upSkill CAMPUS

Kristu Jayanti College
AUTONOMOUS Bengaluru
Reaccredited ‘A’ Grade by NAAC | Affiliated to Bengaluru North University

uct

Industrial Internship Report on

”QUIZ GAME PROJECT”

Prepared by

Sneha Philip

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”.

This project is a GUI-based Quiz Game developed in Python using Tkinter for the front end and Microsoft SQL Server for the backend. It allows users to register, select subjects, and take multiple-choice quizzes while storing all data—such as users, questions, and scores—in a secure SQL database. The project emphasizes backend integration with robust data management and user interaction. All code and related files are version-controlled and hosted on GitHub for easy collaboration, access, and submission. The repository includes a .sql file for database setup, ensuring smooth setup on other systems. This project showcases skills in Python programming, database connectivity, and using GitHub for real-world 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.

Over the course of six weeks, I worked on developing a complete Quiz Game Application using Python, SQL Server, and Tkinter for GUI. I started with understanding the project requirements, then moved on to database creation, designing user interfaces, implementing quiz logic, score tracking, and integrating everything into a smooth user flow. I also learned to use GitHub for version control and project submission.

About need of relevant Internship in career development.

Through this internship, I gained hands-on exposure to application development, database management, and project documentation, which are essential for a successful tech career. Internships provide real-time experience in applying theoretical knowledge to practical tasks.

Brief about Your project/problem statement.

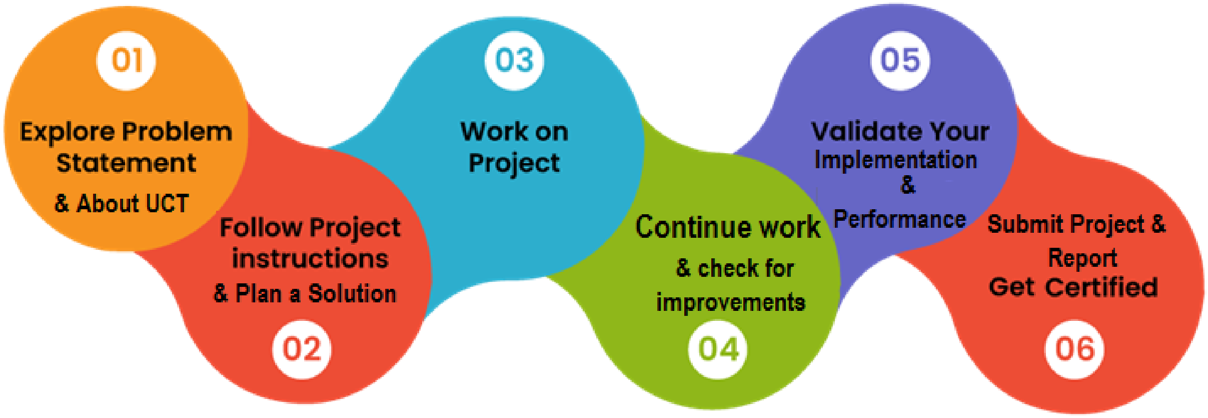
The aim of my project was to build a user-interactive **Quiz Game Application** where users can register, take quizzes on various subjects, and get immediate feedback with scores. The system handles user data, quiz questions, scoring, and result analysis using a Python GUI and SQL Server backend.

Opportunity given by USC/UCT.

I am thankful to Upskill Campus (USC) and Upskill Certified Training (UCT) for giving me the opportunity to work on a meaningful project as part of my internship. Their structured guidance and expectations helped me stay focused and complete the project professionally.

How Program was planned

The program was well-structured into different phases. It started with selecting a project, followed by development of the database, user interface design, backend logic implementation, testing, GitHub documentation, and final report preparation. Each week I had a clear objective and made reports as well , making it easier to track progress and manage tasks effectively.



Your overall experience and learnings.

Throughout this internship, I enhanced my understanding of full-stack application development. I learned how to integrate frontend GUI using **Tkinter**, manage backend databases with **SQL Server**, and ensure smooth functionality across various modules like user registration, login, quiz handling, score tracking, and review systems. Additionally, I gained valuable exposure to **GitHub** for code collaboration and version control, which is a key industry skill. This experience significantly improved my technical confidence, problem-solving approach, and project planning abilities.

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

I would like to express my heartfelt gratitude to **Upskill Campus** for this incredible learning opportunity. Special thanks to my mentors and the support team who were always responsive and helpful during the journey.

Your message to your juniors and peers.

To all my juniors and fellow students:
Make the most out of internship opportunities, even if they feel challenging at the start. Treat your projects seriously, because they help you apply what you've learned and prepare you for the real-world industry. Learn GitHub early, ask questions, stay consistent — and don't give up when things seem tough. Trust the process; it's all worth it in the end.

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

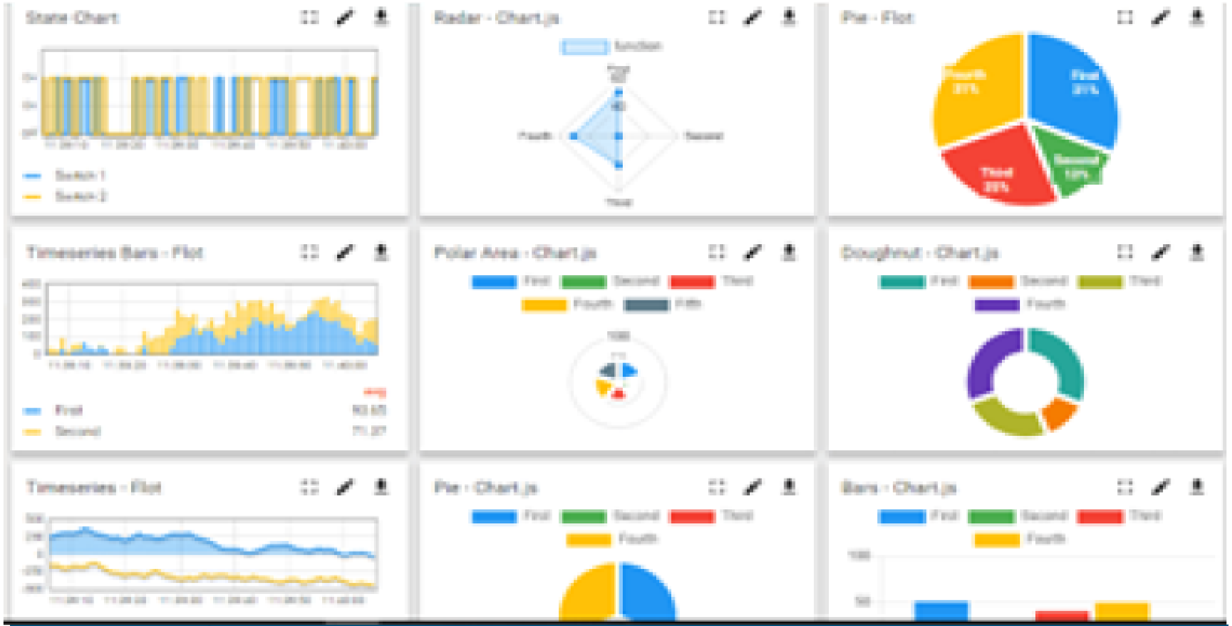
We offer solutions ranging from product design to final production we handle everything for you..

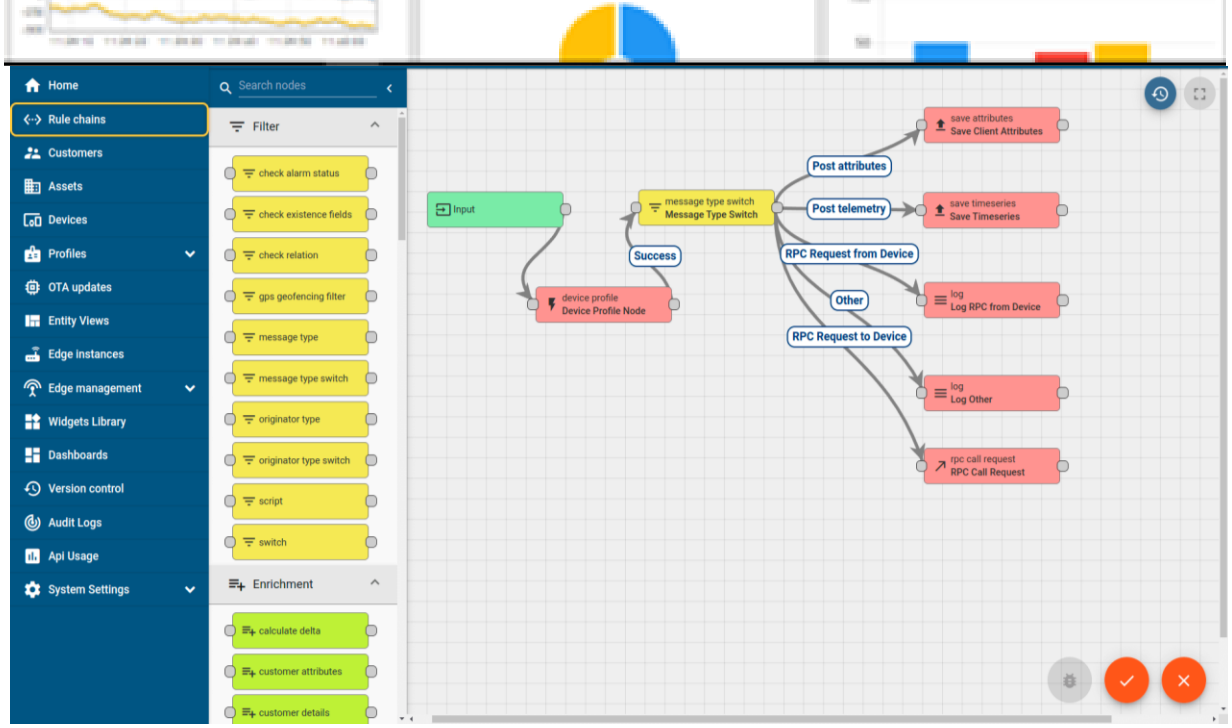
i. UCT IoT Platform (uct Insight)

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

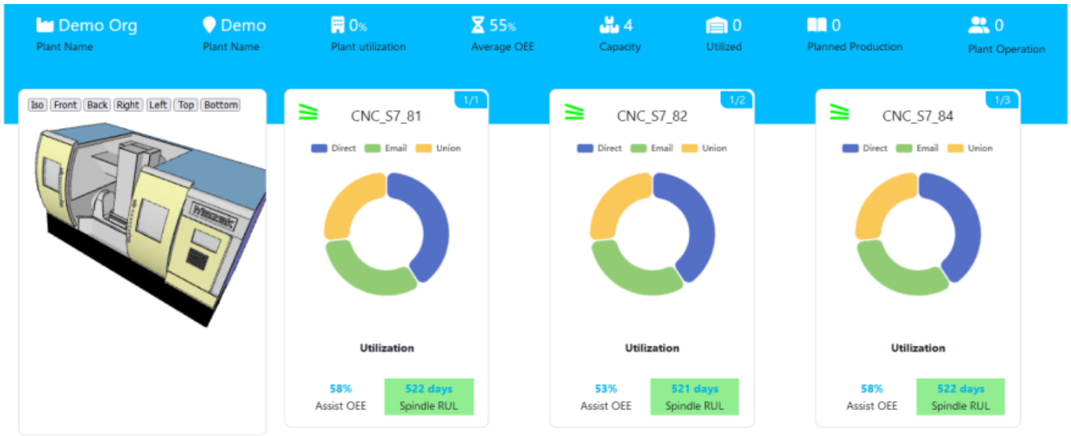
ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

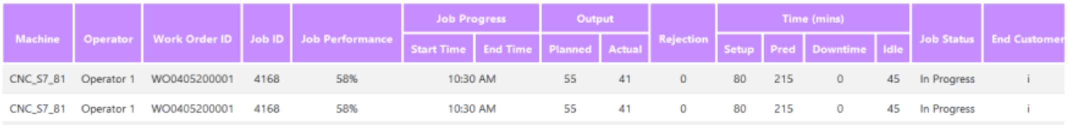
- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular 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.

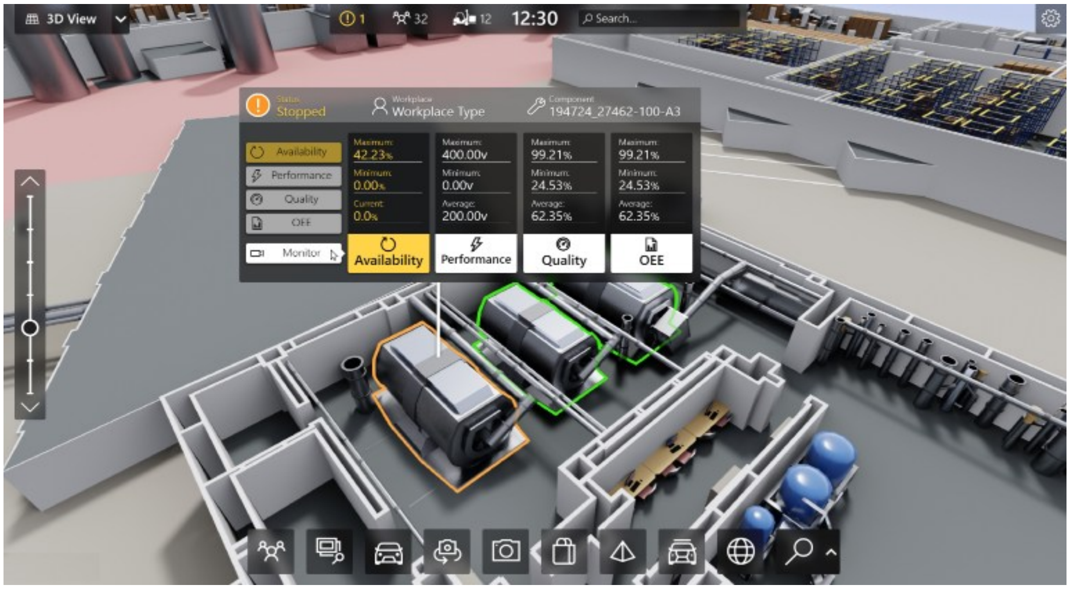


Demo Org	Demo	0%	55%	4	0	0	0
Plant Name	Plant Name	Plant utilization	Average OEE	Capacity	Utilized	Planned Production	Plant Operation

CNC_S7_81 (1/1)	CNC_S7_82 (1/2)	CNC_S7_84 (1/3)
58% Assist OEE	53% Assist OEE	58% Assist OEE
522 days Spindle RUL	521 days Spindle RUL	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

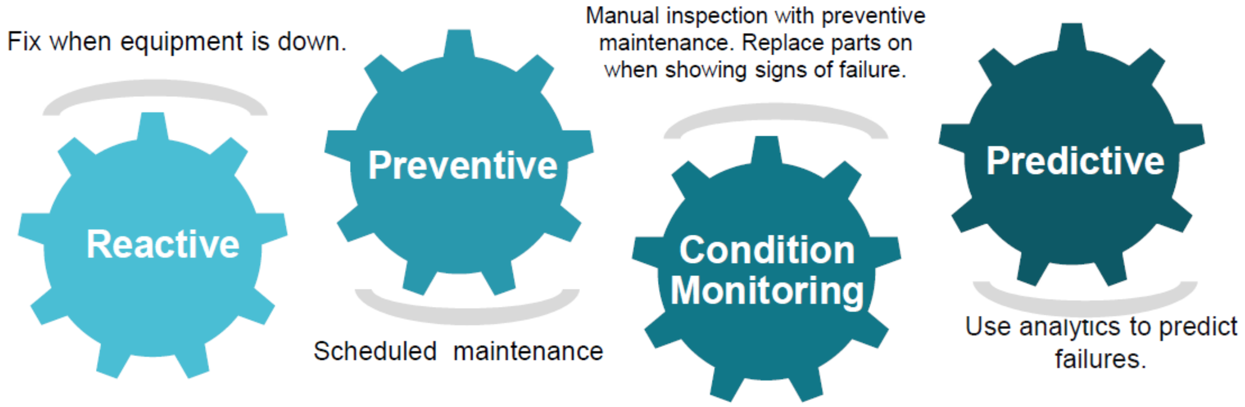
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

iii. LoRaWAN™ based Solution

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

iv. Predictive Maintenance

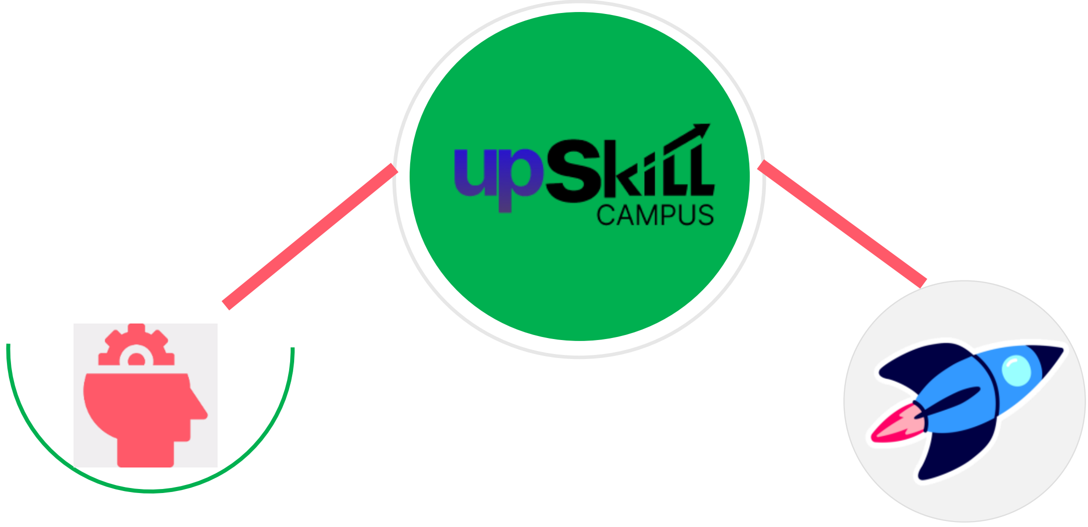
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


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.

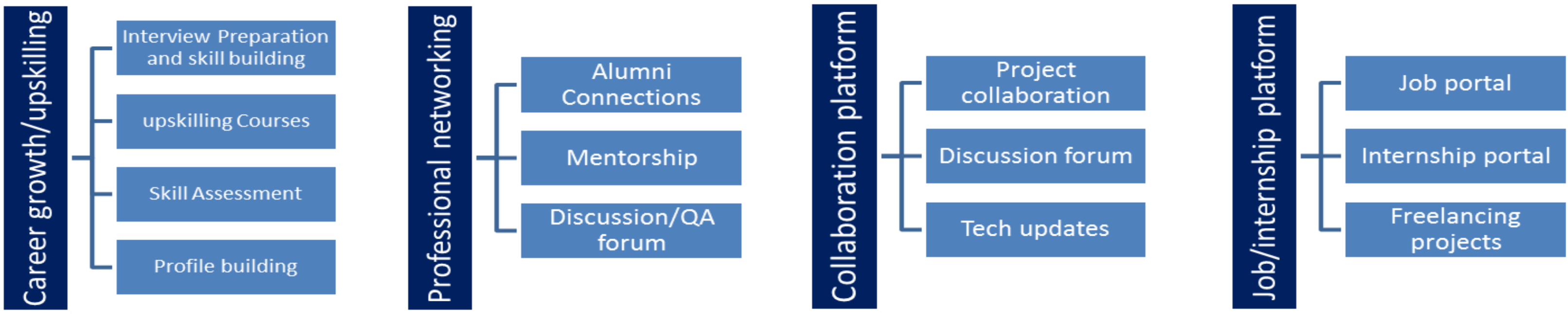
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/

[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

The problem statement assigned was to develop an interactive and user-friendly **Quiz Game Application** using Python. The objective was to create a system where users can register, log in, and take subject-based quizzes in a structured manner. This project aims to simulate a real-world quiz environment where users can answer questions, view their results instantly, and analyze their performance. The application also includes an admin-friendly backend for managing questions and viewing score records.

This project was chosen to understand how quiz-based applications function and to gain hands-on experience in both **frontend (GUI using Tkinter)** and **backend (SQL Server database)** integration. The problem was relevant because it tested concepts like data storage, user authentication, question retrieval, and result evaluation. By building this system, I got the opportunity to apply object-oriented programming, database connectivity, and logical structuring in a real-world scenario.

The final output is a fully functional quiz system where users can register, select a subject, attempt a quiz, view scores, and review their answers. The project also supports score saving, subject-based filtering, and leaderboard features. It was built with scalability in mind, meaning more subjects or questions can be easily added through the backend. This project not only fulfills the internship's technical goals but also enhanced my understanding of building complete applications from scratch.

4 Existing and Proposed solution

There are many online quiz platforms like Kahoot, Quizizz, and Google Forms that allow users to participate in quizzes. While these platforms are feature-rich and widely used, they often require an internet connection, and many lack backend-level customization unless you subscribe to premium plans. Also, most of these tools are third-party hosted, limiting flexibility in database management and quiz control.

The major limitation of existing systems is their lack of control over user data and question banks. These systems are not easily customizable for specific academic or project needs. They also do not allow students or individual developers to experiment with internal logic, scoring mechanisms, or quiz layouts — especially if you want to deeply understand how such applications work behind the scenes.

My proposed solution is a fully self-contained Python-based quiz application with a connected SQL Server backend. It allows total control over user data, quiz content, scoring, and result analysis. The value addition includes local data handling, custom subject selection, instant result display, a leaderboard, and an answer review system. It provides a complete quiz experience with full backend access, suitable for both academic use and personal learning.

4.1 Code submission (GitHub link)

The complete source code of the project including the SQL data scripts and the frontend (GUI using Python and Tkinter) has been uploaded to the following GitHub repository link:

❖ **GitHub Repository Link:** https://github.com/sonil-dot/upskillcampus

4.2 Report submission (GitHub link) :

https://github.com/sonil-dot/upskillcampus/blob/main/QuizGameProject_SnehaPhilip_USC_UCT.pdf

5 Proposed Design/ Model

The proposed design of the Quiz Game Project begins with a simple **homepage interface**, where users can either register as new players or log in using their existing details. This acts as the entry point and validates user identity before starting the quiz. Once logged in, the user is taken through a smooth flow — selecting a subject, answering questions, and viewing results.

After login, the user is allowed to select a **quiz subject** such as General Knowledge, Science, or Computer Basics. Based on the selected subject, the application fetches questions from a connected SQL Server database using backend Python scripts. Each quiz contains a set of multiple-choice questions, and the user selects their answer using a clean and user-friendly interface built using Tkinter.

As the quiz progresses, each answer is evaluated in real-time, and the score is updated internally. At the end of the quiz, the score is displayed along with a message of encouragement. The user can also view correct and wrong answers for learning purposes. This feedback loop adds educational value to the game. Additionally, all scores are stored in a backend database under the user's profile.

The final outcome includes score storage, review options, and the ability to view a leaderboard for comparison with others. The model promotes **learning through practice**, uses **clean GUI-based design**, and is backed by structured **SQL database integration**, ensuring both functionality and smooth user experience.

5.1 High Level Diagram (if applicable)

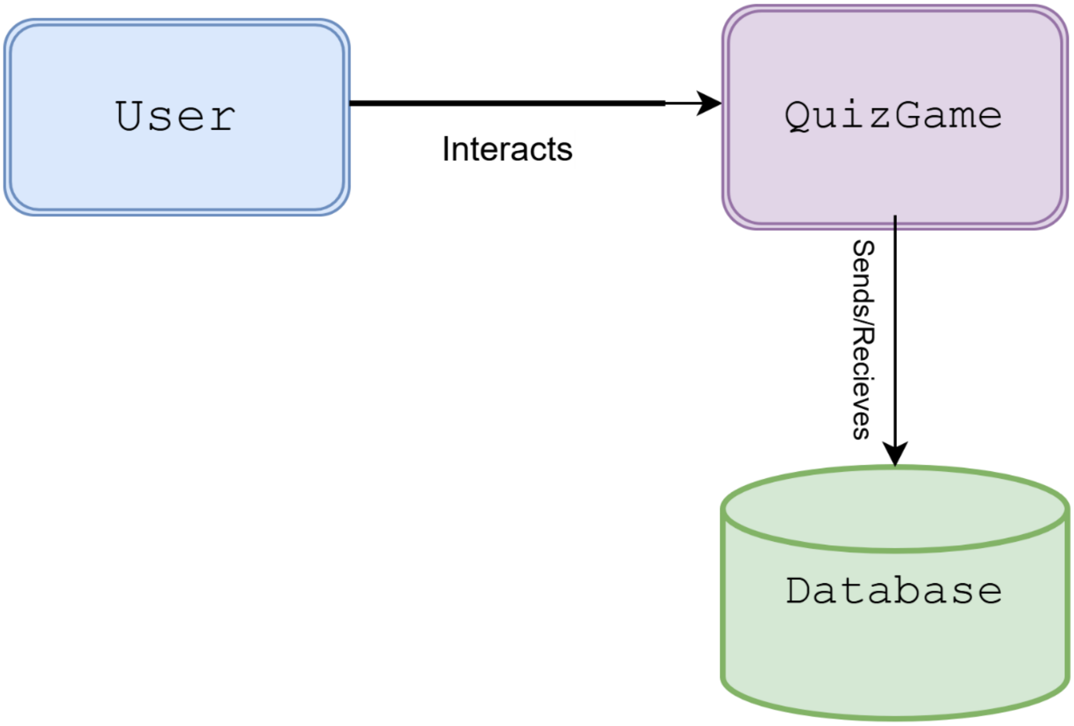


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Low Level Diagram (if applicable)

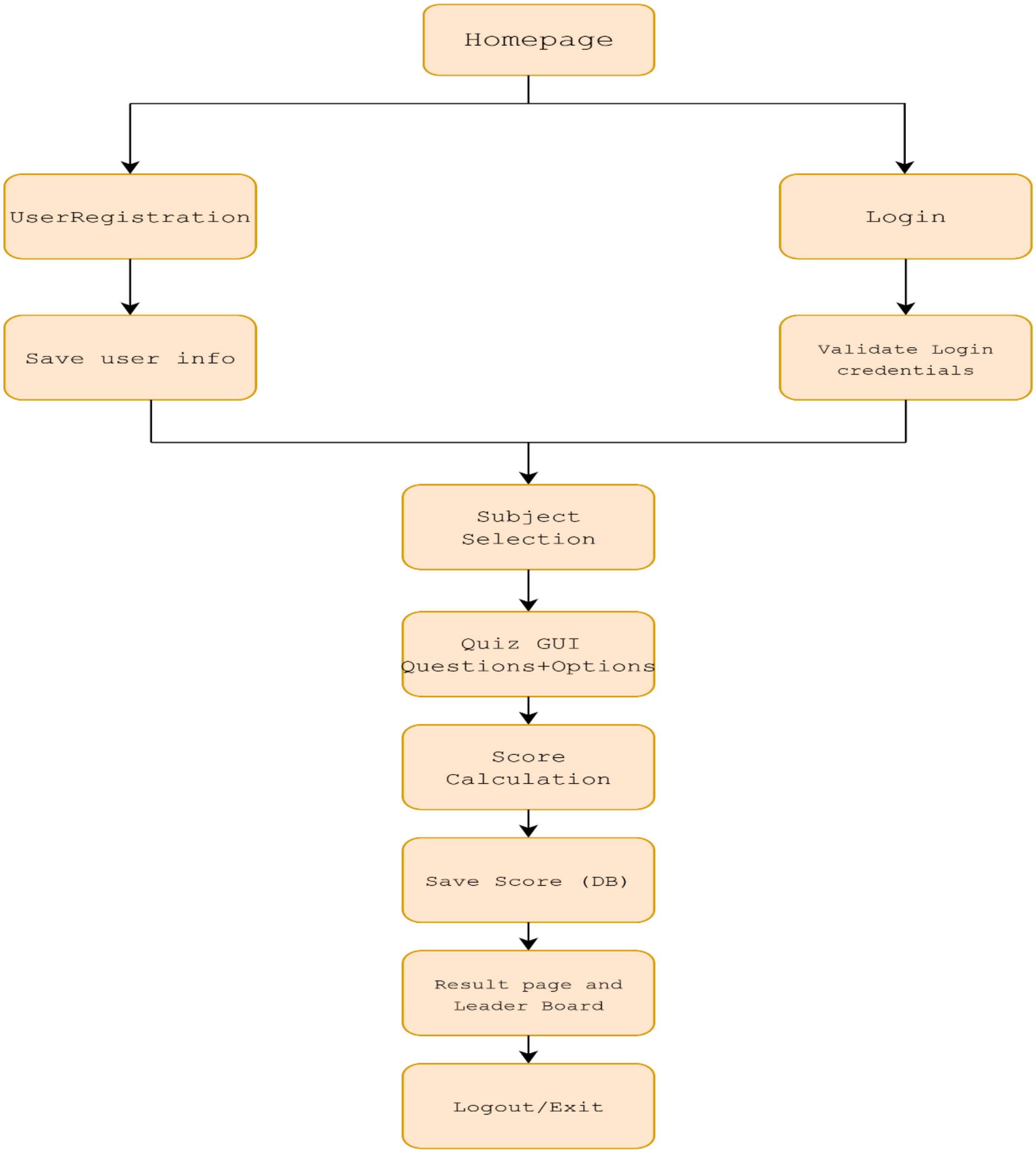


Figure 2: LOW LEVEL DIAGRAM OF THE SYSTEM

5.3 Interfaces (if applicable)

The Quiz Game Project is developed using Python's **Tkinter** library for the Graphical User Interface (GUI). The interface is designed to be simple, user-friendly, and responsive. Below are the key interface screens:

1. Homepage Interface

- First screen of the app.
- Displays buttons: Register, Login.
- User chooses to either register or log in to begin.

2. User Registration Interface

- Form to collect user details: Name, Age, Gender, Email.
- Validates inputs (no empty fields, correct email format).
- On successful registration, user is redirected to login.
- **3. Login Interface**
- User logs in using their registered Name and Email.
- Validates credentials and then allows access to the next screen.

4. Subject Selection Interface

- Displays list of subjects (e.g. History ,Math ,Science).
- User selects a subject to start the quiz.

5. Quiz Interface

- Displays one question at a time with four options (A, B, C, D).
- User selects an answer and clicks **Next**.
- Skipping a question without answering is prevented.
- A total of **20 questions** are shown.

6. Result Interface

- Shows user score at the end of the quiz.
- Displays an emoji and message based on performance.
- Buttons to: **Retry**, **Choose Another Subject**, **View Answers**, **View Leaderboard**, **Logout**, and **Exit**.

7. View Answers Interface

- Shows a summary of all questions.

- Displays user's selected answer and the correct one.
- Highlights correct and incorrect attempts.

8. Leaderboard Interface

- Lists all users who attempted the quiz for that subject.
- Shows scores in descending order (highest to lowest).

6 Performance Test

The Quiz Game Project was primarily designed for desktop systems using Python and SQL Server, which means performance is strongly influenced by memory usage, response time, and database query efficiency. One key constraint identified during development was **speed of data retrieval** — ensuring that questions load quickly without lag, even when accessing from a large question bank. To handle this, the system uses optimized SQL queries with indexed columns and TOP limits, so only necessary data is fetched.

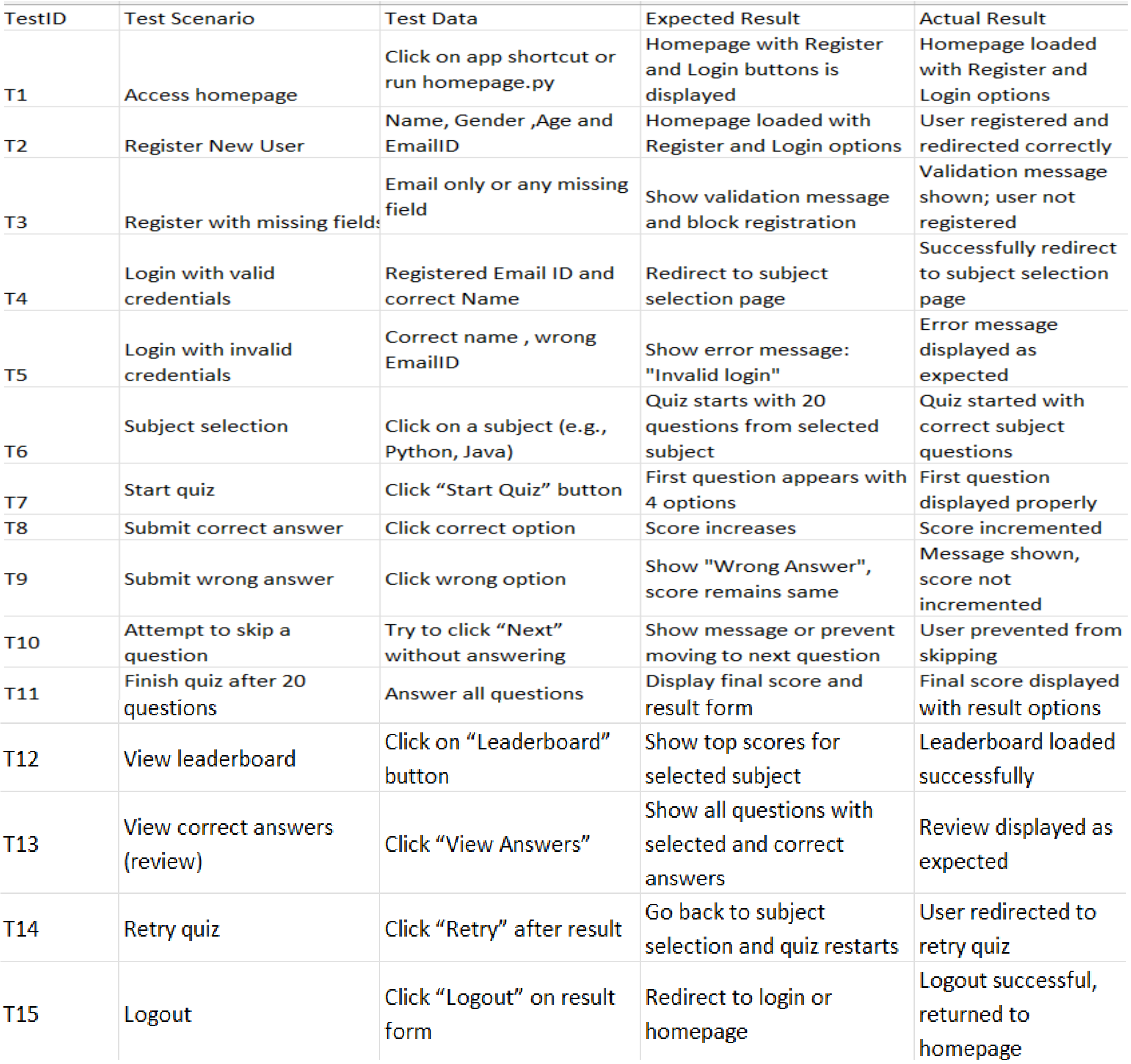
Another major constraint considered was **memory efficiency**. Since the application is GUI-based and relies on multiple Python files, memory management was important to ensure the app doesn't slow down. The use of lightweight libraries like tkinter and structuring the app into smaller functional modules helped in maintaining a smooth user experience. The use of local variables and proper resource closure (e.g., closing database connections after each transaction) ensured no memory leaks occurred.

Although formal benchmarks like MIPS were not conducted due to the nature of the project, **usability tests** showed that the system responded well under normal usage conditions. Loading times were minimal, user input was handled in real time, and all form transitions (homepage → login → quiz → results) were smooth. Error handling was also integrated to prevent crashes in case of invalid inputs or database failures.

In terms of recommendations for industrial deployment, the project would benefit from **stress testing** with hundreds of users and questions. Additionally, converting the app to a web-based system would require attention to scalability and database concurrency. For now, the performance of the project is sufficient for individual users and small group demos, but future enhancements should include performance profiling tools, caching mechanisms, and cloud-based databases to ensure consistent results under higher loads

6.1 Test Plan/ Test Cases

The following test cases were executed to validate the key functionalities of the Quiz Game Project. Each test scenario ensured that the application responded correctly to user actions, verifying both expected and edge case behavior.



TestID	Test Scenario	Test Data	Expected Result	Actual Result
T1	Access homepage	Click on app shortcut or run homepage.py	Homepage with Register and Login buttons is displayed	Homepage loaded with Register and Login options
T2	Register New User	Name, Gender ,Age and EmailID	Homepage loaded with Register and Login options	User registered and redirected correctly
T3	Register with missing fields	Email only or any missing field	Show validation message and block registration	Validation message shown; user not registered
T4	Login with valid credentials	Registered Email ID and correct Name	Redirect to subject selection page	Successfully redirect to subject selection page
T5	Login with invalid credentials	Correct name , wrong EmailID	Show error message: "Invalid login"	Error message displayed as expected
T6	Subject selection	Click on a subject (e.g., Python, Java)	Quiz starts with 20 questions from selected subject	Quiz started with correct subject questions
T7	Start quiz	Click “Start Quiz” button	First question appears with 4 options	First question displayed properly
T8	Submit correct answer	Click correct option	Score increases	Score incremented
T9	Submit wrong answer	Click wrong option	Show "Wrong Answer", score remains same	Message shown, score not incremented
T10	Attempt to skip a question	Try to click “Next” without answering	Show message or prevent moving to next question	User prevented from skipping
T11	Finish quiz after 20 questions	Answer all questions	Display final score and result form	Final score displayed with result options
T12	View leaderboard	Click on “Leaderboard” button	Show top scores for selected subject	Leaderboard loaded successfully
T13	View correct answers (review)	Click “View Answers”	Show all questions with selected and correct answers	Review displayed as expected
T14	Retry quiz	Click “Retry” after result	Go back to subject selection and quiz restarts	User redirected to retry quiz
T15	Logout	Click “Logout” on result form	Redirect to login or homepage	Logout successful, returned to homepage

6.2 Test Procedure

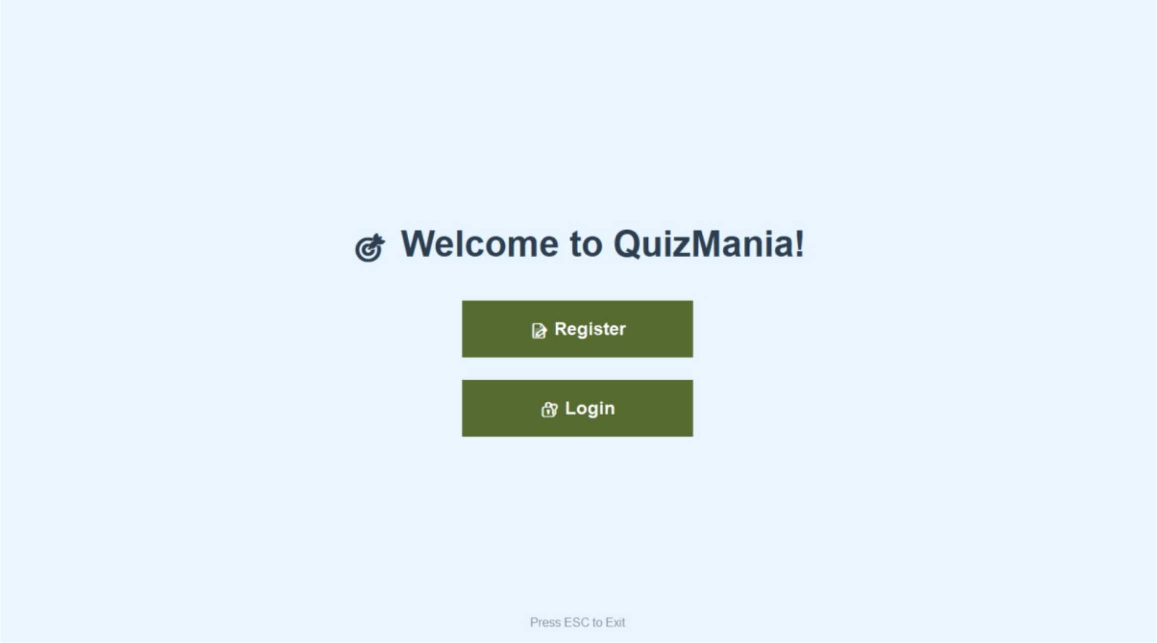
The test procedure involved thoroughly validating each feature and user flow of the Quiz Game Project to ensure all components functioned as intended. Manual testing was conducted for each module in a sequential manner, using various valid and invalid test inputs to simulate real-world usage. Below are the key steps followed in testing:

- **Homepage:** Checked if the user is presented with options to Register or Login and ensured proper navigation from this starting point.
- **User Registration:** Tested with different input combinations to validate field constraints (e.g., email format, age as a number, required fields).
- **User Login:** Verified login using valid and invalid email/password combinations. Ensured redirection to the subject selection page upon successful login.
- **Subject Selection:** Ensured users could see and choose from the available subjects (e.g., Python, Java) and proceed only after selecting one.
- **Quiz GUI:** Verified that 20 questions were loaded based on the selected subject. Ensured navigation between questions worked, and that skipping was restricted.
- **Answer Submission:** Tested correct and incorrect answers and verified that the score was calculated appropriately.
- **Result Form:** Confirmed the score was displayed after quiz completion, along with options like Retry, View Answers, and Logout.
- **Leaderboard:** Checked if the user's score appeared in the leaderboard along with others, and the scores were sorted correctly.
- **View Answers:** Ensured correct answers were shown post-quiz for review and learning purposes.

This detailed test cycle ensured that each part of the project worked in a real-use scenario and provided a smooth, bug-free user experience.

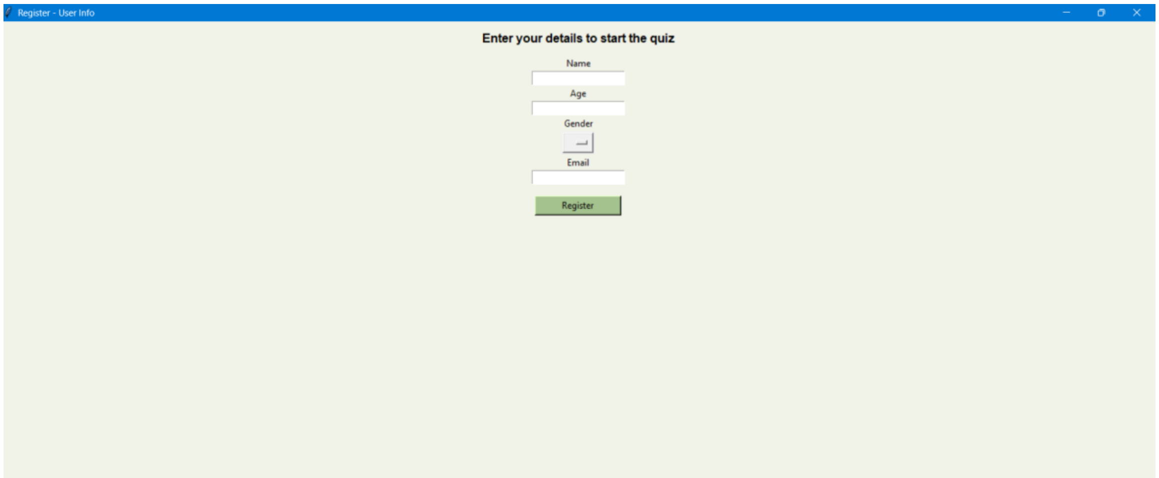
This experience gave me valuable insights into the importance of testing in software development and equipped me with problem-solving and debugging skills that will greatly help in my future career as a developer or tester.

6.3 Performance Outcome

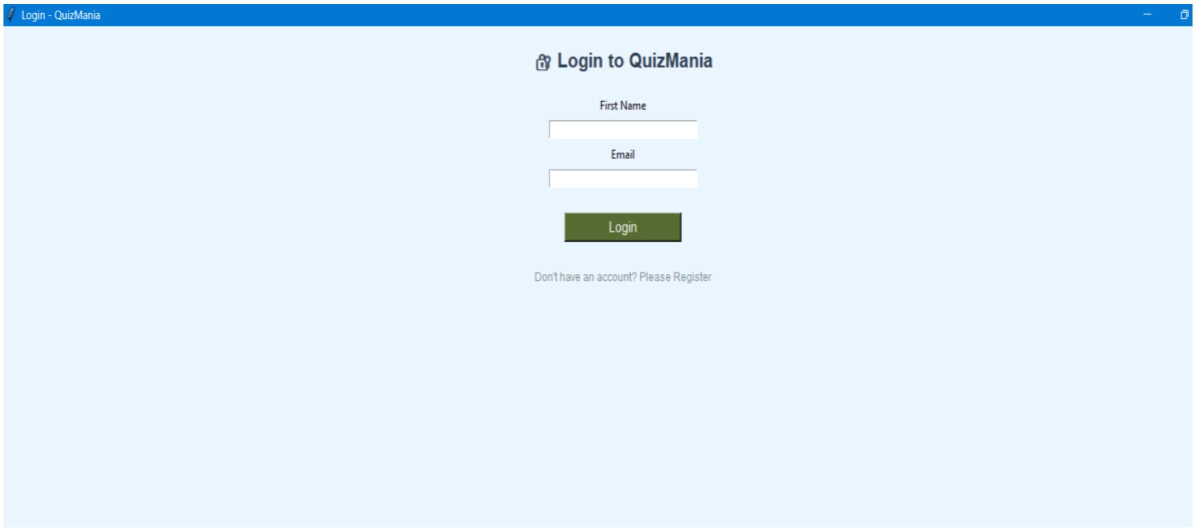


Welcome to QuizMania!

Press ESC to Exit



Enter your details to start the quiz

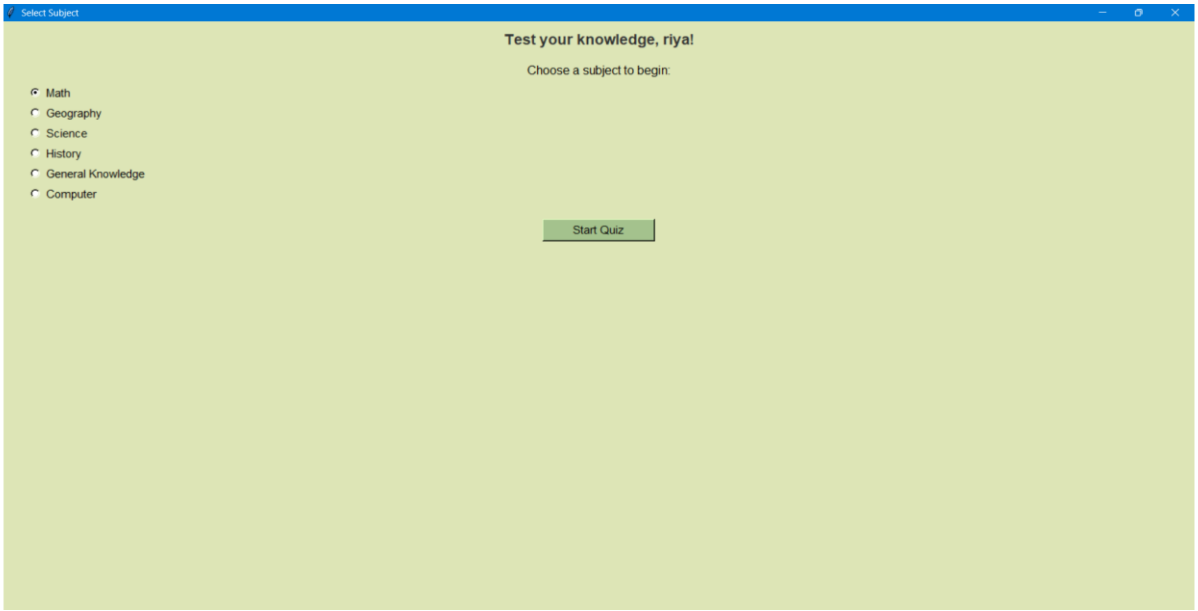

Login - QuizMania

Login to QuizMania

First Name

Email

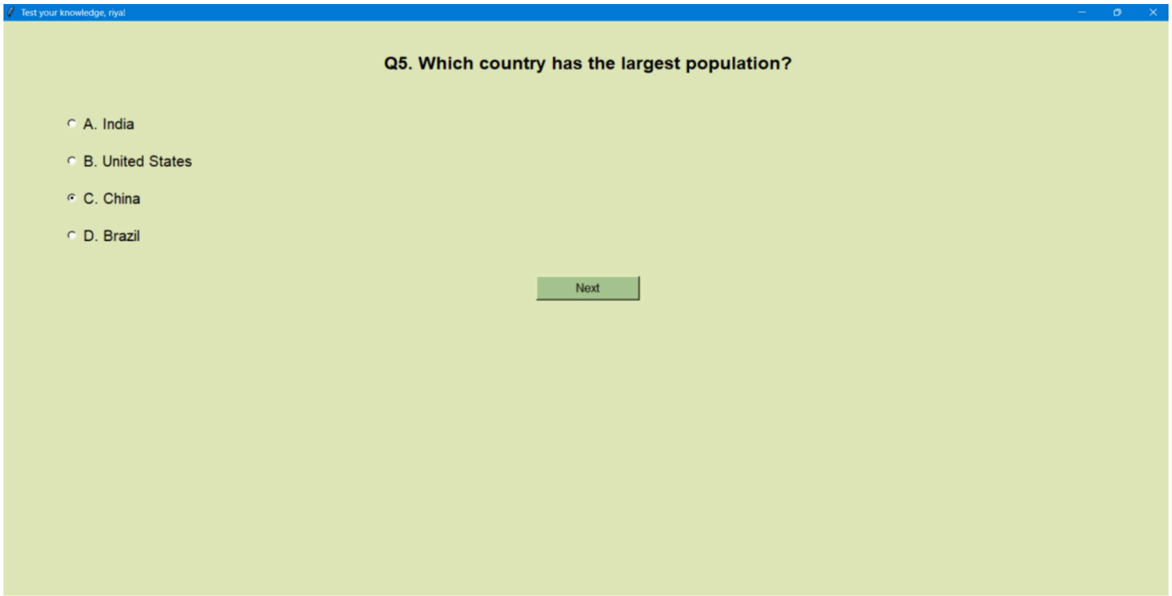
Don't have an account? Please Register


Select Subject

Test your knowledge, riya!

Choose a subject to begin:

● Math
○ Geography
○ Science
○ History
○ General Knowledge
○ Computer



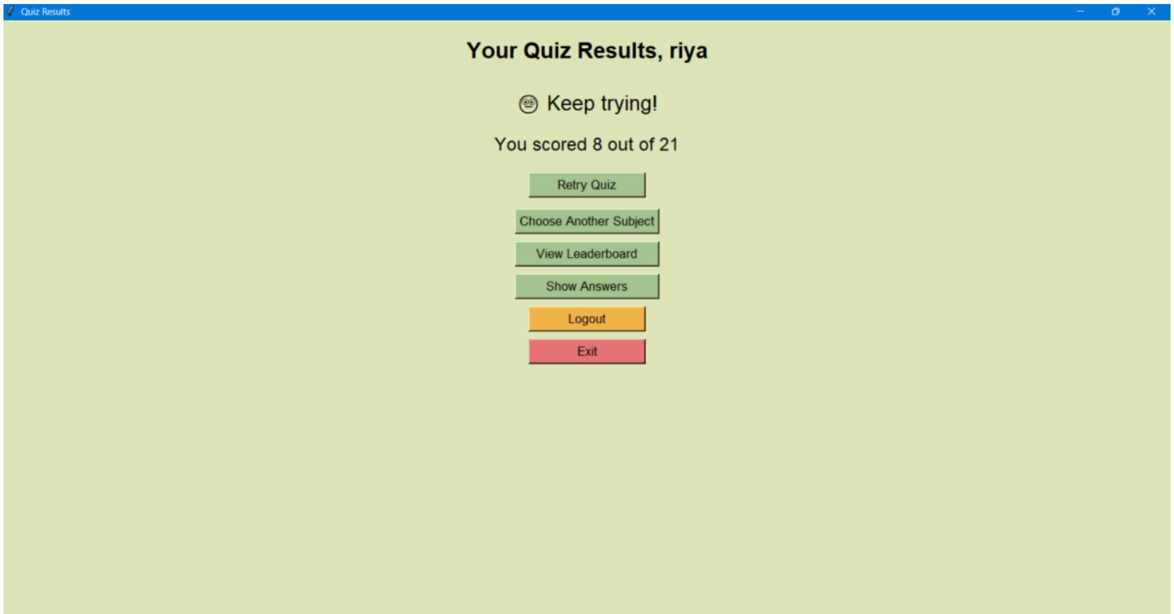
Q5. Which country has the largest population?

○ A. India

○ B. United States

● C. China

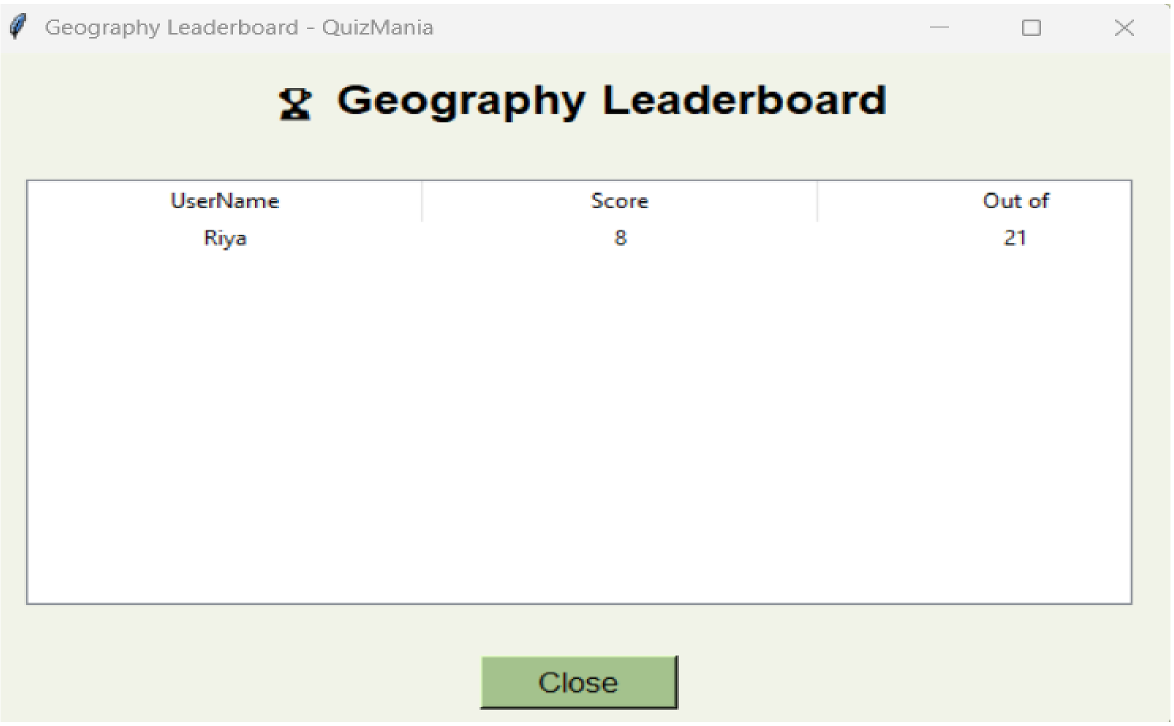
○ D. Brazil



Your Quiz Results, riya

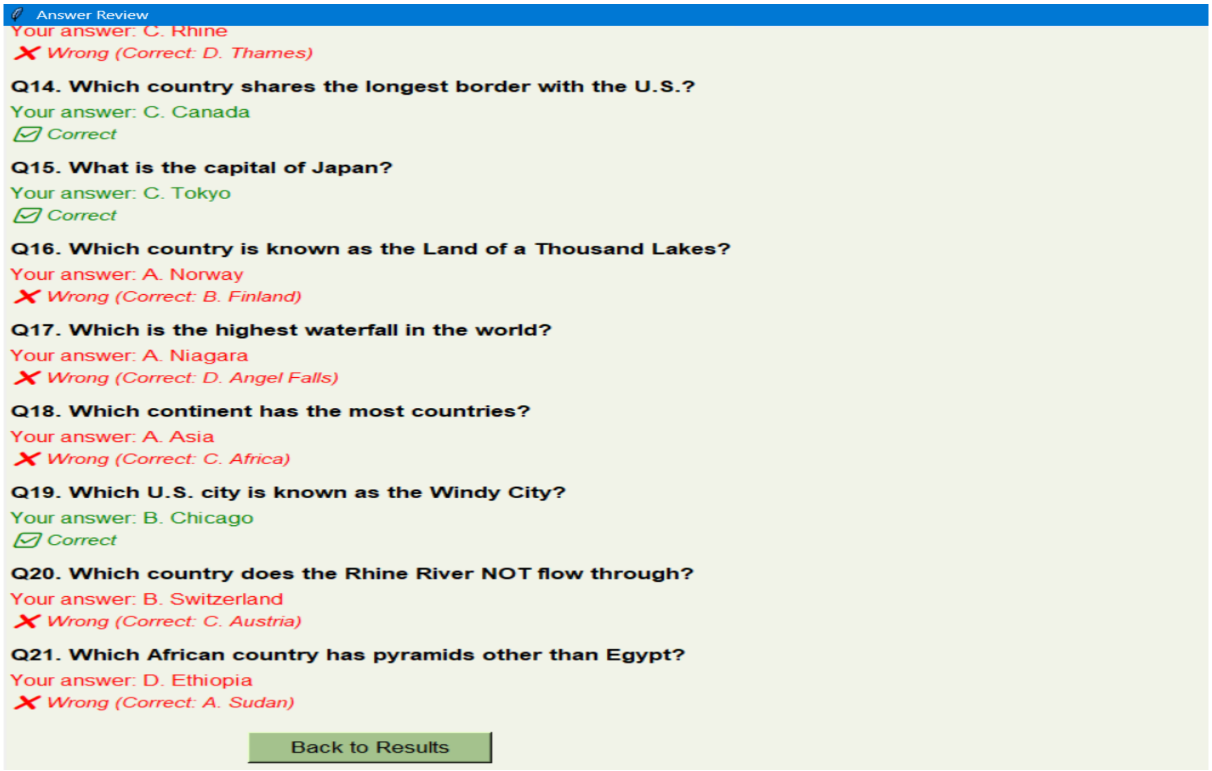
Keep trying!

You scored 8 out of 21



🏆 Geography Leaderboard

UserName	Score	Out of
Riya	8	21



Your answer: C. Rhine
✗ Wrong (Correct: D. Thames)

Q14. Which country shares the longest border with the U.S.?
Your answer: C. Canada
☑ Correct

Q15. What is the capital of Japan?
Your answer: C. Tokyo
☑ Correct

Q16. Which country is known as the Land of a Thousand Lakes?
Your answer: A. Norway
✗ Wrong (Correct: B. Finland)

Q17. Which is the highest waterfall in the world?
Your answer: A. Niagara
✗ Wrong (Correct: D. Angel Falls)

Q18. Which continent has the most countries?
Your answer: A. Asia
✗ Wrong (Correct: C. Africa)

Q19. Which U.S. city is known as the Windy City?
Your answer: B. Chicago
☑ Correct

Q20. Which country does the Rhine River NOT flow through?
Your answer: B. Switzerland
✗ Wrong (Correct: C. Austria)

Q21. Which African country has pyramids other than Egypt?
Your answer: D. Ethiopia
✗ Wrong (Correct: A. Sudan)

7 My Learnings

During this 6-week internship, I gained practical hands-on experience in developing a full-stack Python project using Tkinter for the frontend and SQL Server as the backend. I learned how to design a database schema, create tables, and write queries to interact with real data. I also improved my skills in organizing code into modular files and using version control with Git and GitHub for collaborative and individual project management. Moreover, I understood how important user interface flow, error handling, and user feedback are when creating interactive applications. This internship not only helped me strengthen my technical knowledge but also taught me the discipline of completing a structured project from start to finish, preparing me for future real-world development work.

8 Future Work Scope

The current version of the Quiz Game Project includes essential features such as user login/registration, subject-wise quiz selection, score tracking, and answer review. However, there is ample scope to enhance the functionality. In future versions, we can implement a timer for each question to simulate real-time exam pressure, add difficulty levels (easy, medium, hard), and include multimedia-based questions (images/audio). Another useful upgrade would be to allow users to bookmark questions for review or retry specific quizzes.

From a backend perspective, storing user progress and quiz history can allow the system to suggest personalized question sets based on past performance. We can also explore deploying the application as a web-based platform using Flask or Django so that users can access it from any device. This would require integrating user authentication, responsive UI design, and secure cloud database storage, making the project more robust and scalable.