Industrial Internship Report on

QUIZ APP

Prepared by

Karren S

Punnose

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 **"Quiz Game Application using Python and SQL Server"**.

My project was "Quiz game web application" that allowed users to take timed quizzes, view scores, and get instant feedback. It focused on building an interactive and responsive interface using basic web technologies.

The backend was built using Microsoft SQL Server to store user data, quiz questions, and scores, while the frontend was developed using Python's Tkinter library for the GUI. Git and GitHub were used for version control and project deployment.

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

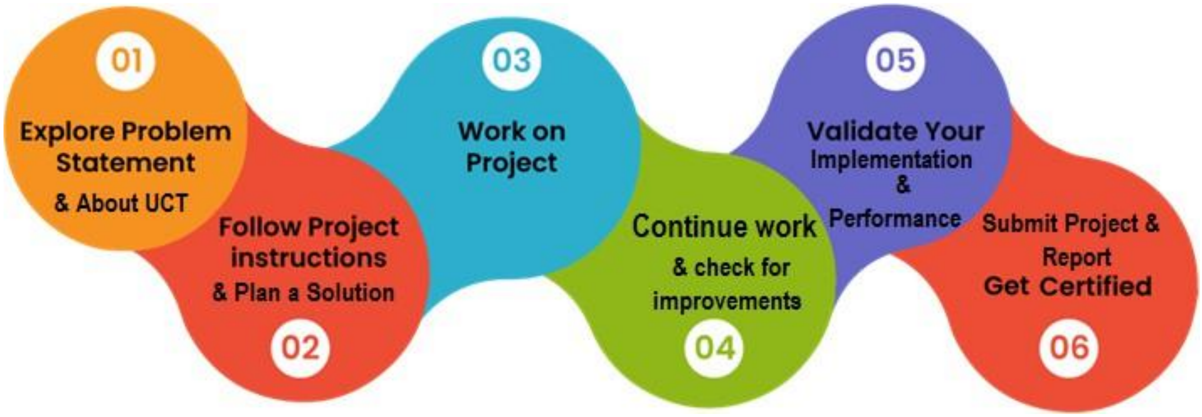
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development. Brief

about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

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.

uct

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

i. UCT IoT Platform (uct Insight)

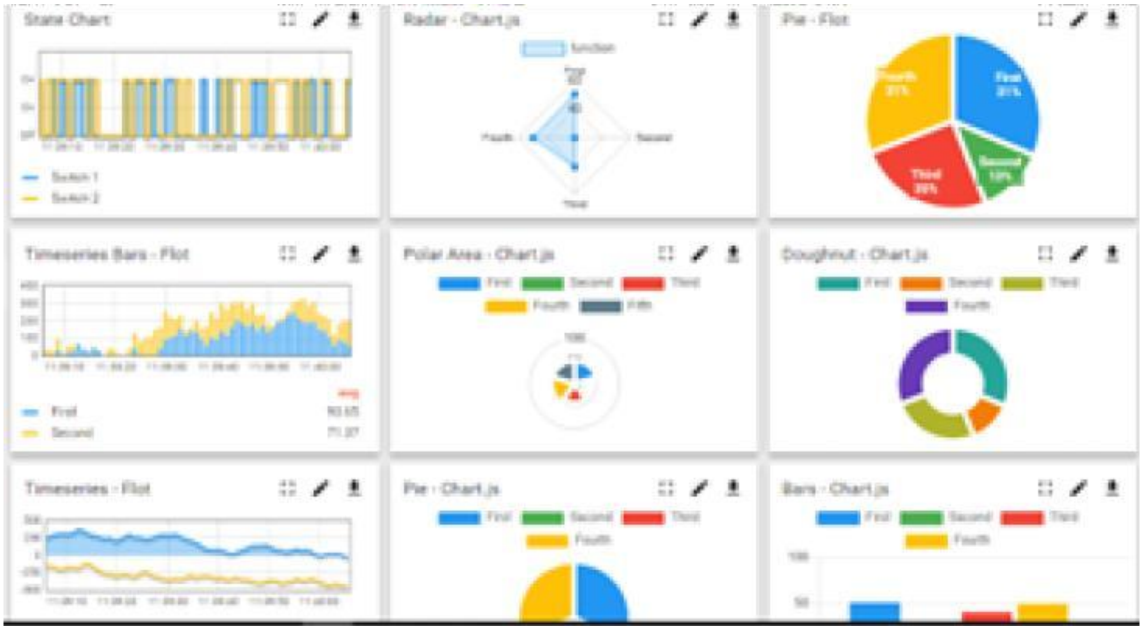
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

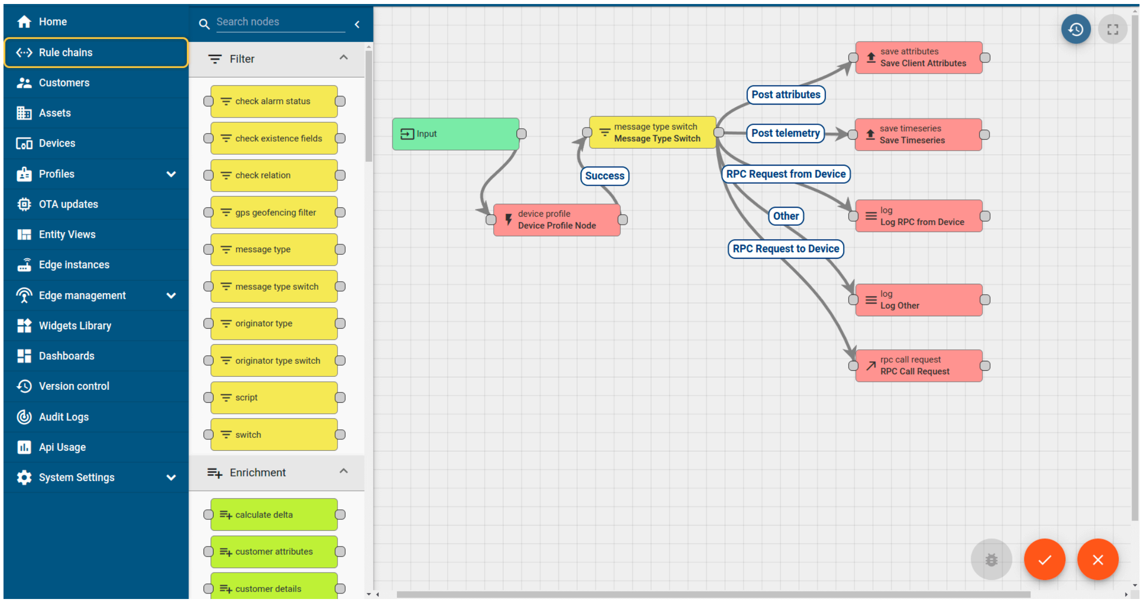
- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA

- 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


State Chart
Switch 1
Switch 2
Radar - Chart.js
function
First
Second
Third
Fourth
Pie - Flot
First 31%
Second 13%
Third 25%
Fourth 31%
Timeseries Bars - Flot
First
Second
Polar Area - Chart.js
First
Second
Third
Fourth
Fifth
Doughnut - Chart.js
First
Second
Third
Fourth
Timeseries - Flot
Pie - Chart.js
First
Second
Third
Fourth
Bars - Chart.js
First
Second
Third
Fourth


Home
Rule chains
Customers
Assets
Devices
Profiles
OTA updates
Entity Views
Edge instances
Edge management
Widgets Library
Dashboards
Version control
Audit Logs
Api Usage
System Settings
Search nodes
Filter
check alarm status
check existence fields
check relation
gps geofencing filter
message type
message type switch
originator type
originator type switch
script
switch
Enrichment
calculate delta
customer attributes
customer details
Input
device profile
Device Profile Node
Success
message type switch
Message Type Switch
Post attributes
Post telemetry
RPC Request from Device
Other
RPC Request to Device
save attributes
Save Client Attributes
save timeseries
Save Timeseries
log
Log RPC from Device
log
Log Other
rpc call request
RPC Call Request

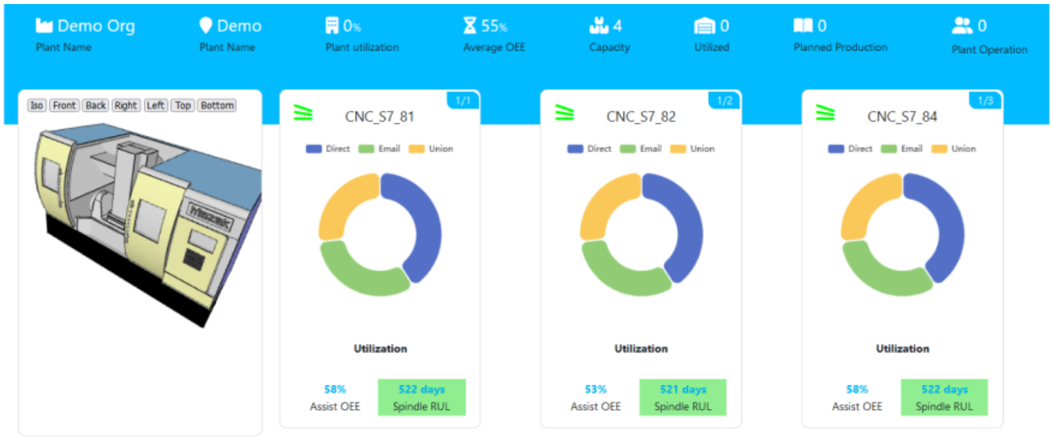
ii. Smart Factory Platform (FACTORY WATCH)

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 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.



Demo Org
Plant Name

Demo
Plant Name

0%
Plant utilization

55%
Average OEE

4
Capacity

0
Utilized

0
Planned Production

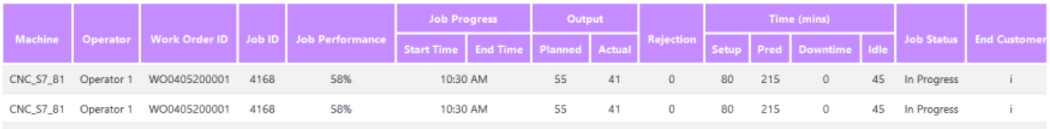
0
Plant Operation

Iso Front Back Right Left Top Bottom

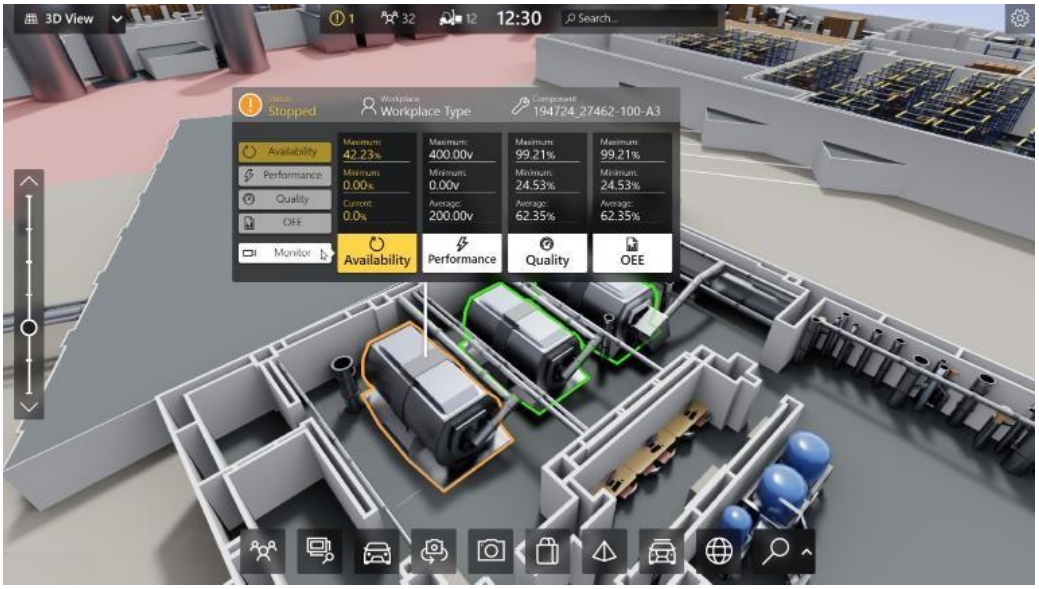
CNC_S7_81 1/1
Direct Email Union
Utilization
58% Assist OEE
522 days Spindle RUL

CNC_S7_82 1/2
Direct Email Union
Utilization
53% Assist OEE
521 days Spindle RUL

CNC_S7_84 1/3
Direct Email Union
Utilization
58% Assist OEE
522 days Spindle RUL



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



Status: Stopped
Workplace: Workplace Type
Component: 194724_27462-100-A3

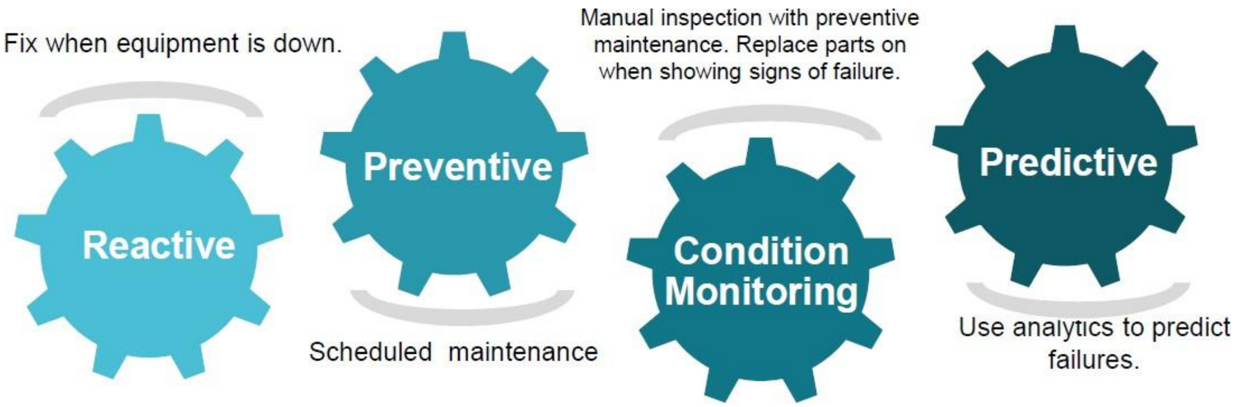
	Availability	Performance	Quality	OEE
	Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
	Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
	Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

iii. LoRaWAN 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.

iv. Predictive Maintenance

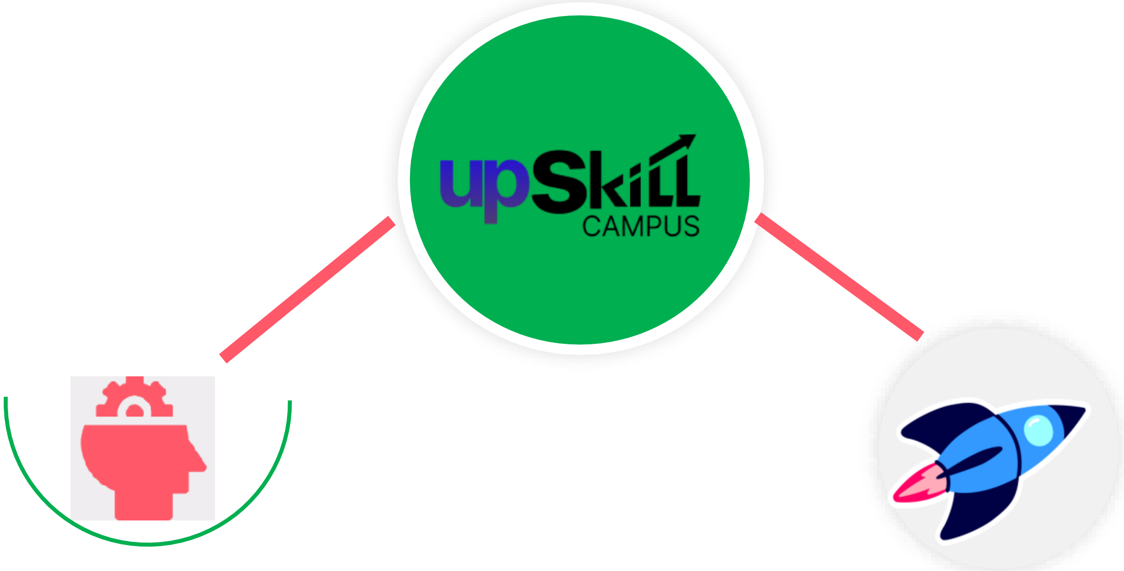
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

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.

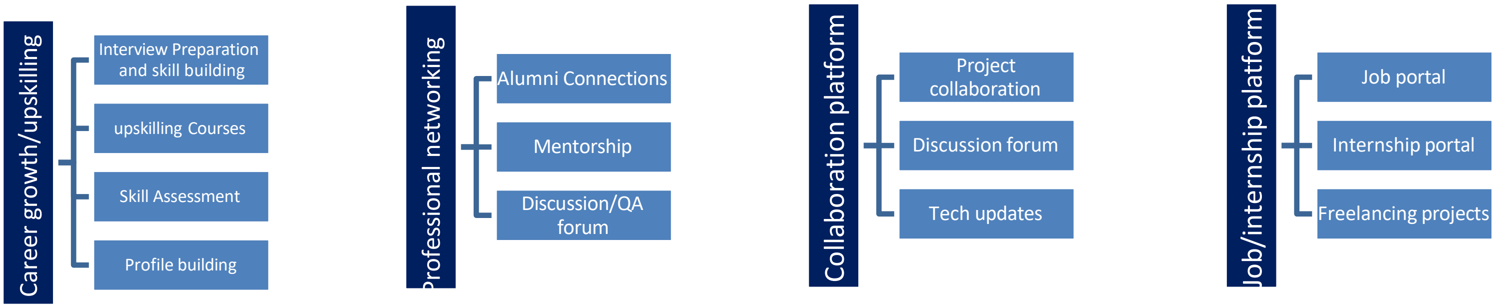
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/



- Career growth/upskilling
  - Interview Preparation and skill building
  - upskilling Courses
  - Skill Assessment
  - Profile building
- Professional networking
  - Alumni Connections
  - Mentorship
  - Discussion/QA forum
- Collaboration platform
  - Project collaboration
  - Discussion forum
  - Tech updates
- Job/internship platform
  - Job portal
  - Internship portal
  - Freelancing projects

2.3 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.

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] Microsoft Docs - SQL Server Documentation

[2] Python Docs - https://docs.python.org

[3] GitHub Docs - https://docs.github.com

2.6 Glossary

Terms	Acronym
Structured Query Language	SQL
Graphical User Interface	GUI
Integrated Development Environment	IDE
GitHub Repository	Repo
Uniform Resource Locator	URL

3 Problem Statement

In the assigned problem statement, I was responsible for developing a **Quiz Game Application** designed to offer an interactive and engaging platform for users to test their general knowledge. The application was required to incorporate key features such as **user registration**, **login authentication**, **quiz participation**, **dynamic score calculation**, and a **real-time leaderboard**. The system was built using **Python with Tkinter** for the graphical user interface and **SQL Server (SSMS)** for the backend database, ensuring that all user information, quiz questions, and scores were stored and retrieved efficiently. Additionally, the application had to function offline and be completed within a six-week timeline, with a daily recommended effort of three to four hours.

The core challenge involved integrating multiple components—frontend interface, backend logic, and database interaction—into a seamless and user-friendly software solution. I focused on ensuring smooth data flow, implementing secure authentication, building a responsive GUI, and establishing robust validation and scoring mechanisms. The system also needed to support multiple users and sessions while maintaining persistent storage for user performance data on the leaderboard.

This project emphasized practical application development and required a strong understanding of **modular coding**, **database design**, and **user experience principles**. All development was tracked through **Git version control**, and the final application was deployed on **GitHub**, enabling transparent progress tracking and laying the foundation for future enhancements.

4 Existing and Proposed solution

There are several existing solutions available in the form of online quiz platforms such as Google Forms, Kahoot, Quizizz, and other web-based tools. These platforms allow users to participate in quizzes, view results, and sometimes view rankings. They are often used in classrooms, corporate training, and online assessments. However, these systems rely heavily on internet connectivity and do not offer much flexibility in terms of customization or local deployment.

The limitations of these existing solutions include the need for continuous internet access, limited control over data management, and lack of offline usability. Additionally, they provide minimal customization of backend logic and database access, which restricts users from building or modifying quiz flows based on specific educational or institutional needs.

To address these challenges, the proposed solution is a desktop-based Quiz Game Application developed using Python (Tkinter) for the user interface and SQL Server for backend data management. The application is designed to run completely offline and supports user registration, login, quiz participation, automatic score calculation, and a leaderboard system. The local database setup ensures that user data, questions, and scores are stored securely and can be easily modified or extended as needed.

This solution adds value by providing complete control over the application and data, eliminating the dependency on internet connectivity, and offering a customizable structure suitable for institutions, learning environments, or self-assessments. The code and supporting documentation are hosted on GitHub for easy access, version control, and future improvements.

4.1 Code submission (Github link)

The complete source code for the Quiz Game Application, including the frontend (GUI using Python Tkinter), backend logic, and SQL Server database scripts, has been uploaded to the following GitHub repository:

GitHub Repository Link: https://github.com/Karren3/upskillcampus.git

4.2 Report submission (Github link) : https://github.com/Karren3/upskillcampus/tree/main

5 Proposed Design/ Model

The proposed design of the Quiz Game Application follows a structured step-by-step flow, ensuring clear separation between user interface, backend logic, and database operations. The system is developed using Python for the frontend GUI (Tkinter) and SQL Server for the backend database, with the entire flow centered around user interaction and data processing.

The design starts with the User Authentication Phase, where users can register or log in using a valid username and password. On successful login, users are redirected to the Quiz Module, which pulls multiple-choice questions from the database. Each question is displayed one at a time, and user responses are stored temporarily for evaluation.

In the Evaluation Phase, after all questions have been answered, the system calculates the total score based on the number of correct answers. This score is then stored in the backend under the user's profile. Once the score is recorded, the user is taken to the Leaderboard Module, where scores of all users are fetched from the database and ranked in descending order to show the top performers.

The final stage is the Logout Phase, where the user can safely exit the session and return to the login screen. The flow ensures data consistency, user security, and a smooth experience throughout the application. The modular nature of the design also makes it easier to maintain, update, and scale in the future.

5.1 High Level Diagram (if applicable)

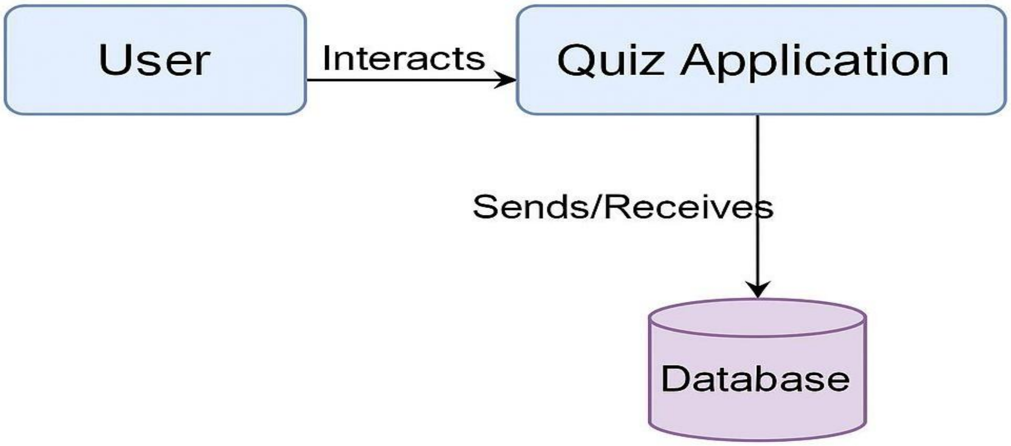
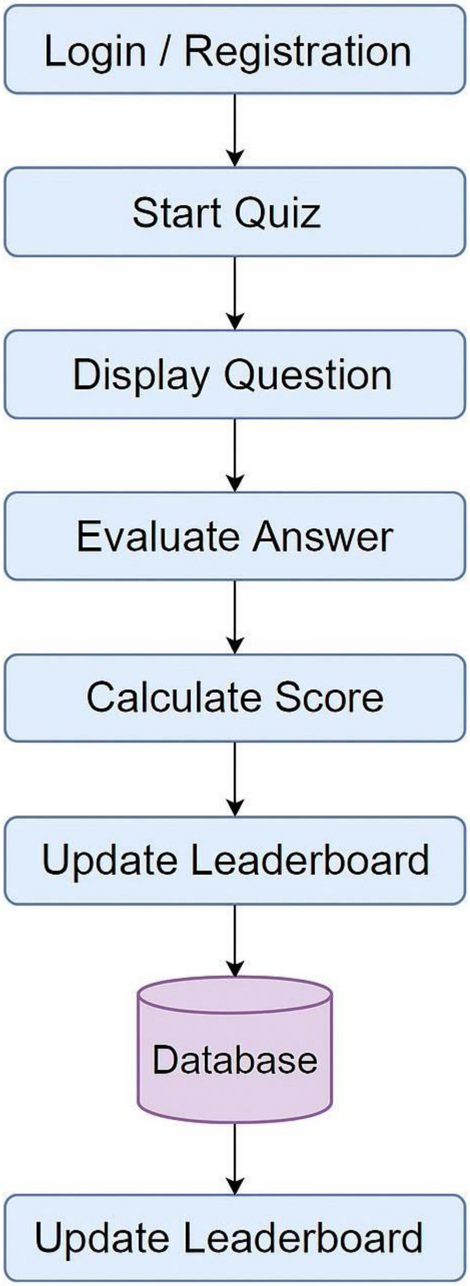


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

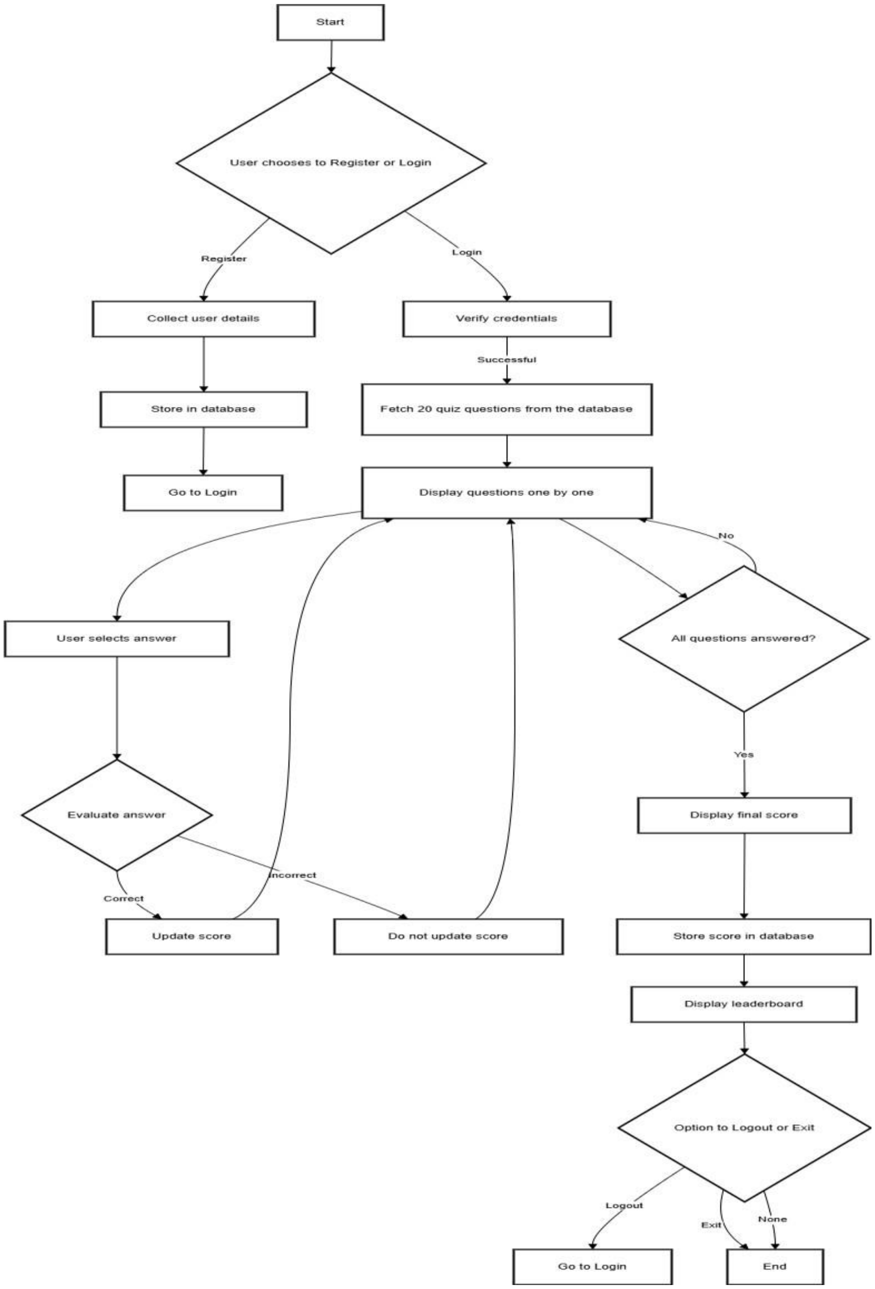
5.2 Low Level Diagram (if applicable)



\

Figure 2: LOW LEVEL DIAGRAM OF THE SYSTEM

5.3 Interfaces (if applicable)



6 Performance Test

The Quiz Game Application was designed to run efficiently on low-resource systems, making it suitable for educational use in real-world settings such as schools, training institutes, and remote environments. Unlike academic-only projects, this application was tested in terms of responsiveness, accuracy, and stability across different modules such as login, quiz execution, and leaderboard updates.

Identified Constraints:
Memory usage: Since the application is GUI-based and runs locally, it was important to ensure low memory consumption.
Speed/Responsiveness: The system had to respond instantly to user actions such as answering questions, logging in, or switching views.
Data integrity: Quiz answers, scores, and user credentials needed to be securely and accurately stored in the database.
Scalability: The design needed to accommodate multiple users and question sets without degradation in performance.

How Constraints Were Handled:
Efficient coding: Modular programming in Python ensured minimal memory footprint and reduced code repetition.
Optimized SQL queries: Use of parameterized queries and indexed tables helped reduce database latency.
Error handling: The app includes input validation and exception handling to prevent crashes and invalid data storage.
Offline capability: Since it works without internet, network dependency was eliminated, improving durability and performance.

Test Outcomes:
The application loaded and executed smoothly on a basic Windows system with 4GB RAM.
Login, registration, and quiz modules responded without lag.
Scores were calculated accurately and stored instantly in the database.
The leaderboard displayed real-time updates as new users completed quizzes.

Recommendations and Future Enhancements:
While the current design performs well, future versions can include:

- Timer-based quizzes to test real-time constraints.
- Exporting results to PDF or Excel for external analysis.
- Adding admin-level controls to review, edit, and delete questions.

6.1 Test Plan/ Test Cases

Test Case ID	Description	Input	Expected Output	Status
TC01	Load quiz data	Run script with quiz dataset	Questions load successfully	Pass
TC02	Correct answer selection	User selects correct option	Score increases by 1	Pass
TC03	Wrong answer selection	User selects wrong option	Score remains the same	Pass
TC04	Score calculation at end	All questions answered	Accurate final score displayed	Pass
TC05	Empty or missing dataset handling	Dataset file not found or empty	Error handled gracefully without crashing	Pass
TC06	Data retrieval from SQL database	Fetch questions using SQL query	Correct data is retrieved in correct format	Pass

6.2 Test Procedure

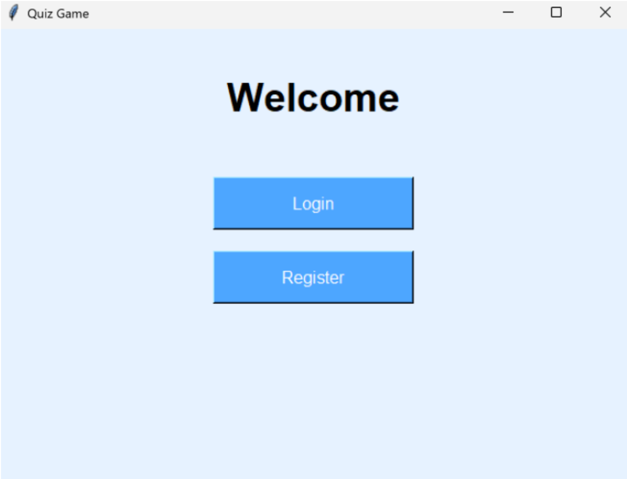
To ensure that the Quiz Game Application functioned correctly across all its components, a structured manual test procedure was followed. The testing focused on validating core functionalities such as registration, login, quiz display, scoring logic, leaderboard updates, and database operations.

Step-by-step test procedure:

1. **Registration Test:**
   Attempted to register with valid and invalid input data to verify input validation and database insertion.
2. **Login Test:**
   Logged in with correct and incorrect credentials to confirm authentication and error message handling.
3. **Quiz Flow Test:**
   Logged in as a user and attempted the quiz. Verified that questions displayed correctly and the app accepted one answer per question.
4. **Score Calculation Test:**
   Completed the quiz and confirmed that the score was calculated accurately based on correct answers.
5. **Database Verification:**
   Checked the SQL Server database to ensure that user records and scores were saved correctly.
6. **Leaderboard Test:**
   Ran the leaderboard module to confirm that it correctly retrieved and displayed user scores in descending order.
7. **UI Interaction Test:**
   Verified all buttons, navigation flows, and screen transitions to ensure smooth GUI behavior and error-free operation.

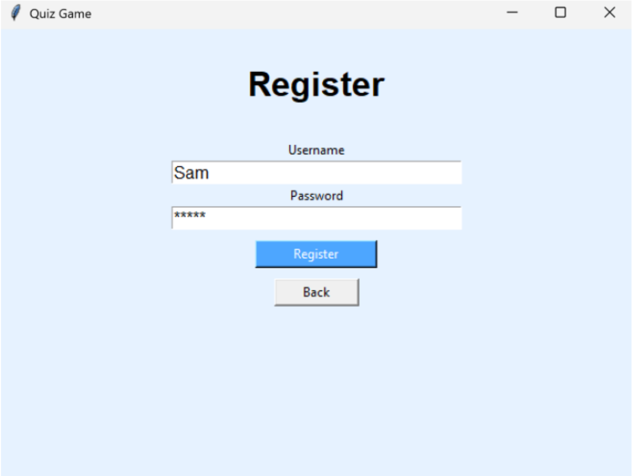
Each of these procedures was repeated multiple times to confirm consistency and reliability. Any detected issues were fixed, and improvements were iteratively applied.

6.3 Performance Outcome



Quiz Game

Welcome

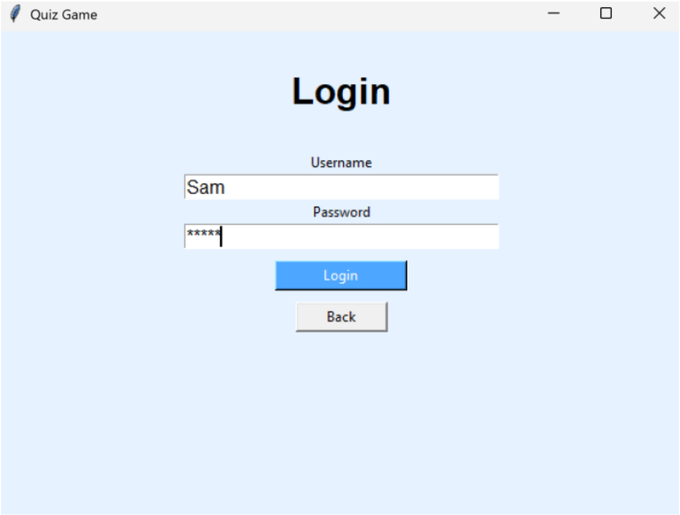


Quiz Game

Register

Username: Sam

Password: *****

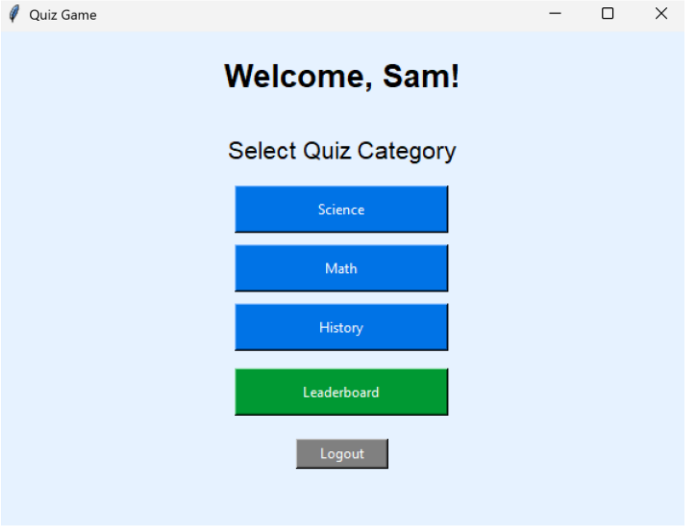


Quiz Game

Login

Username: Sam

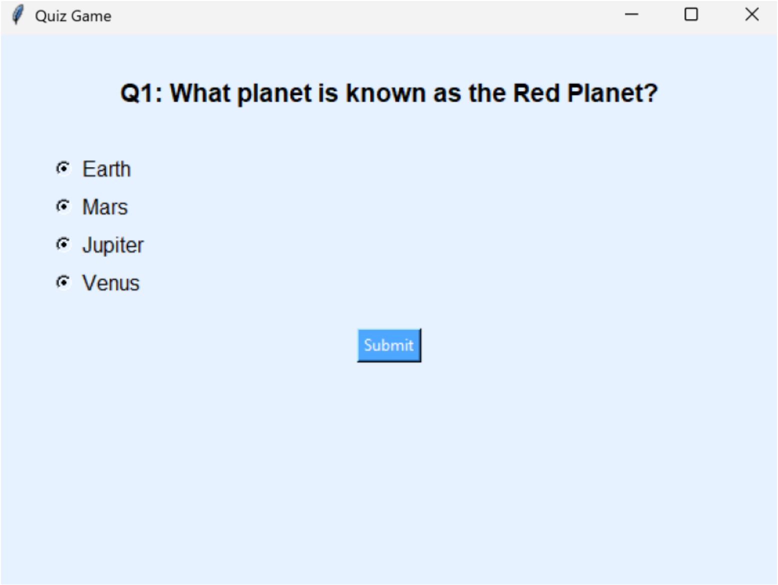
Password: *****



Quiz Game

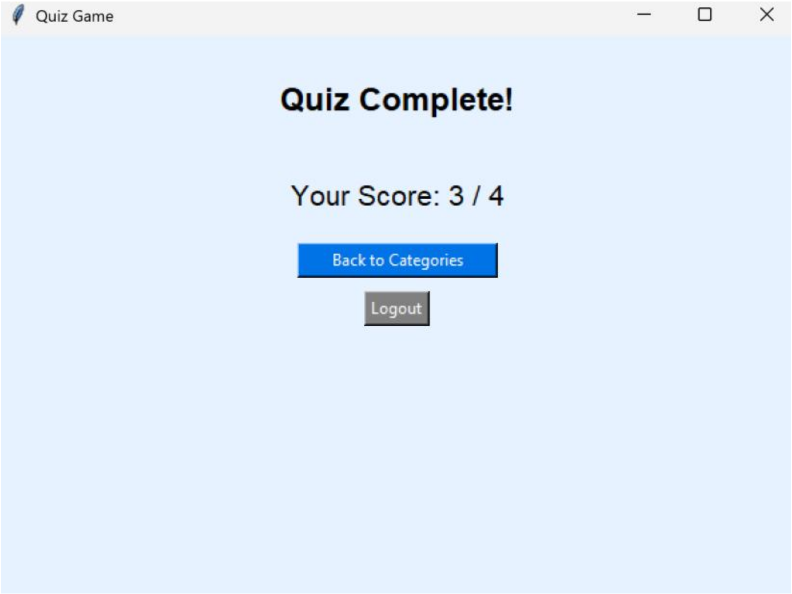
Welcome, Sam!

Select Quiz Category



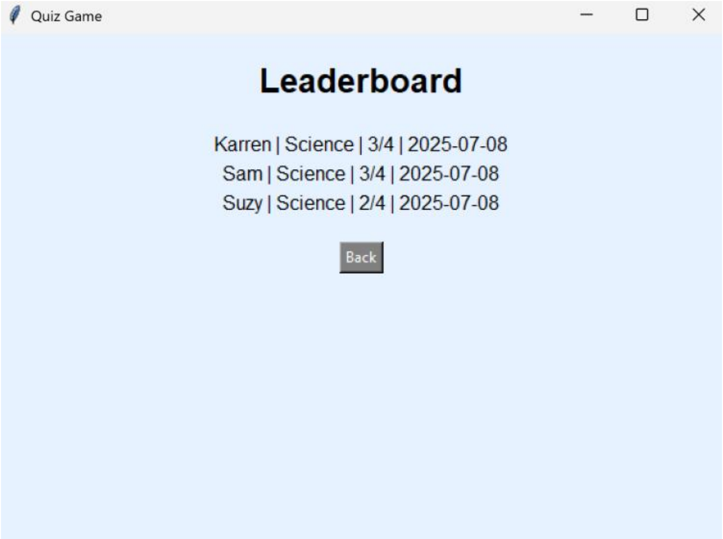
Q1: What planet is known as the Red Planet?

○ Earth
○ Mars
○ Jupiter
○ Venus



Quiz Complete!

Your Score: 3 / 4



Leaderboard

Karren | Science | 3/4 | 2025-07-08
Sam | Science | 3/4 | 2025-07-08
Suzy | Science | 2/4 | 2025-07-08

7 My learnings

During this internship, I gained hands-on experience in full-stack application development using Python for the frontend and SQL Server for the backend. I learned how to design and implement GUI-based applications using Tkinter, interact with a relational database through SQL queries, and structure modular code to handle real-world functionalities like login, registration, scoring logic, and leaderboard display.

Additionally, I improved my understanding of how to break down a project into manageable modules, write clean and maintainable code, and push a project to GitHub for version control and collaboration. This experience also helped me grasp the importance of UI/UX design, data validation, and error handling in application development. These skills have enhanced my technical foundation and will be instrumental in preparing me for future software development roles and industry projects.

8 Future work scope

Due to the limited duration of the internship, some advanced features could not be implemented in the current version of the Quiz Game Application. However, there are several areas where the project can be extended in the future.

In future versions, the application can include a timer-based quiz module to evaluate users under time constraints. Admin functionality can be added to allow the creation, modification, and deletion of quiz questions through an interface rather than directly in the database. User performance tracking over time with analytics charts can also be introduced.

Additionally, features like question category filters (e.g., Science, History, Aptitude), multi-language support, profile management, PDF result generation, and an email system for sending scores could enhance the usability and functionality of the application. Integration with cloud databases and mobile compatibility are also potential enhancements to improve accessibility and scalability.