**Industrial Internship Report on**

**”QUIZ GAME”**

**Prepared by**

**KARISHMA JAISWAL**

|  |
| --- |
| Executive Summary |

|  |
| --- |
| This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).  This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.  My project is a “Quiz Game”. The quiz game project you provided is a simple interactive quiz game built using the Python programming language and the Tkinter library for creating the graphical user interface (GUI). The program presents a series of multiple-choice questions to the user and allows them to select one of the available choices. After answering all the questions, the program displays the user's score in the message box.  This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship. |

**TABLE OF CONTENTS**

[1 Preface 3](#_Toc139702806)

[2 Introduction 4](#_Toc139702807)

[2.1 About UniConverge Technologies Pvt Ltd 4](#_Toc139702808)

[2.2 About upskill Campus 8](#_Toc139702809)

[2.3 Objective 9](#_Toc139702810)

[2.4 Reference 9](#_Toc139702811)

[2.5 Glossary 10](#_Toc139702812)

[3 Problem Statement 11](#_Toc139702813)

[4 Existing and Proposed solution 12](#_Toc139702814)

[5 Proposed Design/ Model 13](#_Toc139702815)

[5.1 High Level Diagram (if applicable) 13](#_Toc139702816)

[5.2 Low Level Diagram (if applicable) 13](#_Toc139702817)

[5.3 Interfaces (if applicable) 13](#_Toc139702818)

[6 Performance Test 14](#_Toc139702819)

[6.1 Test Plan/ Test Cases 14](#_Toc139702820)

[6.2 Test Procedure 14](#_Toc139702821)

[6.3 Performance Outcome 14](#_Toc139702822)

[7 My learnings 15](#_Toc139702823)

[8 Future work scope 16](#_Toc139702824)

# Preface

**Summary of the whole 6 weeks’ work:-**

In week one , I got to know about the python in Data Science. Here I have the detailed description about it:

Python is an excellent programming language for data science due to its simplicity, versatility, and a vast ecosystem of libraries and tools tailored for data analysis, visualization, and machine learning. Here are some of the key reasons why Python is widely used in the data science community:

**1.Rich Libraries and Tools:** Python has various powerful libraries such as NumPy, Pandas, Matplotlib, and Seaborn that facilitate data manipulation, analysis, and visualization. Additionally, libraries like SciPy and statsmodels provide advanced statistical capabilities.

**2.Jupyter Notebooks:** Python integrates seamlessly with Jupyter Notebooks, which are interactive, web-based environments that allow data scientists to mix code, documentation, and visualizations in a single document. This makes it easier to explore data and share results with others.

**3.Machine Learning Libraries:** Python offers popular machine learning frameworks like Scikit-learn, TensorFlow, and PyTorch, which enable data scientists to build and deploy sophisticated machine learning models.

**4.Community and Support:** Python has a large and active community of data scientists and developers. This means that there are ample resources, tutorials, and forums available to help with any data science-related questions or issues.

**5.Integration with Other Technologies:** Python can easily interface with other technologies, making it a preferred choice for integrating data science with web applications, databases, big data tools, and cloud services.

**6.Readable and Expressive Syntax:** Python's clean and readable syntax makes it easy to understand and maintain code, which is crucial when working on complex data science projects.

**7.Open Source**: Python is open-source, meaning it's free to use and can be customized as needed, which is particularly beneficial for data science projects with specific requirements.

**8.Data Visualization:** Python offers robust visualization libraries like Matplotlib and Seaborn, allowing data scientists to create insightful charts, plots, and graphs for better understanding and communication of data.

**9.Data Cleaning and Preprocessing**: Python's libraries like Pandas provide powerful tools for data cleaning, transformation, and preprocessing, which are essential steps before performing data analysis.

10**.**Python's ease of use and various web scraping libraries like BeautifulSoup enable data scientists to extract data from websites for analysis.

In week two , I got to know about the various conditional statements in python. Below is the detailed description about it:

1.The if keyword is used to check if a certain condition is True. If the condition is True, the indented block of code following the if statement will be executed.

2.The elif keyword allows you to check additional conditions. It is optional and can be used multiple times in the same statement. If the previous if or elif conditions are False, the code block following the elif statement will be executed if the corresponding condition is True.

3.The else keyword is optional and used to provide a default block of code to be executed if none of the preceding conditions are True.

In week three , I got to learn about how python help in SEO and the must known topics for data science:

Python can be beneficial in SEO (Search Engine Optimization) in various ways, especially when used for web scraping, data analysis, and automation. Here are some ways Python can assist with SEO:

**1.Web Scraping:** Python's web scraping libraries, such as BeautifulSoup and Scrapy, allow you to extract data from websites, including meta tags, keywords, content, and URLs. This data can be useful for analyzing competitors, researching keywords, or monitoring search engine rankings.

**2.Keyword Research:** Python can be employed to automate keyword research tasks. Using APIs and web scraping, you can collect data from search engines and keyword research tools to identify relevant keywords, their search volumes, and competition levels.

**3.Content Optimization:** Python can help analyze the content on your website or blog. By using natural language processing (NLP) libraries like NLTK or spaCy, you can assess the quality and relevance of the content, ensure appropriate keyword usage, and improve readability.

**4.Automating SEO Tasks:** Python allows you to automate repetitive SEO tasks like generating sitemaps, checking broken links, or monitoring website health. This can save time and effort while ensuring better site performance.

**5.Data Analysis:** Python's data analysis libraries, such as Pandas, can be utilized to process and analyze SEO-related data. This includes traffic data, user behavior, and conversion rates, which can provide insights to enhance SEO strategies.

**6.Rank Tracking:** With Python, you can build custom scripts to track search engine rankings for specific keywords over time. This information can help you understand the effectiveness of your SEO efforts and adjust your strategies accordingly.

**7.Backlink Analysis:** Python can be used to analyze backlinks to your website or your competitors' sites. By gathering and processing backlink data, you can identify valuable link-building opportunities and disavow harmful links.

**8.Technical SEO Audits:** Python can assist in performing technical SEO audits. You can create scripts to check for duplicate content, broken links, page load speed, and other technical aspects that impact search engine rankings.

**9.Competitor Analysis:** Using Python to scrape and analyze data from competitor websites can offer valuable insights into their SEO strategies, allowing you to identify strengths and weaknesses and refine your own approach.

**10.Reporting and Visualization:** Python's visualization libraries like Matplotlib or Seaborn can help you create visual reports and dashboards to communicate SEO insights effectively to stakeholders.

Data science is a multidisciplinary field that combines knowledge from various domains, including mathematics, statistics, computer science, and domain expertise. Here are some essential topics that one must be familiar with to excel in data science:

**1.Programming Languages:** Proficiency in programming languages is crucial. Python and R are the most commonly used languages for data science tasks. Python's versatility and extensive libraries make it a preferred choice.

**2.Mathematics and Statistics:** Understanding foundational mathematical concepts, such as linear algebra, calculus, probability, and statistics, is essential for data analysis and machine learning.

**3.Data Manipulation and Analysis:** Knowledge of data manipulation libraries like Pandas in Python or data.table in R allows efficient data cleaning, transformation, and exploration.

**4.Data Visualization:** Visualization tools like Matplotlib, Seaborn (Python), and ggplot2 (R) help create informative plots and graphs to communicate insights effectively.

**5.Machine Learning:** Understanding the principles and algorithms of machine learning is crucial. This includes supervised and unsupervised learning, decision trees, random forests, support vector machines, neural networks, and more.

**6.Model Evaluation and Validation:** Knowing how to evaluate and validate machine learning models using techniques like cross-validation, precision-recall, ROC curves, etc., is vital to ensure reliable results.

**7.Feature Engineering:** Feature engineering involves selecting, transforming, and creating relevant features to improve model performance. It requires creativity and domain knowledge.

**8.Big Data Tools:** Familiarity with big data tools like Apache Spark or Hadoop can be essential when dealing with large-scale datasets.

**9.Data Ethics and Privacy:** Understanding the ethical implications of working with data, ensuring data privacy, and adhering to relevant regulations is essential.

**10.Version Control:** Proficiency in version control systems like Git helps in collaborative data science projects and tracking changes.

**11.Database Management:** Knowledge of SQL and working with relational databases is valuable for data retrieval and integration.

**12.Natural Language Processing (NLP):** NLP techniques are used to analyze and interpret human language data, which is crucial for tasks like sentiment analysis, text classification, etc.

**13.Deep Learning:** Familiarity with deep learning frameworks like TensorFlow or PyTorch is valuable for tackling complex problems like image recognition, language translation, and more.

**14.Time Series Analysis:** Time series analysis techniques are used for forecasting and understanding temporal patterns in data.

**15.Cloud Computing:** Familiarity with cloud platforms like AWS, GCP, or Azure enables scalable data processing and model deployment.

**16.Data Storytelling:** Being able to communicate data-driven insights effectively to non-technical stakeholders is an essential skill.

**17.Experimental Design:** Understanding how to design experiments and conduct A/B testing for making data-driven decisions.

In week four I got to learn about the Roadmap for learning python and relation between the NumPy and pandas.

In week five I got to learn about the NumPy and Pandas and about its operations in detail.

**About need of relevant Internship in career development:**

**1.Hands-on Experience:** Internships provide an opportunity to gain practical, real-world experience in your chosen field. It allows you to apply the knowledge and skills acquired in your academic studies to actual projects and tasks.

**2.Skill Development:** Internships help you develop and refine essential skills specific to your industry. Whether it's technical skills like programming, data analysis, or soft skills like communication and teamwork, internships offer a valuable platform for growth.

**3.Networking Opportunities:** Internships provide a chance to build connections with professionals and experts in your field. Networking during an internship can lead to mentorship, future job opportunities, and recommendations.

**4.Industry Insight:** Working in a real work environment provides valuable insights into the industry's dynamics, culture, and best practices. It helps you understand the day-to-day operations and challenges faced by professionals in the field.

**5.Resume Building:** Internship experiences enhance your resume and make you stand out to potential employers. Employers often value candidates with practical experience, and having relevant internships can significantly boost your chances of landing a full-time job.

**6.Confidence Building:** Successfully completing an internship can boost your confidence in your abilities and make you feel more prepared for future career challenges.

**7.Learning Company Culture:** Internships allow you to experience the culture of different organizations, helping you determine which work environment aligns best with your values and career goals.

**8.Exploring Career Paths:** Internships can help you clarify your career goals and interests. They provide an opportunity to try out different roles and industries before committing to a long-term career path.

**9.References and Recommendations:** A successful internship can lead to strong references and recommendations from supervisors and colleagues, which can be valuable assets when applying for jobs in the future.

**10.Transition to Full-Time Job:** Some internships can lead to full-time job offers if the employer is impressed with your performance during the internship period. It can be a direct pathway to starting your career with the same company.

**11.Personal Growth:** Internships offer valuable opportunities for personal growth and self-discovery. They challenge you to adapt to new situations, take on responsibilities, and develop a sense of professionalism.

**Brief about Quiz Game:-**

A quiz game program in Python is a fun and interactive application that presents a series of questions to the user and allows them to answer them. The program keeps track of the user's score based on their correct answers and provides feedback at the end. Here's a brief outline of how you can create a simple quiz game program in Python:

**1.Set Up the Questions:** Define a list or dictionary containing the quiz questions along with their corresponding correct answers.

**2.Welcome Message:** Display a welcome message to greet the user and explain the rules of the quiz.

**3.Quiz Loop:** Use a loop to iterate through the list of questions and present each question to the user one by one.

**4.User Input:** Accept the user's input for each question. You can use the input() function to get the user's answer.

**5.Check Answers:** Compare the user's answer with the correct answer for each question and keep track of the score.

**6.Display Results:** After the quiz is complete, display the user's score and provide feedback on their performance.

**Opportunity given by USC/UCT:**

I just want to Thank to each and everyone, who have helped you directly or indirectly for successfully completing this project as well as the report.

Also I want to thank UpSkill Campus and The IOT Academy for providing me such an interesting internship opportunity and also I want to thank the mentors who guided and helped throughout the learning duration.

# Introduction

## About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various**Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end**etc.



1. UCT IoT Platform **(****)**

**UCT Insight** is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

* It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
* It supports both cloud and on-premises deployments.

It has features to  
• Build Your own dashboard  
• Analytics and Reporting  
• Alert and Notification  
• Integration with third party application(Power BI, SAP, ERP)  
• Rule Engine

1. **Smart Factory Platform (****)**

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

* with a scalable solution for their Production and asset monitoring
* OEE and predictive maintenance solution scaling up to digital twin for your assets.
* to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
* A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

1.  based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

1. Predictive Maintenance

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



## About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.





Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

<https://www.upskillcampus.com/>

upSkill Campus aiming to upskill 1 million learners in next 5 year



## The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

## Objectives of this Internship program

The objective for this internship program was to

 ☛ get practical experience of working in the industry.

 ☛ to solve real world problems.

 ☛ to have improved job prospects.

 ☛ to have Improved understanding of our field and its applications.

 ☛ to have Personal growth like better communication and problem solving.

## Reference

[1] Google:- https://www.google.com/

[2] Anaconda {A python compiler}

[3] Chrome:- <https://www.google.com/intl/en_in/chrome/>

[4] Jupyter NoteBook

# Problem Statement:-

Given below is the explanation of the project Quiz Game:

This code is a simple quiz game implemented using the Tkinter library in Python. The game allows the user to select a topic (e.g., Geography or Mathematics) and then presents a series of multiple-choice questions related to the chosen topic. The user can answer the questions and submit their responses to get their final score. Here's a step-by-step explanation of the code:

1. Importing libraries: The code starts by importing the required libraries: **tkinter** for creating the GUI, **simpledialog** for creating a dialog box to get user input, **messagebox** for displaying messages in dialog boxes, and **random** for choosing a random topic.
2. **QuizGame** class: The core functionality of the quiz game is implemented within this class.
3. **\_\_init\_\_** method: This method initializes the **QuizGame** class with the main window (**root**). It sets up the initial state of the game, including the window's title, dimensions, and background color. It also defines the available topics, questions, choices, and correct answers for each topic in the **self.topics** dictionary.
4. **create\_widgets** method: This method creates the GUI elements, including buttons, labels, and radio buttons, and packs them into the main window (**root**).
5. **select\_topic** method: This method is called when the user clicks the "Select Topic" button. It prompts the user to enter the desired topic in a dialog box. If a valid topic is entered, it sets the **current\_topic** attribute, resets the current question index (**current\_question**) and the score to 0, and calls the **update\_question** method to display the first question.
6. **update\_question** method: This method updates the GUI to show the current question and choices based on the selected topic. It sets the question text to the **question\_label** and creates radio buttons for each choice in the current question. The user can then select one of the choices.
7. **check\_answer** method: This method is called when the user clicks the "Submit" button. It checks if a topic is selected and if the selected answer is correct. If the answer is correct, the user's score is incremented. The method then moves to the next question and updates the GUI with the next question using **update\_question**. If all questions for the current topic are answered, it calls **show\_result** to display the final score.
8. **show\_result** method: This method displays a dialog box with the final score of the user, showing the number of correct answers out of the total number of questions for the selected topic.
9. **main** section: The code then enters the main loop by creating a **tk.Tk()** instance (the main window), creating an instance of the **QuizGame** class, and starting the GUI event loop with **root.mainloop()**.

When you run the code, it will open a window displaying the "Select Topic" button. Upon clicking the button, you can enter a topic (e.g., "Geography" or "Mathematics"). Once a topic is chosen, the quiz will begin, and the questions will be displayed one by one with multiple choices. The user can select one choice for each question and submit their answers. After all questions are answered, a dialog box will show the final score.

# Existing and Proposed solution

Provide summary of existing solutions provided by others, what are their limitations?

What is your proposed solution?

What value addition are you planning?

## Code submission (Github link):

https://github.com/Karishma-jaiswal07/upskill\_campus.git

## Report submission (Github link) :

## https://github.com/Karishma-jaiswal07/upskill\_campus.git

# Proposed Design/ Model

Given more details about design flow of your solution. This is applicable for all domains. DS/ML Students can cover it after they have their algorithm implementation. There is always a start, intermediate stages and then final outcome.

## High Level Diagram:-

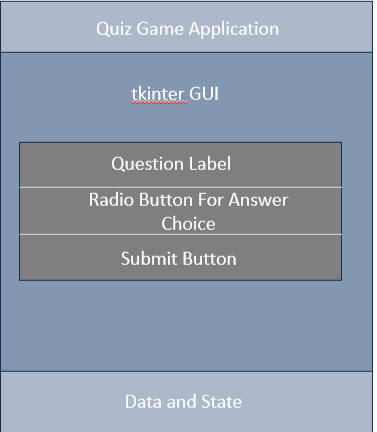


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

# Performance Test:-

Performance testing for the quiz game code involves assessing how well the application performs under different conditions, such as handling a large number of users, managing concurrent users, and response time during heavy usage. Since the quiz game is a simple desktop application, traditional performance testing techniques like load testing or stress testing may not be as applicable.

Performance Test Scenarios:

1. Responsiveness Test:

* Test Scenario: Evaluate how quickly the GUI responds to user interactions, such as selecting an answer or clicking the "Submit" button.
* Metrics: Measure the response time for each user action.

1. Memory Usage Test:

* Test Scenario: Monitor the memory consumption of the application during a quiz session to ensure it remains within acceptable limits.
* Metrics : Memory usage (RAM) of the application during the entire quiz session.

1. Question Rotation Test:

* Test Scenario: Verify that the quiz does not repeat questions within a single session.
* Metrics: Number of unique questions displayed in a single session.

## Test Plan/ Test Cases:-

A test plan outlines the overall approach and objectives of testing a software application, while test cases are specific scenarios or situations that are designed to validate the correct functioning of individual features or components. The objective is to verify the functionality , correctness and user experience of the quiz game application.

1. Test Scope:

* Test the GUI functionality, including question presentation, answer selection, and scoring.
* Test the correctness of scoring based on the user's answers.
* Verify the rotation of questions and that no question is repeated within a session.
* Check the responsiveness and user interface design.

1. Test Environment:

* Python 3.x environment with tkinter library installed.

1. Test Execution :

* Manual testing: Testers will manually run the application and interact with the GUI to validate its behavior.
* Exploratory testing: Testers will explore the application to identify any unexpected behavior or issues.

1. Test Cases:

* Verifying Correct Questions and Answers Display:

Description: Check if the quiz displays the correct question and answer choices.

Steps: Run the quiz application and observe the questions and answers choices displayed on the GUI .

Expected Result: The first question and its corresponding answer choices should be displayed correctly.

* Verify User Answer Selection:

Description: Check if the user can select an answer for the displayed question.

Steps: Run the quiz application and Click on the radio buttons to select an answer.

Expected Result: The selected radio button should be highlighted.

* Verifying Scoring Calculation:

Description: Check if the quiz scores the user's answers correctly.

Steps : Run the quiz application , answer all the questions with known correct answers and click the submit button.

Expected Result: The quiz should calculate the score based on the correct answers, and the final score should be displayed.

## Test Procedure:-

* Test Setup
* Test Cases Executions
* Observations
* Record Results
* Defect Reporting
* Debugging and Fixing
* Retesting
* Regression Testing
* User Acceptance Testing
* Final Evaluation

## Performance Outcome:-

* **Code Efficiency:** Assess the efficiency of the code, especially when dealing with larger question sets or more complex functionalities.
* **Scoring:** The score calculation should be accurate and based on the correct answers provided by the user.
* **Functionality:** The quiz game should function correctly, presenting questions, allowing users to select answers, and providing feedback on their responses.
* **User Interface:** The performance outcome is related to how user-friendly and visually appealing the GUI is. A well-designed and intuitive interface can enhance user engagement and satisfaction.
* **Question Rotation:** If the quiz is intended to be repeated or have multiple sessions, it should handle question rotation correctly, avoiding repetition of questions in the same session.
* **Quiz Completion:** The quiz should recognize when all questions have been answered and display the final score or a completion message.
* **Responsiveness:** The GUI should be responsive and not freeze during question processing or score calculation.

# My learnings:-

My overall experience during the internship:

* **Practical Experience:** Internships provide hands-on experience in a real-world work environment, allowing you to apply theoretical knowledge from your studies to practical tasks.
* **Skill Development:** During an internship, I had the opportunity to develop and refine various skills, including technical skills, problem-solving, and time management.
* **Project Experience:** Working on real projects during an internship gave me a portfolio of work that can showcase to potential employers.
* **Problem Solving Skills:** Internships expose me to various challenges and problem-solving opportunities, enabling me to develop critical thinking and analytical skills.
* **Feedback and Mentorship:** Internships often provide feedback and mentorship from experienced professionals, helping me to identify areas for improvement and learn from seasoned experts.
* **Confidence Building:** Successfully completing an internship boosts your self-confidence and reassures me that I can excel in your chosen field.
* **Continuous Learning:** Internships often shows the areas where I need to improve or gain more knowledge, motivating me to pursue continuous learning and professional development.

# Future work scope:-

Python has a several potentials future scope and enhancements that can make it more feature rich and appealing . Here are some ideas to consider for improving the quiz game:

* **Multiple Choice Questions:** Currently ,the quiz game takes user input as text. You can expand it to include multiple-choice questions where users can select an option (A, B, C, etc.) as their answer.
* **Timed Quiz:** Implement a timer for each question, allowing users a limited time to answer. This adds an element of challenge and excitement to the game.
* **Levels and Categories:** Introduce different levels of difficulty or categories for questions. Users can choose their preferred level or category, and the questions become more tailored to their interests and knowledge.
* **Leaderboard:** Add a feature to store and display high scores on a leaderboard. This encourages users to compete and try to achieve the highest score.
* **Lifelines or Hints:** Include lifelines or hints that users can use during the quiz to help them answer challenging questions.
* **Sounds and Graphics:** Enhance the user experience with sound effects, animations, and graphics to make the game more visually appealing.
* **Database Integration :** Store the quiz questions and user scores in a database to enable persistence and easy management of questions.
* **Users Accounts:** Implement user account functionality, allowing users to create profiles, save progress, and continue the quiz at a later time.
* **Online Multiplayer:** Transform the game into an online multiplayer experience, where multiple users can compete in real-time.
* **Randomized Questions :** Shuffle the questions randomly for each playthrough to keep the quiz fresh and prevent memorization.
* **Feedback and Explanations:** Provide detailed explanations for correct answers or feedback on incorrect answers, helping users learn from their mistakes.
* **Social Sharing:** Allow users to share their quiz results on social media platforms, increasing engagement and exposure.
* **Internationalization**: Make the quiz game multilingual to cater to users from different regions and languages.
* **Machine Learning-based Difficulty Adjustment:** Use machine learning algorithms to adjust the difficulty of questions based on users' performance, ensuring an optimal level of challenge.
* **Voice Assistant Integration:** Enable integration with voice assistants like Siri or Google Assistant, allowing users to play the quiz using voice commands.