International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

# LearnFlow A Web Based Quiz and Classroom Application

**Aniket Dilip Tambe**
Department of Computer Science, M.sc.Computer Science Patkar
Varde College, Mumbai 400064
Email address:- anikettambe508@gmail.com

**Abstract**—This paper presents LearnFlow, a web application designed to assist teachers in managing their classroom quizzes from a single location. In many classrooms, teachers produce questions using one tool, transfer them to a different tool for sharing, and keep the grading separate. This approach becomes confusing when there are multiple classes or a large number of students.

LearnFlow combines all these functions. The teacher is able to set up a classroom, issue a class code, prepare a quiz, choose the classroom to which it should be sent, and then check the students' attempts. A student can join a classroom by using the code and will only be able to view the quizzes associated with that classroom.

The application makes use of Python Flask on the server side, HTML with Jinja templates for the pages, Bootstrap, CSS and JavaScript for the user interface, and SQLite for data storage. The primary objective of the project is not to replace a complete learning management system; rather, it is to provide a simple and practical method for small educational institutions to manage quizzes and class-based assessments.

LearnFlow also includes assignment management to help with regular classroom activities.Teachers can create an assignment for a specific classroom and add the necessary details along with a due date.Students who are part of that classroom can upload their completed work through the app.Teachers can then review the submitted files, download them if needed, assign grades, write feedback, and update the status of the assignment.Students can later check if their work has been reviewed and view the grades or feedback given by the teacher.

The system uses different access levels for Administrators, Teachers, and Students.
This helps keep the application organized because each user only sees the features related to their role.Teachers are responsible for managing classrooms, quizzes, and assignments.Students join classes, take quizzes, submit their work, and check their results.Administrators can monitor user data and system reports.Scores for objective-type questions are automatically calculated once students submit their answers, which saves time compared to manual grading.

The project is designed to be user-friendly on both desktop and mobile browsers.
By connecting classrooms, quizzes, assignments, submissions, and results in one place, LearnFlow reduces confusion and helps teachers keep better track of academic records.It also allows students to easily access only the learning activities related to their own classrooms.The project shows how a simple Flask and SQLite-based application can effectively support everyday quiz and assignment management in small educational settings.

**Keywords: LearnFlow, classroom, quiz, Flask, SQLite, student assessment.**

## I. INTRODUCTION

LearnFlow is a web application designed for use in the classroom and for teachers and students; the concept arose from a straightforward problem in the classroom. A teacher might have different groups of students, and it is easy to accidentally give the same quiz to all of them. Students often have difficulty in working out which quizzes belong to their own class. LearnFlow addresses this by making the classroom the focal point of the process.

There are three different user roles. An administrator has the ability to manage users and to view reports. A teacher can set up classrooms, prepare quizzes, assign work, and look at students' activities.

A student may join a classroom by using a code, access the quizzes which belong to that classroom, submit their answers, and see their score. The fact that the system is based on roles

ensures that each workspace remains focused on the tasks that the user needs to carry out.

## II. THE NECESSITY OF THE SYSTEM

The hard aspect when managing a quiz is usually not concerned with drafting a single question. It lies in making sure that the questions, the classrooms, the submissions, and the marks are correctly linked together. When the information is handled manually, a teacher may fail to keep track of which quiz was sent to which group and students might also end up seeing content that wasn't intended for them. By using a single system, this kind of confusion is avoided since the same record is used from the time the quiz is created right through to the final results.

LearnFlow is deliberately designed as a web application and not as a static website; it takes in information from users, verifies who is logged in, stores the data in a database, and then displays different pages according to the user's role and class membership. It is important that students do not receive the same view or permissions as teachers.

## III. OBJECTIVES OF LEARNFLOW

- To provide separate login access for users such as Administrator, Teacher, and Student.
- To enable Teachers to create and manage classrooms from a single central location.
- To Each classroom will be given a unique code to help identify it.
- To enable Students will be able to join the appropriate classroom by using the correct code.
- To enable Teachers to create the option to manually create quizzes, including questions, answer choices, correct answers, and scoring.
- To build questions based on specific topics quiz-generation feature that enables to teachers
- To assign Quizzes to particular classrooms before they are made available.
- To Students will only be able to see quizzes that have been assigned to the classrooms they are enrolled in.
- To All quiz-related data, including questions, answers, classroom details, and other relevant information, will be stored in the SQLite database.
- To Students will be able to take published quizzes online and submit their answers.
- To Result automatically calculate and save scores for multiple-choice type questions.
- To Teachers will also be able to create assignments, review student submissions, provide grades and feedback, and students will be able to check the status of their assignments.

## IV. SYSTEM OVERVIEW

LearnFlow is based on a three-part structure. The first element is the user interface, which includes the pages that teachers, students, and administrators view in the browser. The second element is the Flask application, which takes in requests, checks permissions, validates forms, and determines which data should be displayed. The third element is the SQLite database, in which the users, classes, quizzes, questions, assignments, and results are stored.
The use of Flask means that the project remains manageable since it is a student-level web application.

With Jinja templates the same page structure can be used with different data. Bootstrap is employed for the responsive layout and for the ready-made form components, and CSS and JavaScript are used when extra styling or small page interactions are required.

SQLite is useful for this project because it does not need a separate database server during development.

| Module | Main Responsibility |
|---|---|
| Authentication | Role-based registration, login, and access control. |
| Classroom | Teacher classroom creation and student membership through a classroom code. |

| | |
|---|---|
| Quiz | Manual and using AI topic-based question creation, editing, saving, and publication. |
| Assignment | Links quizzes to selected classrooms and manages availability. |
| Attempt and Result | Records answers, calculates scores, and provides teacher reports. |

**Table I. The main components of the LearnFlow application**

## V. HOW THE APPLICATION WORKS

### A. Creating and joining a classroom

The teacher starts by setting up a classroom; the system then stores the name of the classroom, the details of the teacher, and a classroom code.

The students join by typing in this code into their own workspace. After a student has joined, the membership record serves as the connection between that student and the classroom, and this link is subsequently used by the system when deciding which quizzes should be displayed.

### B. Creating and saving quizzes

A quiz may be created by inserting questions by hand. The project also has an AI-style feature based on a topic which produces a draft collection of questions from the topic provided by the teacher.

It is essential that each new request should begin with a new set of questions and that the previous quiz questions must not be included once again. When the teacher saves the quiz, the title, the questions, the options, the correct answers, and the classrooms that have been selected are all stored together; this enables the quiz to be opened later and see the same data which was originally saved.

The 'copy and edit' option produces a separate copy of the quiz and copies the chosen questions only once. The copied quiz has its own quiz identity, which means that any changes made to the copied quiz do not affect the original quiz. Using this method avoids the question count rising every time a teacher opens or makes edits to the quiz.

### C. The visibility of quizzes in the classroom

Before it is published or assigned, the teacher chooses the classroom which is to receive the quiz. When a student opens My Library or Assignments, LearnFlow looks at the student's classroom memberships and then displays only the quizzes that are associated with those classrooms.

For instance, a student in Class A can see the Class A quiz but not the quiz designed for Class B. This rule is the primary measure for preventing all of the teacher-created quizzes from being shown to every student.

### D. The student's attempt and the result

The student picks a quiz that is available, chooses their answers, and submits the attempt. LearnFlow then compares the answers that have been selected with the correct answers that are stored and works out the score.

The attempt and the score are saved so that the student can view the result and the teacher can look at the student's activity for the class. Later on the system can use this information to produce reports.

## VI. DATA DESIGN

The system is based on the relationship involving a user, a classroom, and a quiz. Every main record has a unique identifier. A classroom is assigned to a teacher.

Students are linked to classrooms by means of membership records. A quiz includes a number of questions, and each question has a number of options.

An assignment links a quiz with the classroom which is to receive it.
Whenever a student submits a quiz, the attempt, the answers, and the score are associated with that student and that quiz.

Users keep the details of their accounts as well as their role as administrator, teacher, or student.

The class contains the name of the class, its code, and the teacher who created it.

The class membership includes all the students who have joined each class.

Quizzes, Questions, and Options keep the entire saved question set.

Quizzes are linked with classes and have the ability to control which students are visible.

The submitted work and the marks are stored by Attempts, Answers, and Results.



[Figure :- Activity Diagram]

Figure :- Activity Diagram showing the flow of teacher quiz creation, classroom selection, student quiz attempt, score calculation and result storage.

## VII. RESULTS AND DISCUSSION

The workflow demonstrates that quizzes can be created, linked to classrooms, and made available to the students who are part of the relevant learning group.

Depending on the users' roles, different navigation options are offered, thus separating the teachers' management tasks from the students' attempts to take the quizzes.

Once a quiz has been saved, the question collection previously selected is retained, and the copy-and-edit function operates on a separate quiz record.

The classroom filter is particularly important since it ensures that no student account has access to all the quizzes that have been created by teachers.

LearnFlow is more than just a static website since it is a web application that takes in user input, authenticates different users, carries out processing on the server, reads from and writes to a persistent SQLite database, and changes the content that each user sees depending on their role and classroom membership.

## VIII. Conclusion and Future Research

LearnFlow tackles a common issue in classroom management: students should not have access to quizzes created by all teachers.The system connects each quiz to a specific classroom and verifies if a student has enrolled in that classroom before displaying the quiz.

This ensures that teachers can share the right quiz with the right students, preventing confusion from irrelevant content.

The application has distinct roles for Administrator, Teacher, and Student users. Teachers can create classrooms, generate class codes, design quizzes, publish quizzes, create assignments, and track student activity.

Students can join a classroom using the provided code, take assigned quizzes, upload assignment files, and view their scores, marks, and feedback.Administrators can oversee user activity and review reports.

LearnFlow also minimizes the need for manual grading and record-keeping.
All quiz questions, answers, student responses, scores, assignment submissions, grades, and feedback are stored in an SQLite database.Scores for objective-type quizzes are automatically calculated upon submission, providing students with quick results and enabling teachers to keep accurate records of classroom performance.

In summary, LearnFlow showcases how Flask, Jinja templates, Bootstrap, JavaScript, and SQLite can be used together to build a functional platform for classroom quizzes and assignments.

This project serves as a valuable resource for small educational institutions and has potential for future improvements, such as adding notification features, report exports, multimedia-based questions, live quizzes, document content extraction, cloud hosting, and advanced question generation support.

**Future Research:**

LearnFlow has potential for future enhancements that can benefit teachers, students, and administrators.The current system offers classroom management, quizzes, assignments, submissions, grading, and feedback.Upcoming updates could make the application more engaging, adaptable, and suitable for larger educational setups.

One possible improvement is integrating the OpenAI API to generate advanced questions.
Teachers could specify a topic, subject, difficulty level, and number of questions, and the system would create a suggested set.The teacher would have the option to review, edit, save, or delete the generated questions before finalizing the quiz.

The system can also be expanded to extract text from PDF and DOC/DOCX files.
A teacher could upload study materials or notes, and the application would process the content to create topic-related questions.This could reduce the time needed for preparing quizzes from long documents.

Future versions could include export features for reports.
Teachers and administrators would be able to download quiz results, assignment records, classroom activities, student scores, and feedback in PDF or Excel format.This would simplify keeping offline records and sharing progress reports when necessary.

LearnFlow can also incorporate live quiz competitions.
A teacher could initiate a quiz at a scheduled time, and students can join to answer questions together.A live leaderboard showing scores and rankings can make classroom quizzes more engaging and interactive.

An email-based password reset feature can be introduced.
If a user forgets their password, they can enter their registered email and receive a reset link or verification code, improving user convenience and reducing the need for manual password assistance.

As the number of users grows, the application can transition from local SQLite storage to a cloud-based database.
This would allow the system to support more teachers, students, classrooms, and learning records, while also making the application accessible from different locations via an internet connection.

Future updates can also support image, audio, and video-based quiz questions.
Teachers can upload diagrams, pictures, recorded audio, or short videos as part of a question.This would be beneficial for subjects requiring visual or multimedia-based learning and assessment.

Additional notification features can also be added.
Students can receive alerts when new quizzes or assignments are published, when assignment deadlines are approaching, or when teachers provide grades and feedback.These improvements can make LearnFlow a more comprehensive, interactive, and modern educational platform.

**ACKNOWLEDGMENT-** I thank Mrs. Ahinsa Gaikwad, Department of Computer Science, Patkar-Varde College, for guidance on the project that this paper condenses, and the college for laboratory access. We also thank the Research Methodology faculty for the chapter structure that separated problem, method, and deferred work.

**REFERENCES**

[1] Pallets Projects, “Flask Documentation,” Online: https://flask.palletsprojects.com/en/stable/

[2] Pallets Projects, “Quickstart — Flask Documentation,” O nline: https://flask.palletsprojects.com/en/stable/quickstart/

[3] Pallets Projects, “Using SQLite 3 with Flask,” Online: https://flask.palletsprojects.com/en/stable/patterns/sqlite3/

[4] Pallets Projects, “Uploading Files — Flask Documentation,” Online: https://flask.palletsprojects.com/en/stable/patterns/fileuploads/

[5] Pallets Projects, "Template Designer Documentation — Jinja," Online: https://jinja.palletsprojects.com/en/stable/templat es/

[6] Bootstrap Team, "Get Started with Bootstrap," Online: https://getbootstrap.com/docs/5.3/getting-started/introduction/

[7] SQLite Consortium, "Appropriate Uses for SQLite," Online: https://www.sqlite.org/whentouse.html

[8] Python Software Foundation, "sqlite3 — DB-API 2.0 Interface for SQLite Databases," Online: https://docs.python.org/3/library/sqlite3.html

[9] Miguel Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd Edition, O'Reilly Media, 2018.

[10] Ian Sommerville, *Software Engineering*, 10th Edition, Pearson, 2016.

[11] Thomas Connolly and Carolyn Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th Edition, Pearson, 2015.

[12] Eric Matthes, *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*, 3rd Edition, No Starch Press, 2023.