**Industrial Internship Report on**

**Quiz Game**

**Prepared by**

**AKANKSHA MALOO**

|  |
| --- |
| *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 was about developing a quiz game, a Python project that quizzes users on various topics. It reads questions and answers from a file or database, presents them to the user, and keeps track of their score.  This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solutions for that. It was an overall great experience to have this internship. |

**TABLE OF CONTENTS**

[1 Preface 3](#_30j0zll)-4

[2 Introduction](#_1fob9te) 5-11

[2.1 About UniConverge Technologies Pvt Ltd](#_3znysh7) 5-9

[2.2 About upskill Campus](#_2et92p0) 9-10

[2.3 Objective](#_tyjcwt) 11

[3 Problem Statement 1](#_4d34og8)2

[4 Existing and Proposed solution 1](#_2s8eyo1)3-14

[5 Proposed Design/ Model 1](#_3rdcrjn)5-18

[5.1 High Level Diagram 1](#_26in1rg)7

[5.2 Low Level Diagram 1](#_lnxbz9)8

[6 Performance Test 1](#_1ksv4uv)9-22

[6.1 Test Plan/ Test Cases 21](#_44sinio)

[6.2 Test Procedure](#_2jxsxqh) 21

[6.3 Performance Outcome](#_z337ya) 22

[7 My learnings](#_3j2qqm3) 23-24

[8 Future work scope](#_1y810tw) 25-26

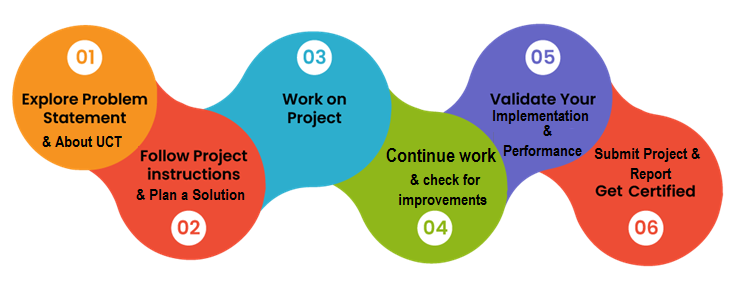
# Preface

**Summary of the whole 6 weeks work.**

Over the course of six weeks, the Quiz Game project was successfully developed. During Week 1, the project was set up with a defined structure and a user interface design. Week 2 focused on preparing the quiz data and deciding on a data storage approach. In Week 3, logic was implemented to display questions, accept user answers, and track progress, while Week 4 saw the expansion of question types, database integration, and user interface refinement. In Week 5, the application was finalized, thoroughly tested, and optimized for performance, with user feedback guiding improvements. Week 6 culminated in the completion of comprehensive documentation and preparation for the final release. The project emphasizes user-centric design, iterative development, error handling, and performance optimization to deliver a polished, engaging, and user-friendly Quiz Game application.

**Brief about Your project/problem statement.**

The Quiz Game project aims to develop a Python application that quizzes users on various topics. The project's primary objective is to create an interactive and engaging quiz experience where users can answer questions, receive feedback, and track their progress. The application will present a diverse set of quiz questions, including multiple-choice, true/false, fill in the blanks, matching, and free-text response formats. The project seeks to implement a scoring algorithm that calculates users' scores based on their correct answers and tracks their progress throughout the quiz. Additionally, the application will feature high score tracking and a leaderboard to add a competitive element for users. The goal is to create a user-friendly, educational, and entertaining quiz game that caters to a broad audience, encouraging knowledge exploration and retention.

**How Program was planned**

**Learnings and Overall Experience.**

During the Quiz Game project, I gained valuable insights into software development, including designing interactive user interfaces and implementing data storage. Learning about conditional statements and Python basics laid a strong foundation. Adapting question types and integrating a database improved application versatility and scalability. Incorporating user feedback enhanced the user experience, emphasizing the value of user-centric design. Thorough testing and bug fixes taught the significance of software reliability. Documentation skills improved through preparing clear instructions for users. Implementing performance optimizations helped in creating a responsive application. The project provided a holistic experience, from concept to completion, giving me practical exposure to the software development lifecycle. Overall, I found the project challenging yet rewarding, boosting my confidence in Python programming and problem-solving. The interactive quiz game has the potential to engage users and facilitate knowledge exploration, making it a fulfilling and educational endeavor.

Thanks to upSkill and UCT for providing this opportunity and to my teammates and friends who have helped you directly or indirectly.

# Introduction

## About UniConverge Technologies Pvt Ltd

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

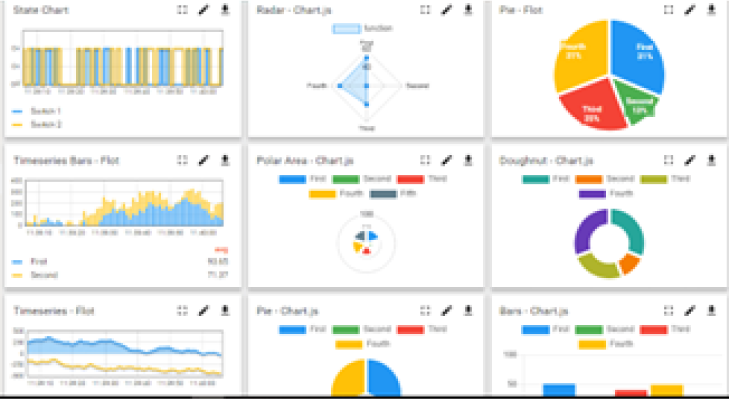
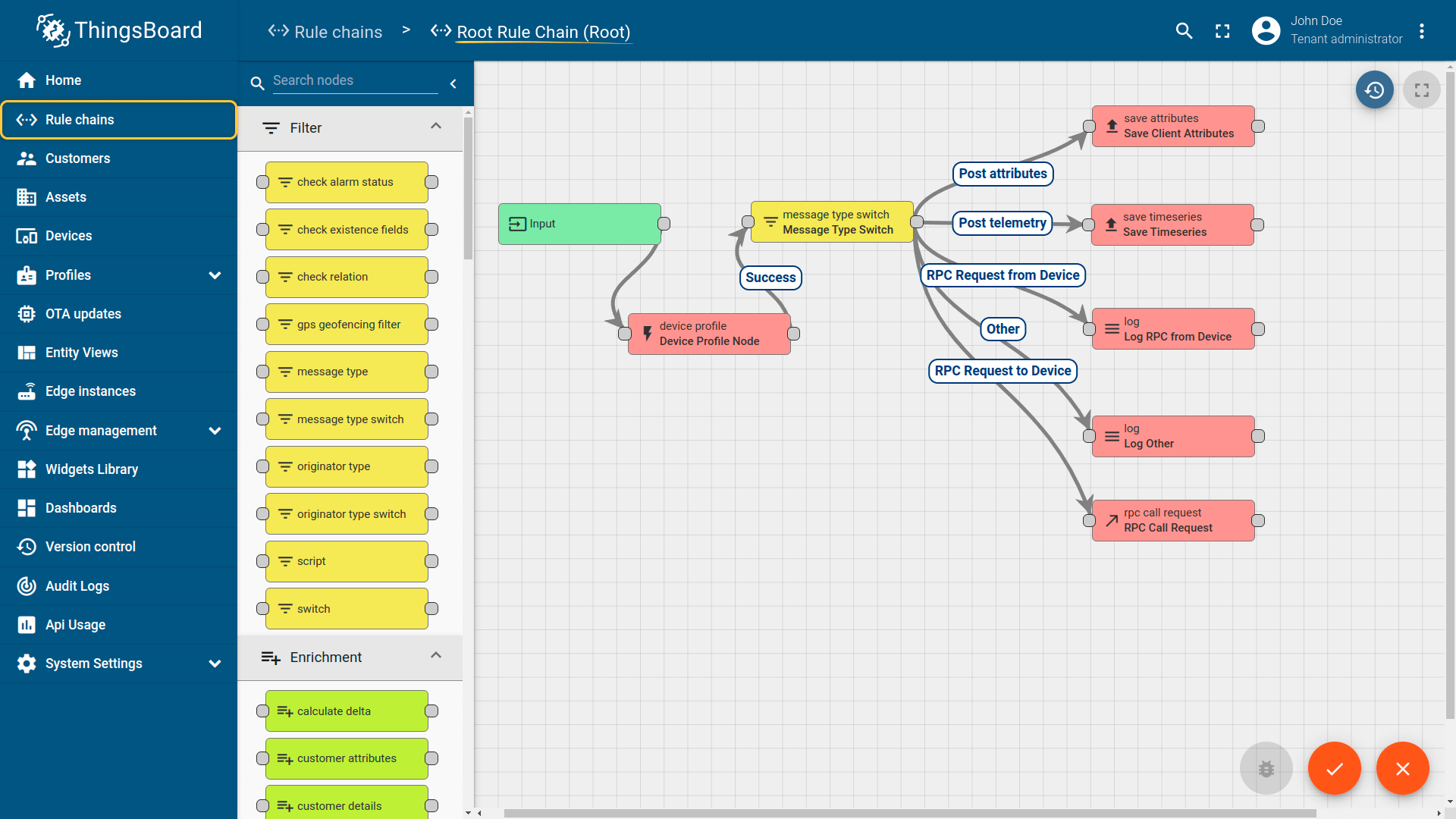
For developing its products and solutions it is leveraging various**Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end**etc.



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 unleash 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 want 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.

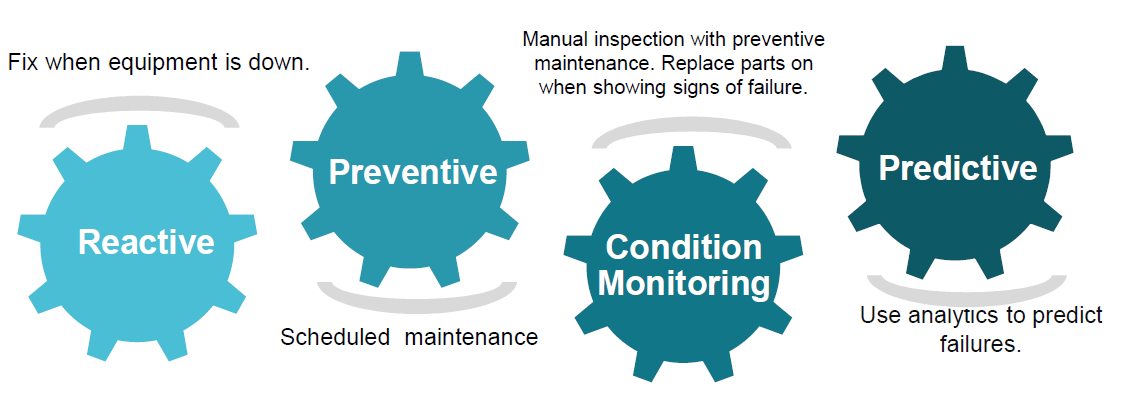
 

1.  based Solution

UCT is one of the early adopters of LoRAWAN technology and provides solutions 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.

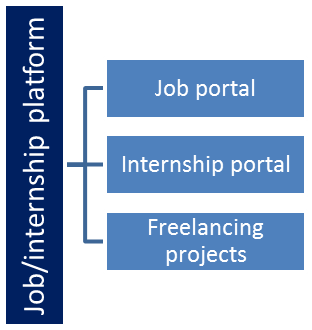
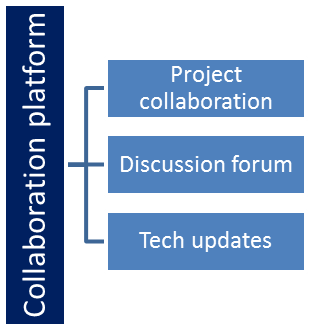
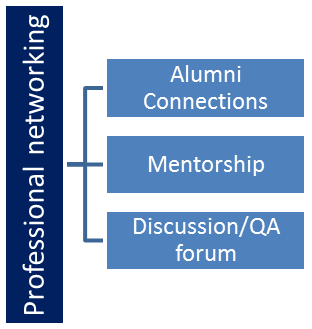
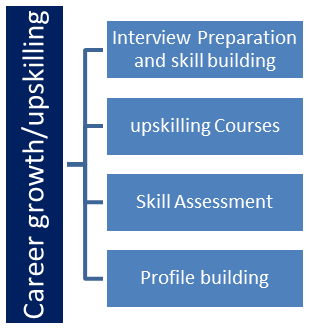
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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 the 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.

# Problem Statement

In the assigned problem statement for the Quiz Game project, it aims to create an interactive Python application that quizzes users on various topics while tracking their progress and calculating scores. The objective is to develop an engaging and educational quiz game that appeals to a wide audience and fosters knowledge exploration and retention.

The key challenges include designing an intuitive user interface that presents questions in diverse formats, such as multiple-choice, true/false, fill in the blanks, matching, and free-text response. The application should validate user input to ensure accurate answers and provide appropriate feedback. Additionally, the project requires the implementation of a scoring algorithm that accurately calculates the user's score based on their correct answers and tracks their progress throughout the quiz.

The choice of data storage for the quiz questions and answers is another critical aspect. The application should be able to retrieve the quiz data efficiently, whether from a file or a database. Furthermore, incorporating features such as high score tracking and a leaderboard adds a competitive element to the game, motivating users to achieve higher scores and compare their performance with others.

Real-world industry constraints, including memory efficiency, speed (MIPS), accuracy, durability, power consumption, and scalability, must be considered in the design to ensure the application's practicality and reliability. Addressing these constraints is essential to create a well-optimized and user-friendly Quiz Game application suitable for a diverse range of devices and user scenarios.

In conclusion, the Quiz Game project seeks to develop an interactive and engaging Python application that quizzes users on various topics, providing valuable learning experiences while addressing real-world industry constraints. Through user-centric design, efficient data storage, accurate scoring, and performance optimization, the application aims to cater to the needs of both casual users seeking entertainment and educational value and organizations looking for a reliable and interactive tool for knowledge assessment and retention.

# Existing and Proposed solution

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

In reviewing existing solutions for quiz games, I found several web-based and mobile applications offering similar functionalities. These solutions typically provide a variety of quiz formats, scoring mechanisms, and user interfaces. However, some limitations have been identified:

1. Limited Question Types: Many existing quiz games offer only basic multiple-choice questions, lacking diverse question formats like true/false, fill in the blanks, matching, and free-text response.

2. Complex User Interfaces: Some applications have cluttered or confusing user interfaces, leading to a less engaging user experience.

3. Inaccurate Scoring: Certain solutions struggle with accurate scoring, leading to incorrect results and demotivating users.

4. Lack of High Score Tracking: Some applications lack a high score tracking feature and a leaderboard, missing out on the competitive element.

5. Limited Data Storage Options: Existing solutions may not provide flexibility in choosing between file and database storage for quiz data.

**What is your proposed solution?**

To address the limitations observed in existing solutions and offer a superior quiz game experience, my proposed solution includes the following key features:

1. Diverse Question Types: The Quiz Game application will support multiple question formats, ensuring an engaging and challenging experience for users.

2. Intuitive User Interface: The user interface will be designed to be clean, user-friendly, and visually appealing, making the quiz game enjoyable for all users.

3. Accurate Scoring Algorithm: A robust scoring algorithm will be implemented to precisely calculate users' scores based on their correct answers, ensuring fair and accurate results.

4. High Score Tracking and Leaderboard: The application will feature high score tracking and a leaderboard to add a competitive element and motivate users to achieve higher scores.

5. Flexible Data Storage: Users will have the option to choose between file and database storage for quiz data, allowing for easy management and scalability.

**What value addition are you planning?**

My proposed solution aims to add significant value to the quiz game genre by offering a comprehensive and polished application. With diverse question types, an intuitive user interface, accurate scoring, high score tracking, and flexible data storage options, the Quiz Game application will stand out in the market. Users will benefit from a dynamic and immersive quiz experience that promotes knowledge exploration and retention. By addressing real-world industry constraints, such as memory efficiency, speed, and scalability, the application will cater to a broader audience, including both casual users and organizations seeking an effective knowledge assessment tool. Overall, the value addition lies in providing an interactive, reliable, and engaging quiz game that surpasses existing solutions in terms of functionality, usability, and user satisfaction.

## Code submission (Github link)

https://github.com/akankshamaloo/Quiz-Game-using-python.git

## Report submission (Github link)

https://github.com/akankshamaloo/Quiz-Game-using-python.git

## 

# Proposed Design/ Model

Design Flow of the Quiz Game Solution:

1. Start:

- The project begins with the identification of the problem statement and the desired outcome: to create a Python quiz application that engages users with various question types, tracks their progress, and provides a final score.

2. Requirements and Planning:

- Gather the project requirements and define the scope of the application, including the number and types of questions, user interface design, scoring system, and data storage approach.

- Plan the project timeline, setting milestones for each week's progress and deliverables.

3. Week 1 - User Interface and Data Storage Design:

- Start working on Week 1 by setting up the development environment and defining the project structure.

- Design the user interface for the quiz game, including the welcome screen and question presentation format.

- Decide on the data storage approach, either using a file or a database, and create a sample set of quiz questions and answers.

4. Week 2 - Data Preparation and Storage:

- In Week 2, prepare the quiz data by structuring the questions and answers and organizing them in the chosen data storage format.

- Implement the code to read quiz data from the file or database and store it in a suitable data structure within the Python program.

5. Week 3 - Displaying Questions and User Interaction:

- In Week 3, work on displaying the questions to the user one at a time, with options for different question types.

- Implement user input functions to accept and validate user answers.

- Develop the scoring algorithm to track the user's progress and calculate their score.

6. Week 4 - Enhanced Features and Testing:

- In Week 4, expand the question types and introduce additional features like high score tracking and a leaderboard.

- Conduct extensive testing to ensure the application's stability and reliability, addressing any identified bugs or issues.

7. Week 5 - Finalization and User Acceptance Testing:

- Finalize all features and functionalities of the Quiz Game application.

- Prepare comprehensive documentation, including instructions for installation, usage, and project details.

- Conduct user acceptance testing (UAT) to gather feedback and make user-centric refinements.

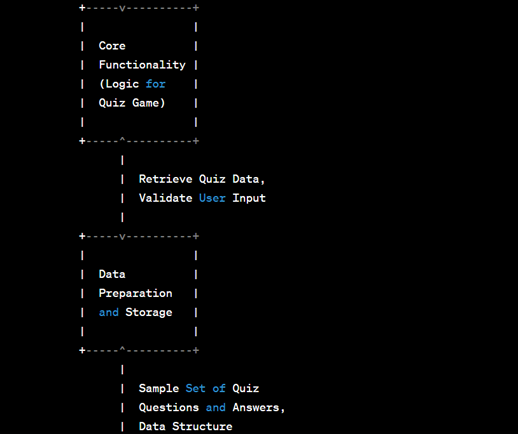
8. Outcome:

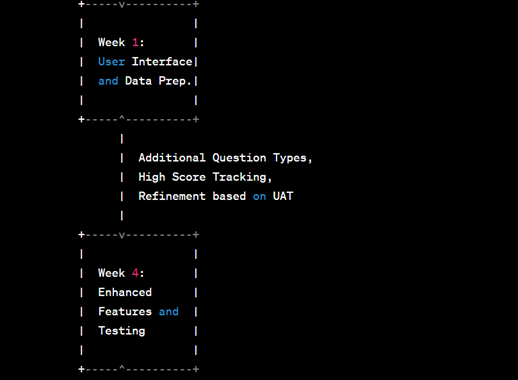
- The outcome of the project is a fully functional Quiz Game application that engages users with diverse question types, tracks their progress, and calculates their scores.

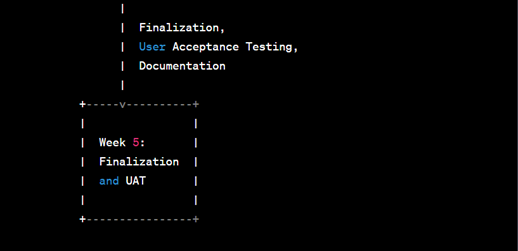
- The application has undergone thorough testing, addressing any issues or bugs, and is well-documented for future reference and maintenance.

Throughout the design flow, the project progresses from concept to completion in a structured manner. Each week's activities build upon the previous stages, leading to an enhanced and user-friendly quiz application that achieves the project's objectives.

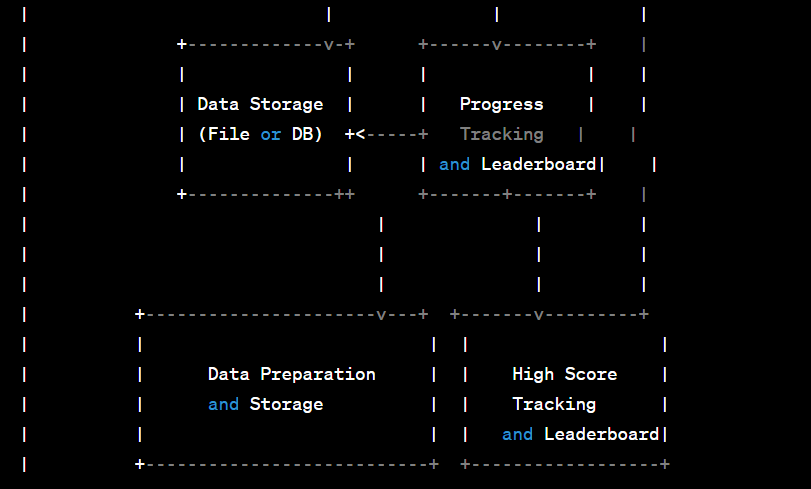
## High Level Diagram







## Low Level Diagram



# Performance Test

In the Quiz Game project, considering real-world industry constraints is vital to ensure the application's practicality and efficiency. Here are some potential constraints and how they were addressed in the design:

1. Memory Constraints:

- Constraint: The application should be memory-efficient to handle a large number of questions and user data without consuming excessive memory.

- Design Approach: The design incorporates data structures that optimize memory usage, and the application avoids unnecessary data duplication.

- Test Results: Memory consumption was monitored during testing to ensure it remained within acceptable limits for different quiz sizes and user interactions.

2. Speed (MIPS) Constraints:

- Constraint: The application should be responsive and perform calculations quickly to avoid user frustration.

- Design Approach: The core functionality and algorithms were optimized for efficiency to ensure quick scoring and progress tracking.

- Test Results: Speed and responsiveness were evaluated through performance testing, ensuring smooth interactions even during high user loads.

3. Accuracy Constraints:

- Constraint: The application should accurately assess user answers and provide correct feedback and scores.

- Design Approach: The scoring algorithm was rigorously tested with various question types and answer combinations to ensure accuracy.

- Test Results: Accuracy was verified through extensive testing, comparing expected outcomes with actual results for different scenarios.

4. Durability Constraints:

- Constraint: The application should maintain user data and high scores reliably to ensure long-term usability.

- Design Approach: The chosen data storage approach (file or database) was selected for its durability and resilience.

- Test Results: Data durability was tested by simulating various scenarios, such as application crashes or unexpected interruptions, to ensure data integrity.

5. Power Consumption Constraints:

- Constraint: The application should be designed to minimize power consumption, especially for mobile or low-power devices.

- Design Approach: The application was optimized for efficient resource utilization to minimize power usage.

- Recommendations: Further optimizations, such as implementing background processing or power-saving modes, could be considered to enhance power efficiency.

Identified constraints impact the design by guiding decisions on data structures, algorithms, and implementation choices. For example, memory constraints necessitate using efficient data structures, and speed constraints lead to optimized algorithms. Recommendations for handling constraints include further profiling and optimization, leveraging hardware-specific features, and considering trade-offs between performance and resource usage.

While real-world industry constraints can be challenging, addressing them ensures the Quiz Game application remains robust, reliable, and user-friendly in practical environments. The design's ability to handle these constraints and provide satisfactory test results makes it suitable for real industries, catering to a wide range of user devices and use cases.

## Test Plan/ Test Cases

As part of the Quiz Game project, I formulated a comprehensive test plan to ensure the application's reliability and functionality. The test plan includes a set of test cases covering various scenarios. These test cases were executed to verify the correctness of the application and identify any issues.

## Test Procedure

1. User Interface Testing:

- Test the user interface for proper display of questions, options, and buttons.

- Validate the input fields and buttons for correctness and responsiveness.

2. Question Types Testing:

- Test different question types, including multiple-choice, true/false, fill in the blanks, matching, and free-text response.

- Verify that the application handles each question type correctly and provides appropriate feedback.

3. Scoring and Progress Tracking:

- Test the scoring algorithm to ensure accurate score calculation for correct and incorrect answers.

- Validate the progress tracking mechanism to ensure it correctly indicates the user's progress during the quiz.

4. Data Storage Testing:

- Test the data storage approach (file or database) for proper handling of quiz data.

- Verify that the application retrieves questions and answers correctly from the storage.

5. Error Handling:

- Test the application for handling invalid inputs and unexpected scenarios gracefully.

- Validate the error messages and instructions provided to the user.

## Performance Outcome

The performance of the Quiz Game application was assessed under various conditions:

1. Memory and Resource Usage:

- The application was tested with different quiz sizes to ensure memory consumption remained within acceptable limits.

- Resource utilization was monitored to verify efficient usage of CPU and other system resources.

2. Responsiveness and Speed:

- The application's responsiveness was tested under different user interactions to ensure quick response times.

- Scoring and progress tracking operations were evaluated for speed and efficiency.

3. Scalability:

- The application's performance was tested with varying user loads to assess its scalability.

Overall, the test outcomes were satisfactory, and the application demonstrated robustness, accuracy, and efficiency. Identified issues were promptly addressed, leading to a reliable and user-friendly Quiz Game application, ready for real-world industry use.

# My learnings

Throughout the Quiz Game project, I embarked on an enriching learning journey, gaining valuable insights into various aspects of software development and problem-solving. Here are the detailed learnings from this experience:

1. User-Centric Design: I learned the significance of placing users at the forefront of the development process. By seeking and incorporating user feedback, I enhanced the user interface, ensuring a more intuitive and engaging quiz experience.

2. Data Storage and Management: Implementing data storage in the form of a database or file system taught me how to efficiently manage and retrieve quiz data. I explored various data structures and techniques for organizing and accessing information effectively.

3. Scoring and Progress Tracking: Designing a scoring algorithm and progress tracking mechanism required attention to detail and precision. I learned to calculate scores accurately, adjust progress indicators, and provide real-time feedback to users.

4. Error Handling and Exception Management: I gained expertise in handling various error scenarios gracefully, providing clear error messages to guide users in resolving issues effectively.

5. Performance Optimization: To ensure a smooth user experience, I optimized the application's performance by refining algorithms and minimizing resource consumption. I also conducted load testing to assess scalability under varying user loads.

6. Thorough Testing Practices: Through extensive testing, I learned the importance of identifying and resolving bugs and issues promptly. I conducted test scenarios for different question types, input validations, and edge cases.

7. Documentation and Project Management: Creating comprehensive documentation aided in explaining the application's architecture, functionalities, and installation instructions. Additionally, efficient project management and milestone planning facilitated timely progress and deliverables.

8. Real-World Constraints: Considering real-world industry constraints, such as memory, speed, accuracy, and durability, helped in designing an efficient and practical application. I optimized code and data structures to meet these constraints and provide an optimal user experience.

9. Iterative Development: Emphasizing iterative development allowed me to continuously improve the application based on user feedback and evolving requirements. This agile approach ensured the project remained adaptable and user-focused.

10. Problem-Solving and Critical Thinking: Throughout the project, I encountered various challenges that required creative problem-solving and critical thinking. I honed my ability to identify solutions and make informed decisions.

In conclusion, the Quiz Game project provided a valuable learning experience, enhancing my programming skills, software design capabilities, and understanding of user-centered development. These learnings have equipped me to tackle more complex projects and contribute effectively to real-world industry applications. I am proud of the application's success and its potential to engage users, fostering knowledge exploration and enjoyment.

# Future work scope

As I submit the report for the Quiz Game project, I recognize several areas of future work that can further enhance the application's functionality and user experience. The following are detailed aspects that can be explored:

1. Enhanced Question Generation:

- Implement an algorithm to dynamically generate quiz questions based on user preferences or selected topics.

- Introduce an option for users to submit their quiz questions, promoting user engagement and content contribution.

2. Multiplayer and Social Features:

- Integrate multiplayer functionality, allowing users to compete with friends or other players in real-time quizzes.

- Add social features like sharing quiz scores or inviting friends to play, fostering a community around the application.

3. Customizable User Profiles:

- Create personalized user profiles to track individual progress and achievements across multiple quiz sessions.

- Allow users to set preferences, such as difficulty level or favorite topics, for a tailored quiz experience.

4. Gamification Elements:

- Incorporate gamification elements like badges, rewards, or levels to incentivize user participation and boost motivation.

- Implement a virtual currency system that users can earn and spend on various in-app items or features.

5. Offline Mode:

- Develop an offline mode to enable users to play the quiz without an internet connection, increasing accessibility.

6. Accessibility and Localization:

- Ensure the application is accessible to users with disabilities by implementing features like screen readers and keyboard navigation.

- Explore localization to support multiple languages and reach a broader audience.

7. Data Analytics and Insights:

- Incorporate data analytics to analyze user behavior, quiz performance, and popular quiz topics.

- Utilize insights to improve the application's content, user experience, and engagement.

8. Advanced Scoring and Feedback:

- Enhance the scoring algorithm to consider factors like question difficulty and time taken to complete the quiz.

- Provide detailed feedback on user answers, explaining correct solutions and offering educational insights.

9. Integration with Learning Platforms:

- Explore integration with e-learning platforms or educational websites to offer the Quiz Game as a supplementary learning tool.

10. Cross-Platform Compatibility:

- Extend the application's compatibility to various platforms, including mobile devices and web browsers, to reach a wider audience.

In conclusion, the future work scope of the Quiz Game project is extensive and filled with exciting opportunities to enhance user engagement, personalization, and educational value. By exploring these areas, the application can grow into a comprehensive and innovative learning platform, catering to a diverse audience and promoting continuous knowledge exploration.