

Quiz Game Application

(Internship Project Report)

Submitted in partial fulfillment of the requirements
For the Internship Program

Submitted by

ASHI SHRIVASTAVA

Internship Program

Python Programming Internship

Organization / Platform

Upskill Campus (USC – UCT)

Project Duration

November 2025 – January 2026

Technology Stack Used

Python 3, Tkinter, JSON

GitHub Repository Link

<https://github.com/ashishrivastava07/upskillcampus>

Submitted On

15 January, 2026

ABSTRACT

This internship report presents a detailed overview of the work carried out during my Python Development internship at **Uniconverge Technologies**. The primary objective of the internship was to provide hands-on experience in application development and to bridge the gap between theoretical knowledge and practical implementation.

During the internship, I developed a **Quiz Game Application** using Python, integrating core logical functionalities with a Graphical User Interface (GUI) using the Tkinter module. The project includes various interactive features such as category-based questions, randomized question display, a scoring system, and a countdown timer to make the quiz engaging and challenging. The application was developed using modular programming concepts, file handling methods, and event-driven GUI approaches.

In addition to technical learning, this internship helped enhance my problem-solving ability, debugging skills, and documentation practices. Weekly progress tracking, structured development cycles, and continuous improvements enabled successful completion of the project within the given timeline.

Overall, this internship experience significantly strengthened my programming skills and provided a strong foundation for future learning in software development, making me more confident and industry-ready.

LIST OF FIGURES

<i>Figure 1 Flowchart</i>	
---------------------------------	--

LIST OF TABLES

<i>Table 1 Weekly Task Workflow</i>	8
<i>Table 2 Tools & Technologies</i>	11
<i>Table 3 Features Implemented</i>	11
<i>Table 4 Challenges Faced</i>	12
<i>Table 5 Weekly Progress</i>	13
<i>Table 6 Sample Test Cases</i>	14

INTERNSHIP REPORT

Table of Content

1.	INTRODUCTION.....	6
2.	ABOUT THE ORGANIZATION — UNICONVERGE TECHNOLOGIES	6
2.1	Mission & Vision	6
2.2	Services & Offerings	6
3.	ABOUT THE INTERNSHIP	7
3.1	Internship Duration & Mode	7
3.2	Internship Workflow	7
3.3	Internship Objectives	7
4.	INTERNSHIP ACTIVITIES	8
4.1	Assigned Responsibilities	8
4.2	Weekly Task Workflow	8
4.3	Projects / Modules Undertaken	8
5.	PROJECT DESCRIPTION — QUIZ GAME APPLICATION.....	9
5.1	Purpose & Scope	9
5.2	Functional Specifications	9
5.3	Non-Functional Specifications.....	9
6.	SYSTEM DESIGN & ARCHITECTURE.....	10
6.1	Logical Flow (Flowchart Description)	10
7.	TOOLS, TECHNOLOGIES & PROGRAMMING CONCEPTS	11
8.	FEATURES IMPLEMENTED IN THE PROJECT.....	11
9.	CHALLENGES FACED & SOLUTIONS IMPLEMENTED.....	12
10.	LEARNED SKILLS — TECHNICAL & SOFT SKILLS	12

10.1	Technical Skills.....	12
10.2	Professional/Soft Skills	13
10.3	Personal Growth & Confidence	13
11.	WEEKLY PROGRESS SUMMARY	13
12.	TESTING & VALIDATION	14
12.1	Testing Methodology	14
12.2	Sample Test Cases.....	14
13.	FUTURE ENHANCEMENTS & RECOMMENDATIONS.....	14
14.	CONCLUSION	15
15.	REFERENCES.....	15

Python Development – Quiz Game Project

1. INTRODUCTION

In the rapidly evolving field of information technology, students benefit immensely from internships that bridge academic learning and real-world exposure. Internships offer the opportunity to work on actual projects, develop programming skills, and understand professional workflows.

During this internship, I worked with Python — a versatile programming language widely used in software development, automation, data science, and web technologies. The goal was to design and develop a functional application while applying theoretical knowledge in a practical context. This report provides a comprehensive record of tasks, techniques, learning experiences, and reflections from this journey at Uniconverge Technologies.

2. ABOUT THE ORGANIZATION — UNICONVERGE TECHNOLOGIES

Uniconverge Technologies is a technology services organization that offers industry-oriented training programs, practical projects, and internship opportunities for students and fresh graduates. The organization focuses on empowering learners by providing real coding assignments, mentorship, and exposure to standard development practices.

2.1 Mission & Vision

Mission: To build competent developers by offering hands-on training and real-life project experiences.

Vision: To bridge the gap between academic learning and industry-level technical requirements, thereby producing industry-ready professionals.

2.2 Services & Offerings

- Virtual and on-site Internship Programs
- Project-based Training Modules in Programming, Web Development, and Software Engineering
- Skill Development Workshops & Certifications
- Guidance and Mentorship from Experienced Professionals

Uniconverge Technologies aims to foster student development by providing a structured environment with project planning, implementation cycles, testing protocols, and final deliverables — replicating real industry workflows.

3. ABOUT THE INTERNSHIP

3.1 Internship Duration & Mode

- **Duration:** 4 weeks
- **Mode:** Online/Remote — allowing flexibility while ensuring project deadlines and weekly evaluations.
- **Domain:** Python Development

3.2 Internship Workflow

- Orientation and initial guidance on project scope
- Assignment of the main project — the Quiz Game
- Weekly submission of progress reports to mentors
- Regular feedback and code reviews by Uniconverge instructors
- Final project submission along with documentation and report writing

3.3 Internship Objectives

The primary objectives of the internship included:

- Gaining practical experience in Python programming and software design
- Understanding project development lifecycle from requirement gathering to delivery
- Learning GUI development and user interaction design
- Enhancing debugging, file handling, and program flow logic skills
- Building soft skills like documentation, time management, and systematic development

4. INTERNSHIP ACTIVITIES

4.1 Assigned Responsibilities

My roles and responsibilities during the internship included:

- Preparing structured question data sets for the quiz
- Writing Python code to handle quiz logic, user inputs, and score calculation
- Designing GUI interfaces to enhance user experience
- Implementing features like category selection, timed quizzes, and result display
- Debugging, testing, and validating code for different edge cases
- Maintaining weekly documentation and progress reports

4.2 Weekly Task Workflow

Period / Week	Primary Activities
Week 1	Set up project skeleton; implement console-based quiz logic; basic question input/output handling
Week 2	Add features: category selection, improved scoring, question database expansion; input validation
Week 3	Transition from console to GUI — start designing interface using GUI library; basic screen layout
Week 4	Integrate timer, GUI navigation, result screen, answer validation; improve user experience
Week 5	Final testing, bug fixes, refine GUI layout, and documentation preparation

Table 1 Weekly Task Workflow

4.3 Projects / Modules Undertaken

The major project developed was:

Project Title: Quiz Game Application using Python

Modules & Components:

- Data module: Stores and manages question sets (by category, difficulty).
- Core logic module: Handles question selection, user input validation, scoring, and randomization.
- GUI module: Provides graphical user interface (menu, quiz screen, result screen).
- Timer module: Implements countdown timer for each question to make quiz time-bound.
- Input validation & error-handling module: Ensures smooth gameplay without runtime crashes.

5. PROJECT DESCRIPTION — QUIZ GAME APPLICATION

5.1 Purpose & Scope

The Quiz Game is designed as an interactive application for learning and self-assessment. It allows users to:

- Select quiz categories
- Answer multiple-choice questions within a time limit
- Receive instant feedback (correct/wrong)
- View final score and performance summary

The application serves as a practical project to apply Python programming logic, learn GUI development, and understand end-to-end software creation.

5.2 Functional Specifications

- **Multi-category quizzes:** Users can pick from different subjects/topics.
- **Random question selection:** Questions are shuffled each session to avoid repetition or memorization.
- **Time-bound answering:** A timer enforces discipline and timely responses.
- **User feedback:** Immediate indication if an answer is correct or incorrect.
- **Result summary:** Final score displayed at the end of the quiz session.
- **Resilient error handling:** Prevent crashes on invalid inputs or unexpected behavior.

5.3 Non-Functional Specifications

- Usable GUI interface with easy navigation
- Responsive design to user inputs

- File-based data storage to keep question sets flexible
- Modular code for ease of maintenance and scalability

6. SYSTEM DESIGN & ARCHITECTURE

6.1 Logical Flow (Flowchart Description)

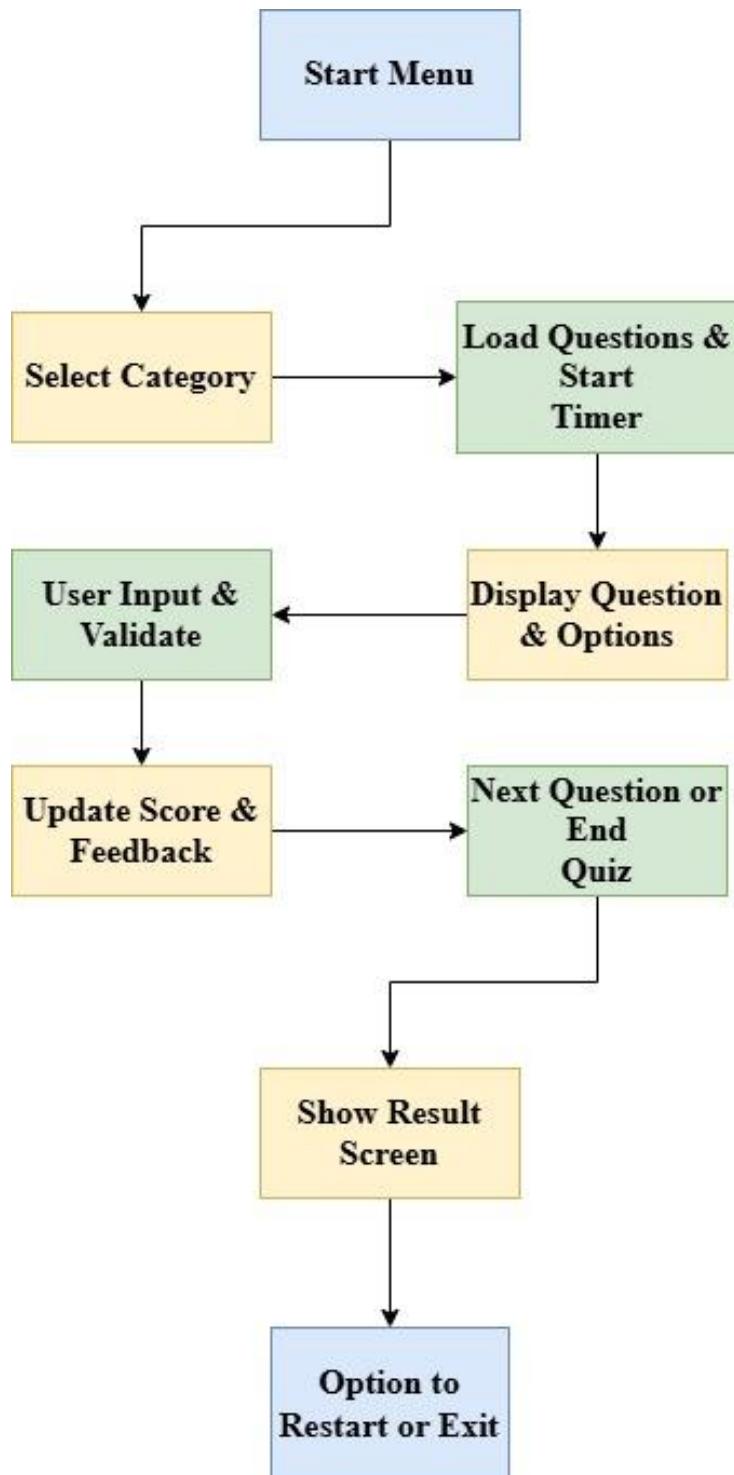


Figure 1 Flowchart

7. TOOLS, TECHNOLOGIES & PROGRAMMING CONCEPTS

Category	Tools / Technologies / Concepts
Programming Language	Python 3.x
GUI Framework	Tkinter
Libraries / Modules	random, time, tkinter, os (for file handling)
Data Storage	Text or JSON files for questions & answers
IDE / Editor	VS Code / IDLE
Concepts Used	Functions, loops, conditional logic, exception handling, modular programming, event-driven GUI programming

Table 2 Tools & Technologies

These tools enabled rapid prototyping, easy debugging, and flexibility in application design.

8. FEATURES IMPLEMENTED IN THE PROJECT

Feature	Description	Status
Multiple-choice Quiz	Present questions with 4 options each	Completed
Category-wise Quiz Selection	Users choose topics before starting	Completed
GUI Interface	Uses graphical windows rather than console	Completed
Timer Countdown per Question	Limits response time, adds challenge	Completed
Randomized Questions	Different question order each time	Completed
Input Validation & Error Handling	Handles invalid user inputs gracefully	Completed
Score Tracking & Result Display	Calculates and shows final result	Completed
Modular Code Architecture	Separate modules for data, logic, UI — easy to maintain	Completed
Future Enhancements (planned)	High-score storage, sound effects, animations, online database	In Progress

Table 3 Features Implemented

9. CHALLENGES FACED & SOLUTIONS IMPLEMENTED

Challenge	Description	Solution / Remediation
GUI freezing during timer updates	Simultaneous loops & UI refresh caused freezing	Used GUI's built-in scheduling method (after()) to update timer without freezing UI
Incorrect score update logic	Score not updating with correct answers due to logic error	Used debug printing & logical restructuring to correct the issue
Difficulty in question data management	Randomization and category-based selection caused question duplication or errors	Implemented structured data storage (list/dict) and thorough validation before use
Screen layout & alignment issues in GUI	Buttons/text overlapped or misaligned on different screens	Used grid / frame based layout in GUI for consistent display
Handling invalid user inputs (empty selection, timeouts)	Program crashed when input was missing or user took too long	Added exception handling, input validation and timer-based timeouts

Table 4 Challenges Faced

These challenges deepened my understanding of both programming logic and user-interface design, and underlined the importance of testing and error-handling in real applications.

10. LEARNED SKILLS — TECHNICAL & SOFT SKILLS

10.1 Technical Skills

- Strong grasp on Python fundamentals: variables, loops, functions, modules
- GUI design and event-driven programming using Tkinter
- File handling for storing and retrieving data
- Use of built-in libraries for randomness, timing, and data management
- Modular code structure and maintainability

10.2 Professional/Soft Skills

- Time management — adhering to weekly delivery deadlines
- Documentation — writing weekly progress logs and final report
- Planning & design — drafting logic, flowcharts, and system structure before coding
- Problem-solving — debugging, testing, and fixing issues under time constraints
- Self-discipline & independent learning — working remotely without direct supervision

10.3 Personal Growth & Confidence

- Increased confidence in handling real-world programming tasks
- Better analytical thinking and systematic approach to tasks
- Improved patience and attention to detail
- Motivated to learn advanced technologies and tackle more complex projects

11. WEEKLY PROGRESS SUMMARY

Week	Activities / Modules Completed
Week 1	Implemented console-based version of quiz — logic for question display and scoring
Week 2	Expanded question database; added category selection and validated user inputs
Week 3	Started GUI implementation — basic layout and user interaction screens
Week 4	Completed GUI integration, added timer functionality, result screen, improved UI flow
Week 5	Thorough testing, bug fixes, interface polishing, prepared documentation & report

Table 5 Weekly Progress

This weekly structure ensured steady progress and managed workload.

12. TESTING & VALIDATION

12.1 Testing Methodology

- Manual testing for each feature (quiz flow, timer, GUI navigation)
- Boundary testing: checking for invalid inputs, timeout, empty selections
- Randomized question order tests to ensure randomness works

12.2 Sample Test Cases

Test Case	Input / Action	Expected Result	Actual Result
Correct answer selected in time	Choose correct option before timer ends	Score increments, next question displayed	Passed
Wrong answer selected	Select incorrect option	Score remains unchanged, feedback shown	Passed
No option selected within time	Wait for timer to expire	Show timeout message or move to next question	Passed
Finish all questions	Answer all questions	Show result screen with final score	Passed
Re-start game after finishing	Click restart on result screen	Quiz resets with fresh questions & score = 0	Passed

Table 6 Sample Test Cases

All test cases passed successfully, demonstrating code stability and reliability.

13. FUTURE ENHANCEMENTS & RECOMMENDATIONS

- Add **high-score storage** to save and compare past performances
- Include **sound effects & animations** for correct / wrong answers
- Store question data in a **database** or **external file** instead of static text for scalability
- Implement **user login system** and **player profiles**
- Convert GUI to a more advanced interface using frameworks like **PyQt**, **Kivy**, or even a **web-based front-end**.
- Add **leaderboard and multiplayer mode** for competitive features

14.CONCLUSION

This internship experience at Uniconverge Technologies has been immensely valuable. It transformed theoretical knowledge into practical skill through structured project work. I learned to design, implement, test, and document a functional software application end-to-end.

The project strengthened my programming abilities, introduced me to GUI and application design, and gave me confidence to take on more complex software tasks. This journey has clarified my career direction — aiming for a role in software development, application design, or advanced Python-based projects.

I am grateful for this opportunity and believe this foundation will help me succeed in future academic and professional endeavors.

15.REFERENCES

- Official Python Documentation (python.org)
- Tkinter GUI Library Documentation
- Online Python & Tkinter Tutorials (e.g., GeeksforGeeks, W3Schools)
- StackOverflow threads for debugging & logic issues
- Internship materials and guidance provided by Uniconverge Technologies

Code Link : <https://github.com/ashishrivastava07/upskillcampus/blob/main/QuizGame.py>

Report Link : https://github.com/ashishrivastava07/upskillcampus/blob/main/QuizGame_Ashi_USC_UCT.pdf