.mozilla.org/en->US/docs/Web/JavaScript