

Quiz Application Report

1. Introduction

This project involved the development of a desktop-based quiz application using Python and the Tkinter library. The application is designed to assess a student's understanding of Python fundamentals through a timed multiple-choice test. It implements essential features such as user validation, randomized question sequencing, scoring logic, timer-based control, and result persistence. The project complies with the following academic and functional guidelines.

2. Features Implemented

2.1 User Authentication and Validation

- Students must provide: First name, Last name, and a 6-digit numeric student ID.
- Input is validated to ensure names are not empty and student ID is exactly 6 digits.
- Validation errors are shown using message boxes.

2.2 Multiple Choice Questions

- 20 Python-related questions are implemented.
- Each question includes 4 options (A-D), with one correct answer.
- Questions are stored in a dictionary format with keys: q, opts, ans.

2.3 Randomization

- Question and answer order is randomized at the start of each session using `random.shuffle`.

2.4 Timer Feature

- Strict time limit of 5 minutes (300 seconds).
- Countdown timer displayed; quiz auto-submits upon timeout.

2.5 Navigation

- One question is displayed at a time.
- Users can move forward and backward between questions.

2.6 Scoring System

- Each question is worth 1 point.
- Score is shown at end with percentage and pass/fail result ($\geq 50\%$ to pass).

2.7 Result Reporting

- Popup displays result summary after quiz.
- Result saved in a file named: results-<studentID>.txt.

- File includes full name, ID, and score percentage.
- 5+ sample entries ensure non-sparse data.

2.8 Retake Option

- Prompt appears asking if the user wants to retake or exit.

2.9 Error Handling

- File I/O exceptions handled gracefully with error messages.

3. Technologies Used

Component	Details
Language	Python 3
GUI Library	Tkinter
OS Compatibility	Cross-platform (Linux, Windows, macOS)
Text File I/O	For storing quiz results

4. Sample Result File

John Doe, ID: 123456, Score: 85.00%
Sample User1, ID: 123450, Score: 72%
Sample User2, ID: 123451, Score: 64%
Sample User3, ID: 123452, Score: 58%
Sample User4, ID: 123453, Score: 90%
Sample User5, ID: 123454, Score: 77%

5. User Interface Design

- Background color: #f0f8ff (soft blue)
- Font: Arial, 14–18 pt
- Buttons styled for clarity and usability
- Real-time timer shown prominently

This project successfully meets all outlined requirements for a Python quiz application. It demonstrates core concepts of GUI design, event handling, file I/O, and input validation. The modular approach ensures ease of maintenance and future enhancements such as graphs, CSV/database storage, or admin interface.