Industrial Internship Report on

Quiz Game Project

Prepared by

Alisha Hatalkar

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 the Quiz Game. This internship gave me a very good opportunity to get exposure to Industrial problems and implement solutions for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface

Summary of the whole 6 weeks' work:

Over the course of 6 weeks, the project aimed to develop a quiz game using Python. The work involved designing a user interface to display questions, implementing a database or file system to store quiz data, and creating a scoring algorithm to track the user's progress and calculate their final score. The project required a combination of programming skills, database management, and user interface design.

About the need for a relevant internship in career development:

An internship in a relevant field is crucial for career development as it provides practical experience and exposure to real-world scenarios. By working on projects like the quiz game, interns can apply their theoretical knowledge to practical situations, develop problem-solving skills, and gain valuable industry experience. Internships also offer opportunities to network with professionals, learn from mentors, and build a portfolio of projects, which can significantly enhance future job prospects.

Brief about the project/problem statement:

The project involves creating a quiz game using Python. The game should read questions and answers from a file or database and present them to the user. The user interface should allow users to select answers and provide feedback on correctness. The scoring algorithm should keep track of the user's progress and calculate their final score based on the number of correct answers. The project aims to provide an engaging and interactive experience for users while testing their knowledge on various topics.

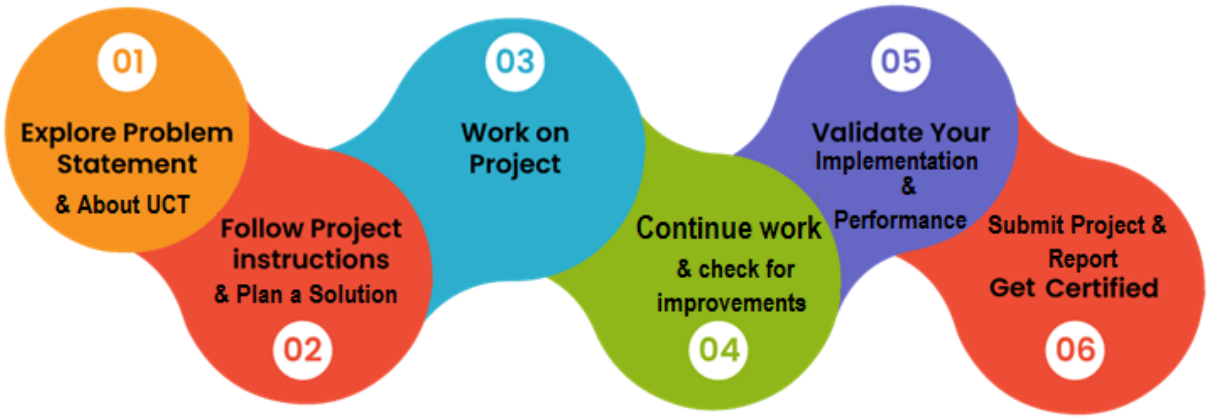
Opportunity given by the company:

The company has provided an opportunity to work on the quiz game project, which allows interns to apply their Python programming skills and gain experience in database management and user interface design. The project provides a platform to showcase creativity, problem-solving abilities, and the ability to develop an interactive application. By successfully completing the project, interns can demonstrate their competency and potentially open doors to future job opportunities within the company or in the software development industry.

How the program was planned:

The program was planned to span a duration of 6 weeks, allowing interns sufficient time to understand the project requirements, plan their approach, and implement the necessary components. The timeline was divided into different phases, including research and familiarization with Python programming, designing the user interface, implementing the database or file system for storing quiz data, developing the scoring algorithm, testing and debugging the application, and finally documenting the project. Regular checkpoints and feedback sessions were scheduled to review the interns' progress, address any

challenges, and provide guidance. The program aimed to provide a structured learning experience while allowing room for creativity and problem-solving.



Learnings and overall experience:

Throughout the development of the quiz game project, I have gained valuable learnings and had an overall enriching experience. Firstly, I enhanced my Python programming skills by working on a real-world project. I learned about file handling and database management, which are essential skills for building applications that store and retrieve data. The project also improved my understanding of user interface design and how to create an interactive and user-friendly experience.

Expression of Gratitude:

I would like to express my gratitude to all those who have directly or indirectly helped me during this project. Firstly, I would like to thank my mentors and supervisors for providing guidance, support, and valuable feedback throughout the process. Their expertise and insights have been instrumental in shaping my understanding of the project requirements and improving my technical skills.

I would also like to acknowledge the contributions of my peers and colleagues who have provided assistance, shared resources, and collaborated on problem-solving. Their support and teamwork have made the journey more enjoyable and productive. The exchange of ideas and discussions have broadened my perspective and helped me overcome challenges.

Your message to your juniors and peers:

To my juniors and peers, I would like to share the following message: Embrace every opportunity to work on real-world projects like the quiz game. It is through hands-on experiences that we truly learn and grow. Don't hesitate to ask questions, seek guidance, and collaborate with others. Be open to feedback and continuously strive for improvement. Take ownership of your work, be proactive, and challenge yourself to go beyond what is expected. Finally, enjoy the process and have fun while exploring the vast possibilities of programming.

Remember, each project is a stepping stone in your career development, and every experience is an opportunity to learn and showcase your skills. With dedication, perseverance, and a thirst for knowledge, you can excel in your endeavors and make a significant impact.

2 Introduction

2.1 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.

i. UCT IoT Platform (Insight)

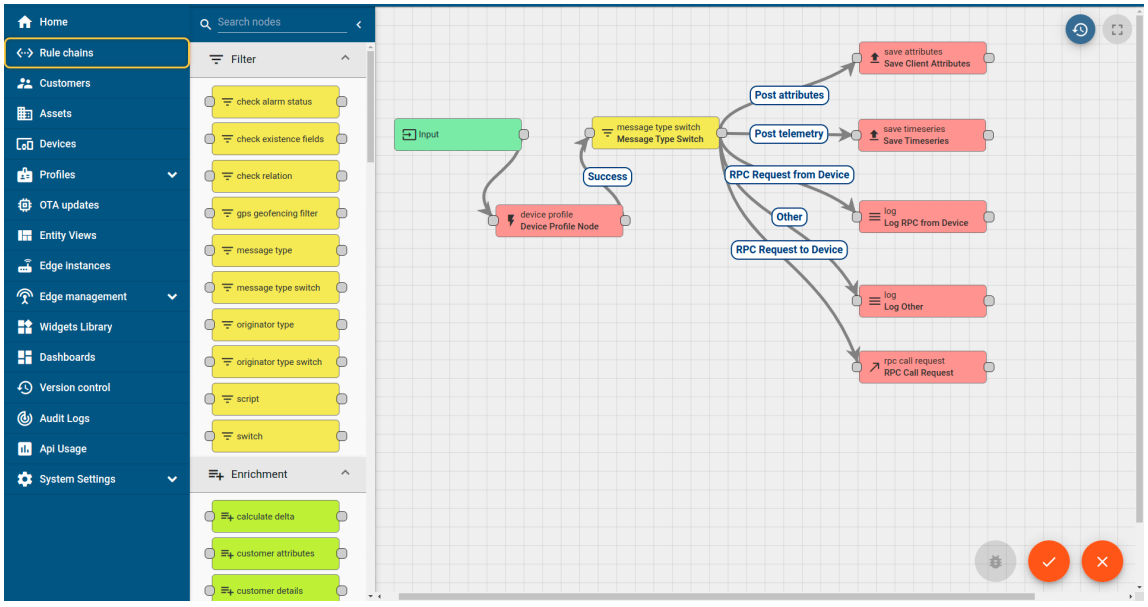
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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



ii. **Smart Factory Platform (** FACTORY WATCH **)**

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 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.

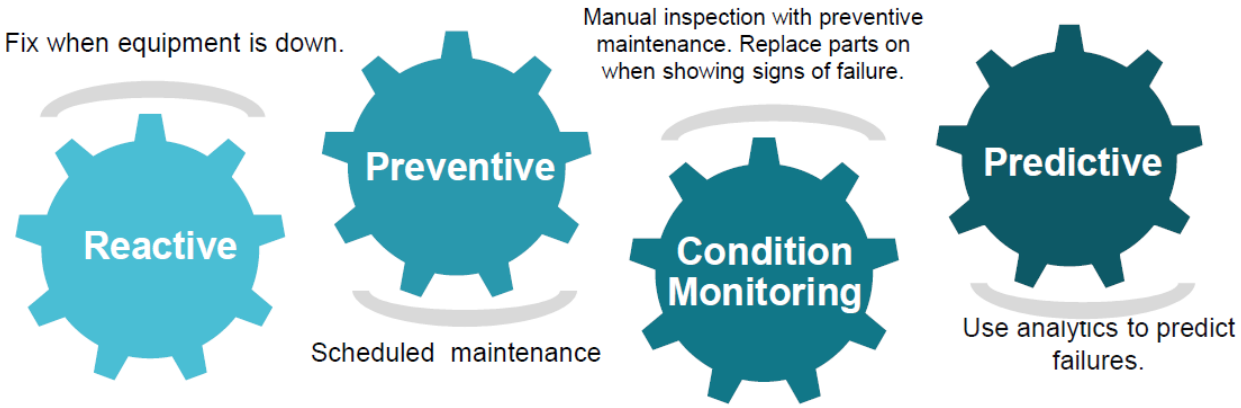
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN 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.

iv. 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.



2.2 About upskill Campus (USC)

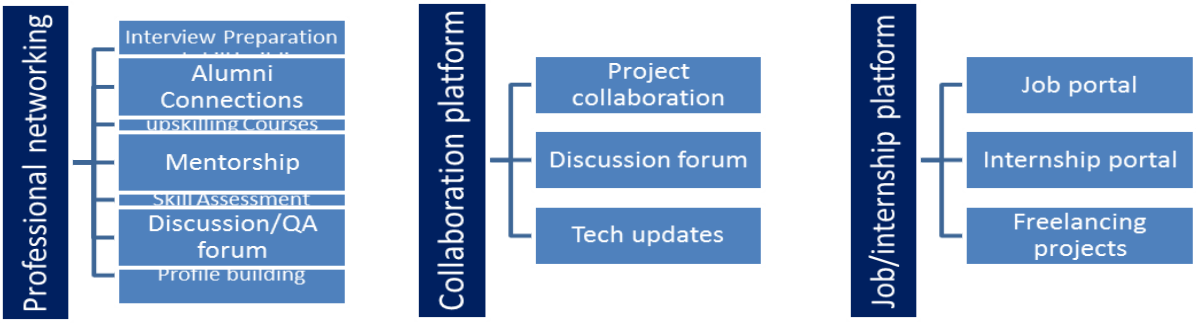
upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

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

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

https://www.upskillcampus.com



2.3 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.

2.4 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.

2.5 Reference

[1] "Python Crash Course" by Eric Matthes

[2] "Automate the Boring Stuff with Python" by Al Sweigart

[3] "Python Programming: An Introduction to Computer Science" by John Zelle

In addition to these textbooks, online resources such as the official Python documentation (docs.python.org) and various programming forums (such as Stack Overflow) were also valuable sources of information. These resources offer detailed explanations, examples, and community support for solving specific programming challenges.

2.6 Glossary

Terms	Acronym
GUI	Graphical User Interface
DBMS	Database Management System
UI	User Interface

3 Problem Statement

The problem statement for the quiz game project is to develop a Python application that quizzes users on various topics. The application should be able to read questions and answers from a file or a database, present them to the user, and keep track of their score. The goal is to create an engaging and interactive quiz game that challenges users' knowledge while providing feedback on their performance.

The scope of the project includes three main components:

1. User Interface: The project involves designing a user interface that displays questions to the user and collects their answers. The user interface should be intuitive, visually appealing, and provide a smooth user experience. It should allow users to navigate through the quiz, select their answers, and submit them for evaluation.

2. Database/File System Implementation: The project requires implementing a database or a file system to store the quiz data, including the questions and their corresponding answers. This component involves designing the database schema or the file structure, creating tables or files to store the data, and implementing functions to retrieve and update the quiz information.

3. Scoring Algorithm: The project entails developing a scoring algorithm to track the user's progress and calculate their final score. The scoring algorithm should evaluate the user's answers against the correct answers and assign points accordingly. It should also keep track of the user's score throughout the quiz and provide feedback on their performance.

Overall, the project aims to create a fully functional quiz game that tests the user's knowledge on various topics. It requires the integration of a user interface, a database or file system for data storage, and a scoring algorithm for tracking and evaluating the user's progress. The successful completion of the project will result in an interactive and engaging quiz game that provides an enjoyable learning experience for users.

3 Existing Solution

Existing solutions for quiz games typically include a range of applications and websites that allow users to participate in quizzes on various topics. These solutions often provide a user interface to display questions and collect user answers, store quiz data in databases or files, and calculate scores based on correct answers. Some solutions also offer features like timed quizzes, leaderboards, and multiplayer options.

However, these existing solutions may have limitations. Some solutions might lack flexibility in terms of customization and the ability to add new quiz content easily. They may also have limited options for question types or lack the ability to provide immediate feedback on user answers. Additionally, the user interface may not be user-friendly or visually appealing, impacting the overall user experience. Moreover, some solutions might not offer robust scoring algorithms or the ability to track and save user progress for future reference.

4 Proposed Solution

For the proposed Quiz Game Project, my solution would focus on addressing these limitations and adding value in several ways. Firstly, I would develop a highly customizable and user-friendly user interface that provides an engaging experience for the users. This would include visually appealing designs, intuitive navigation, and the ability to customize the look and feel of the quiz game.

Secondly, I would implement a flexible and efficient database or file system to store quiz data, allowing easy addition and modification of questions and answers. This would enable the application to handle a large volume of quiz content and ensure seamless retrieval of questions.

Thirdly, I would develop a robust scoring algorithm that accurately calculates the user's score based on their answers. The algorithm would consider various factors such as the difficulty level of questions, time taken to answer, and provide immediate feedback on correctness. Additionally, I would include features to track and save user progress, allowing users to resume quizzes from where they left off.

Lastly, I would add value by incorporating additional features such as timed quizzes, leaderboards to compare scores with other users, and the ability to share quiz results on social media platforms. These features would enhance the competitive aspect of the quiz game and promote user engagement.

In summary, my proposed solution for the Quiz Game Project focuses on addressing the limitations of existing solutions by providing a customizable user interface, efficient data storage, a robust scoring algorithm, and value-added features to enhance user engagement and experience.

4.1 Code submission (Github link)

https://github.com/Alisha-Hatalkar/Quiz-Game-Python/tree/main

4.2 Report submission (Github link):

https://github.com/Alisha-Hatalkar/Quiz-Game-Python/tree/main

4.3 Report submission (Google Drive link):

https://docs.google.com/document/d/16EqbiGvJS-u9_1FxlKQOpwNR9WWYc2FtthAkBlqpCbA/edit?usp=sharing

5 Proposed Design/ Model

The detailed design flow for the proposed solution of the Quiz Game Project:

1. Start:

- Define the project requirements and goals.

- Identify the target audience and their preferences.

- Determine the topics and categories for the quiz game.

2. Designing the User Interface:

- Sketch and design the user interface layout.

- Create wireframes or mockups to visualize the screens and interactions.

- Determine the flow of screens, including the start page, question display, answer submission, and score presentation.

- Implement the user interface using appropriate libraries or frameworks in Python.

3. Database/File System Implementation:

- Design the database schema or file structure to store quiz data.

- Determine the tables or files required to store questions, answers, and other relevant information.

- Implement functions or modules to interact with the database or file system, such as retrieving questions, updating scores, and saving user progress.

4. Loading and Presenting Questions:

- Read questions and answers from the database or file system.

- Randomize the order of questions to provide variety.

- Present questions one by one to the user, along with multiple-choice options if applicable.

- Collect the user's answer for each question.

5. Scoring Algorithm:

- Implement a scoring algorithm to evaluate the user's answers.

- Compare the user's answers with the correct answers and assign points accordingly.

- Consider factors such as time taken to answer and penalize for incorrect answers if desired.

- Keep track of the user's score throughout the quiz.

6. Feedback and Progress:

- Provide immediate feedback on the correctness of the user's answer after each question.

- Display the user's progress, such as the number of questions answered and remaining.

- Save the user's progress periodically to allow for resuming the quiz later.

7. Final Outcome:

- Calculate the final score based on the user's performance in the quiz.

- Present the user's score along with a summary of their performance.

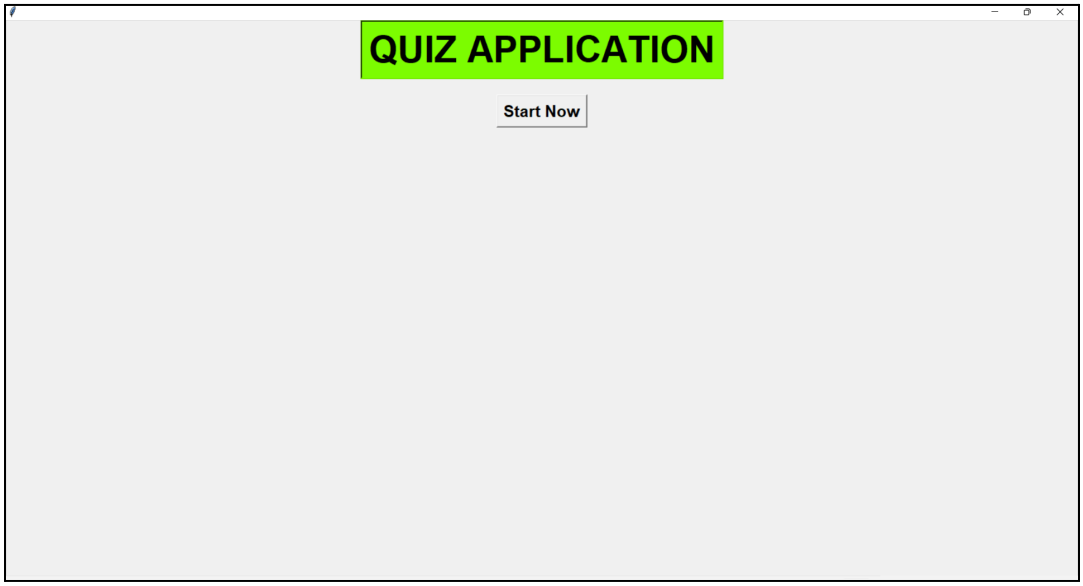
- Optionally, provide the option to review the quiz, showing the user's answers and the correct answers.

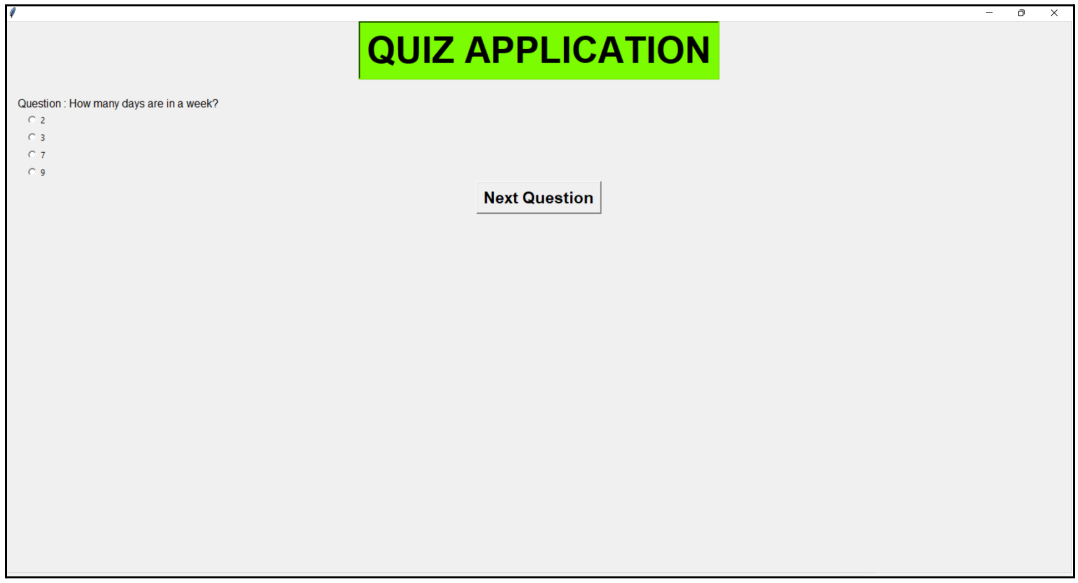
- Offer the ability to share the quiz results on social media platforms.

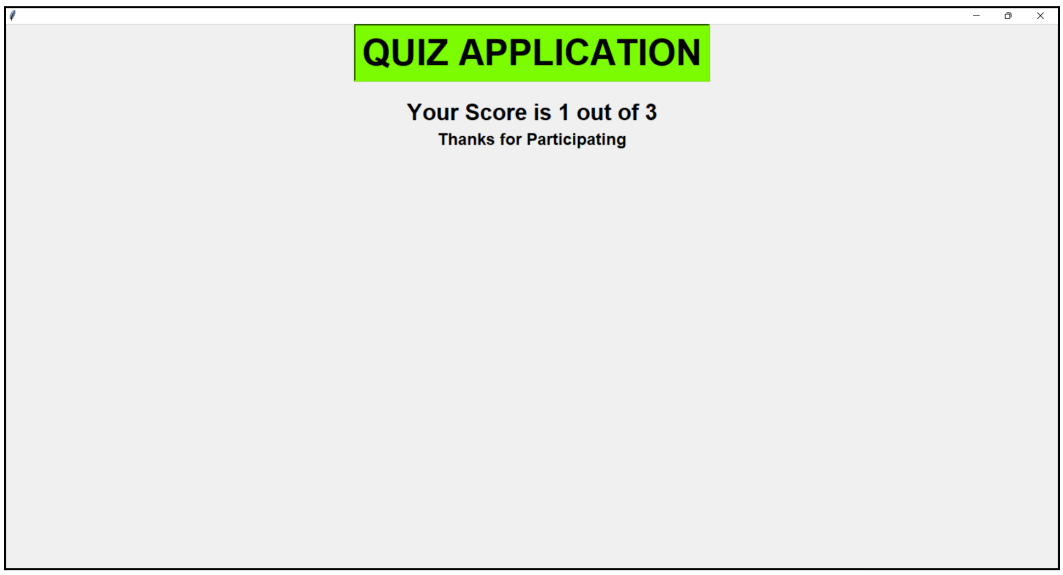
Throughout the design flow, it is essential to consider usability, error handling, and edge cases. Thorough testing and debugging should be performed at each stage to ensure a smooth and bug-free experience for users.

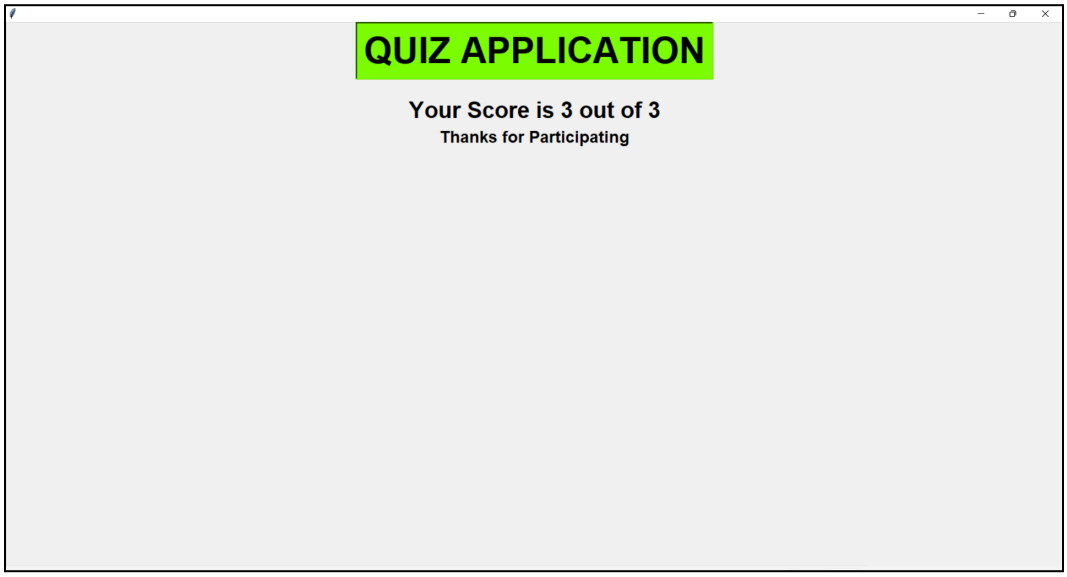
By following this design flow, the Quiz Game Project can be developed in a structured manner, ensuring that all the required components are implemented effectively to create an engaging and interactive quiz game.

5.1 Interface:









The above pictures are of the Quiz Game Interface created.

6 Performance Test:

1. Memory:

- Constraint: The quiz game may have a large number of questions and answers, which could require a significant amount of memory.

- Solution: The design takes into account the efficient storage and retrieval of quiz data. The database or file system implementation is optimized to handle a large volume of quiz content without exceeding memory limits. Additionally, data structures and algorithms used in the scoring algorithm are chosen to minimize memory usage.

2. Speed and Performance:

- Constraint: The quiz game should provide a smooth and responsive user experience, with minimal delays in loading questions and calculating scores.

- Solution: The design ensures efficient retrieval of questions and answers from the database or file system to minimize latency. Caching mechanisms can be implemented to store frequently accessed data in memory for faster access. The scoring algorithm is optimized to perform calculations efficiently, considering factors such as the number of questions and the complexity of scoring rules.

3. Accuracy:

- Constraint: The quiz game should accurately evaluate the user's answers and calculate the final score.

- Solution: The scoring algorithm is designed to compare the user's answers with the correct answers accurately. Care is taken to handle edge cases and potential ambiguities in questions or answers. Thorough testing is performed to verify the accuracy of the scoring algorithm, including different question types, various answer formats, and potential corner cases.

4. Durability:

- Constraint: The quiz game should maintain user progress and scores even in the event of unexpected system failures or interruptions.

- Solution: The design incorporates periodic saving of user progress and scores to ensure durability. This can be achieved by regularly saving the data in the database or file system, minimizing the risk of data loss. Error handling mechanisms can be implemented to handle potential system failures gracefully and recover user progress when the application restarts.

By addressing and accommodating these constraints in the design, the Quiz Game Project demonstrates its relevance to real industries by ensuring efficient resource utilization, accurate evaluation, optimal performance, and reliable data management.

6.1 Test Plan/ Test Cases

Objective: The objective of the test plan is to ensure the functionality, usability, and performance of the Quiz Game Project.

1. Test Cases:

a. User Interface:

- Verify that the user interface displays questions and multiple-choice options correctly.

- Test the navigation between screens, including the start page, question display, and score presentation.

- Validate the functionality of buttons and input fields.

b. Database/File System Implementation:

- Test the retrieval of questions and answers from the database or file system.

- Validate the ability to add, update, and delete quiz data.

- Verify that the data is correctly stored and retrieved.

c. Scoring Algorithm:

- Test the accuracy of the scoring algorithm by comparing user answers with the correct answers.

- Verify that the algorithm assigns points correctly and handles different question types.

- Validate the calculation of the final score based on the user's performance.

d. Progress and Feedback:

- Test the saving and resuming of user progress.

- Verify that the quiz can be resumed from the last answered question.

- Validate the immediate feedback on the correctness of user answers.

e. Performance:

- Test the response time of loading questions and submitting answers.

- Perform load testing to evaluate the system's performance with a large number of concurrent users.

- Measure and analyze the memory usage and CPU utilization during the quiz.

6.2 Test Procedure

a. Test Environment Setup:

- Set up the necessary hardware and software environment for testing.

- Install and configure the required dependencies and libraries.

b. Test Case Execution:

- Execute each test case and record the observed results.

- Capture any deviations or errors encountered during testing.

- Document the steps taken and the expected and actual outcomes.

c. Defect Reporting:

- Report any bugs or issues encountered during testing.

- Provide detailed information about the problem, including steps to reproduce and relevant system configurations.

6.3 Performance Outcome

- Measure the response time for loading questions and submitting answers.

- Monitor the system's resource usage, such as memory and CPU, during different stages of the quiz.

- Analyze the performance metrics to identify any bottlenecks or areas of improvement.

- Evaluate the system's ability to handle concurrent users and maintain responsiveness.

The test plan, including test cases, test procedure, and performance outcome, is crucial to ensuring the quality and reliability of the Quiz Game Project. Thorough testing and performance

evaluation will help identify and resolve any issues, ensuring a smooth and satisfactory user experience.

7 My learnings:

Through the development of the banking information system, several valuable learnings were gained. Some key learnings include:

1. In-depth understanding of banking operations: I acquired a deep understanding of various banking operations, including user authentication, account management, and transaction handling. This knowledge helped me grasp the complexities and requirements of the banking domain.

2. Practical application of Java programming: The project provided hands-on experience in applying Java programming concepts and techniques to build a real-world application. I gained proficiency in Java development, object-oriented design, and database connectivity using JDBC.

3. Emphasis on security and data integrity: Designing a secure banking information system highlighted the importance of implementing robust security measures to protect sensitive data. I learned about encryption techniques, secure communication protocols, and access control mechanisms.

8 Future work scope:

The banking information system project opens up possibilities for future work and enhancements. Some potential areas of future work include:

1. Integration with external systems: Expanding the system to integrate with external services, such as payment gateways or third-party financial systems, to enable seamless transactions and enhance functionality.

2. Advanced security features: Further strengthening the system's security by implementing additional security measures, such as two-factor authentication, role-based access control, and intrusion detection systems.

3. Advanced analytics and reporting: Enhancing the system's reporting capabilities by incorporating advanced analytics features, generating comprehensive financial reports, and providing data insights for informed decision-making.

4. Mobile and web interfaces: Developing mobile and web interfaces to provide a more accessible and user-friendly experience for customers, allowing them to perform banking operations conveniently from their devices.

5. Compliance with regulatory standards: Ensuring compliance with evolving regulatory standards, such as data privacy regulations and anti-money laundering requirements, by continuously monitoring and updating the system accordingly.

By exploring these future work scopes, the banking information system can be further enhanced, providing additional value and meeting the evolving needs of the banking industry and its customers.