

ESL GenAI Quiz Platform

This project aims to develop a web-based platform designed for English as a Second Language (ESL) students and instructors. Leveraging **Flask** for the web interface, **LangGraph** for advanced GenAI agents, and **SQLite** for data persistence, the platform will facilitate AI-driven quiz creation, assignment, and tracking of student progress. The primary goal is to deliver a functional prototype within a four-week timeframe, enabling instructors to generate comprehension and description quizzes from text or images, and allowing students to take these quizzes and review their performance.

Main Goals Per Week:

Week 1: Foundation & Core Backend

Goal: Set up the core Flask application, establish the SQLite database schema, and implement fundamental user authentication (registration, login) for both student and instructor profiles.

This week builds the structural backbone of the platform.

Week 2: GenAI Integration & Quiz Generation Backend

Goal: Integrate LangGraph to build the GenAI agents capable of generating multiple-choice quizzes from uploaded text (for comprehension) and images (for description). Develop the Flask backend logic for instructors to utilize these agents and persist the generated quizzes in the database.

Week 3: Frontend & Quiz Taking Logic

Goal: Develop the student-facing interface for viewing and taking assigned quizzes. Implement the core logic for quiz submission, automatic grading, and updating student performance records. Additionally, enable instructors to assign quizzes to students.

Week 4: History, Refinement & Deployment Prep

Goal: Implement historical data views for both students (past attempts, scores) and instructors (student performance tracking). Refine the user interface and experience, add basic error handling, and complete project documentation for demonstration readiness, culminating in a functional platform.

Deliverables

I. Core Application & Infrastructure

1. Complete Flask Project Structure
2. Initialized SQLite Database
3. Python Virtual Environment & Dependencies

II. Backend Functionality

1. Functional User Authentication System
2. LangGraph-based GenAI Agents
3. Backend Logic for Quiz Generation
4. Automated Quiz Grading System

III. Frontend Functionality

1. Responsive Web User Interfaces (HTML/CSS/JS):
 - a. Authentication Pages
 - b. Instructor Dashboard
 - c. Student Dashboard
 - d. Quiz Creation Forms
 - e. Quiz Taking Interface
 - f. History Views
 - g. Instructor Quiz Assignment Feature

IV. Data Management & Views

1. Stored Quizzes and Questions
2. User Profiles
3. Comprehensive Quiz Attempt History

V. Project Documentation & Readiness

1. Comprehensive documentation covering project description, setup instructions, how to run the application, features implemented, and any known limitations.
2. Git Version Control Repository
3. Functional Prototype Application