

**EVEREST ENGINEERING COLLEGE**

**(AFFILIATED TO POKHARA UNIVERSITY)**

**A Minor Project Proposal**

**On**

**E-Learning Platform with Quizzes**

**Submitted By:**

Abhinav Tamrakar [22120106]

Bam Bahadur Karki [22120113]

Hemant Khadka [22120123]

Parikshit Bhandari [22120130]

**Submitted To:**

**Department of Computer and IT**

**Engineering Everest Engineering College**

**Sanepa-2, Lalitpur**

**June 13, 2025**

## ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who contributed to the successful preparation of this project proposal titled **"E-Learning Platform with Quizzes."**

I am especially thankful to my project supervisor for their valuable suggestions, constant encouragement, and insightful feedback which helped shape the direction of this proposal. Their guidance has been essential in identifying the project objectives, scope, and implementation strategy.

I also extend my appreciation to the faculty of the Computer Science Department for providing a supportive academic environment and the necessary resources to explore this idea.

Finally, I would like to acknowledge the support of my peers and family members who motivated me throughout the proposal writing process.

This proposal represents the foundation of a project I am excited to bring to completion through further research, design, and development.

## ABSTRACT

The rapid advancement of digital technology in education has created opportunities for more interactive and accessible learning environments. This project proposal outlines the development of an **E-Learning Platform with Quizzes** — a web-based platform that enables students to take course-related quizzes online, track their performance, and engage in self-assessment.

The proposed system will be developed using the Django web framework and will provide essential features such as user registration, login, quiz-taking functionality, automatic score calculation, and result history. An administrative interface will allow educators to create and manage courses and quizzes dynamically.

This system aims to enhance traditional learning by providing students with a flexible and user-friendly tool for practicing and testing their knowledge anytime, anywhere. It also helps educators reduce the manual workload involved in conducting and evaluating quizzes.

The implementation of this project will demonstrate the practical application of web development and database management concepts while contributing to the growing field of digital education.

### Table of Contents

[ACKNOWLEDGEMENT i](#_TOC_250024)

[ABSTRACT ii](#_TOC_250023)

[LIST OF FIGURES i](#_TOC_250022)v

[LIST OF TABLES v](#_TOC_250021)

LIST of ABBREVIATIONS vi

[Chapter 1: INTRODUCTION 1](#_TOC_250020)

* 1. [Background 1](#_TOC_250019)
  2. [Problem Statement 1](#_TOC_250018)
  3. [Objectives… 1](#_TOC_250017)
  4. Scope and Application 1

[Chapter 2: LITERATURE REVIEW 2](#_TOC_250016)

[Chapter 3: METHODOLOGY 3](#_TOC_250015)

* 1. System Block Diagram 3
  2. [Description of Working Principal 3](#_TOC_250014)
  3. Flowcharts / SDLC 4
  4. [Assumptions… 5](#_TOC_250013)
  5. [Hardware and Software Required… 5](#_TOC_250012)
  6. [Implementation Plan… 6](#_TOC_250011)

[Chapter-4 EXPECTED OUTPUTS 8](#_TOC_250010)

* 1. Performance Metrics 8

[Chapter-5 PROJECT SCHEDULE 9](#_TOC_250009)

* 1. [Schedule… 9](#_TOC_250008)

[Chapter 6: FEASIBILITY ANALYSIS 10](#_TOC_250006)

6.1 [Technical Feasibility. 10](#_TOC_250005)

* 1. [Operational Feasibility. 10](#_TOC_250004)
  2. [Economic Feasibility 10](#_TOC_250003)

6.4 Time Feasibility 10

[Chapter 7: CONCLUSION 11](#_TOC_250002)

[REFERENCES 12](#_TOC_250001)

BIBLIOGRAPHY………………………………………………………………..13

### List of Figures

Figure 3.1 Block Diagram of E-Learning Platform with Quizzes 3

Figure 3.2 System Development Life Cycle……………………………………………………4

### List of Tables

Table 5.1 Gantt Chart 16

### List of Abbreviations

ER - Entity Relationship

AI - Artificial Intelligence

DBMS - Database Management System

IDE - Integrated Development Environment

|  |  |
| --- | --- |
|  |  |
|  |  |

vi

### Chapter 1: INTRODUCTION

### Background

In the digital age, education systems are rapidly evolving to incorporate technology for enhanced learning experiences. Online assessment tools have become essential for self-evaluation, remote education, and academic flexibility.

This project proposes the development of an E-Learning with Quiz Generator using Django, a high-level Python web framework. The system will enable students to register, take quizzes on various subjects, and track their performance, while administrators (educators) can manage courses and quiz content efficiently.

### Problem Statement

Traditional assessment methods are time-consuming and inflexible. Manual grading and quiz distribution involve significant effort and limit access for remote learners. There is a need for a streamlined, accessible, and efficient quiz system that supports digital learning and self-assessment.

### Objectives

* To design and develop a web-based quiz system for students.
* To provide students with instant feedback and performance history.

### Scope and Applications

* **Scope**: User Module: Registration, login, quiz access, view results.
* **Admin Module:** Add/manage courses, questions, and monitor quiz participation.
* **Quiz Module:** Randomized question selection, scoring logic, and result tracking.
* **Database:** Storage for user data, quiz content, and performance history.

**Application:**

The E-Learning with Quiz Generator has a wide range of applications, including:

Academic Institutions: Schools, colleges, and universities can use it for regular assessments, practice quizzes, and online examinations.

Self-Learning Platforms: Independent learners can use it to test their understanding of various subjects at their own pace.

### Chapter 2: LITERATURE REVIEW

Several studies and systems have explored the use of technology in education, particularly in the area of assessments. Online quiz systems have been widely adopted in Massive Open Online Courses (MOOCs) and Learning Management Systems (LMS) like Moodle and Blackboard.

These platforms demonstrate the effectiveness of automated assessment in improving engagement and learning outcomes. Research indicates that instant feedback and self-paced testing contribute positively to knowledge retention and student motivation.

KRENARE PIREVA NUCI1, RABAIL TAHIR AND ALI SHARIQ IMRAN, ALF INGE WANG (2020) conducted a study which investigates the effect of in-lecture quizzes in online classes and correlating the effect of students in the learning curve over four months. Their findings show that a significant increase in students engagement and interaction levels in lectures with systematic in-lecture quizzes. Further, the results show that the learning curve is steeper when using in-lecture quizzes (with 73%) in contrast to classes where in-lecture quizzes are not used (57.5%).[1]

Zamfiroiu Alin, Radu Boncea, Ionut Petre (2019) conducted a study in the learning environment it starts to use mobile technologies to bring the education closer to the students. Teachers are producing a lot of materials and content for these applications, content that will be stored on mobile memory or it is better to use cloud technologies, and the mobile applications only to access the data from the cloud. The content stats to be very largely and easily we can speak about Big Data. Also, the activity of the students on applications is logged. These logs are saved for every action undertaken by the students and saved in the database. Using these logs in a smart way we can analyze them and predict the student’s evolution. Also based on the analysis of these logs the teacher can improve the teaching mode adapting to the student’s necessity. In this paper we analyze the logs of the student’s activity in a quiz on the MOODLE platform to see the behavior of the students in the moments of evaluation. [2]

Zamzami Zainuddin, Muhammad Shujahat, Hussein Haruna, Samuel Kai Wah Chu (2019) conducted a study which investigated the differences in learners’ performance and perceived engagement between three intervention groups in a science class, using two types of pedagogical intervention: traditional instruction with paper-based quizzes and gamified instruction with gamified e-quizzes as formative assessments. Their study shows ways to apply games or game concepts in the classroom can be a promising and innovative too for educators to engage their students in creative learning skills and attractive competition.[3]

Bello Alhaji Buhari, Abubakar Roko (2017) conducted a that research deals with the design and implementation of an improved e-learning system taking Computer Science Unit, Mathematics Department of Usmanu Danfodiyo University, Sokoto as the case study. It allows upload of learning materials online and gives room for one-on-one interaction with the lecturer by creating an avenue for the students to ask questions and get their answers online. The system is aimed at being user-friendly, reliable and improved with better specifications. The e-learning system is designed using HTML, CSS, PHP, Ajax, and MySQL. To ensure proper interaction between students and lecturers, this proposed system incorporate audio calls to lecturers through Skype and also video conferencing through webinar (Web Based Seminar); a software that enables lecturers deliver lecture live. It gives the system the ability to give, receive, and discuss information in real time.[4]

ELPID (E-learning Platform for Innovative Product Development) proposed e-learning platform for virtual project-based design courses are described in the conference paper “E-learning infrastructure prototype for geographically distributed project-based learning”. The platform has to cover the basic theoretical project background through textual, audio, and video material (lectures and tutorials). Further, it needs to integrate virtual and digital technologies to provide students with the opportunity for collaborative and creative problem-solving.[5]

### Chapter 3: METHODOLOGY

### System Block Diagram

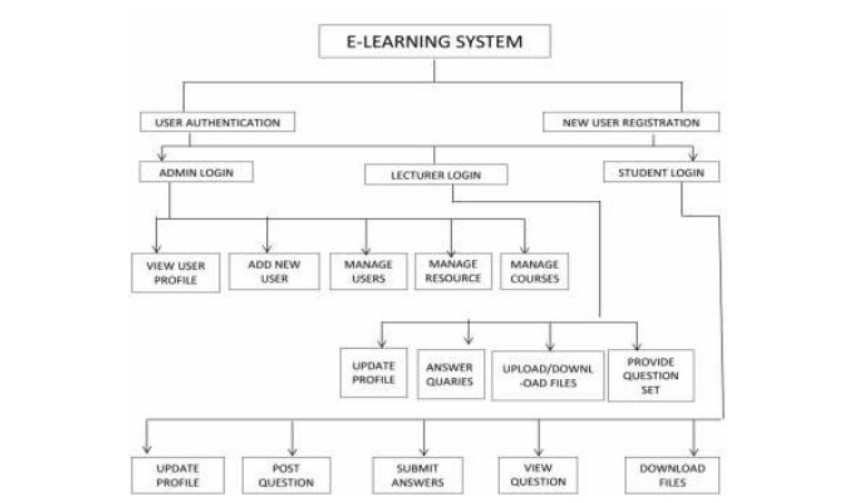


Figure 3-1: Block Diagram of Proposed E-learning System Architecture

### Description of working principle

The diagram illustrates the workflow of an E-Learning System. Here's a step-by-step breakdown of how it operates:

### 3.2.1. Entry Point:

* E-Learning System is the main system that users interact with.

### 3.2.2. User Authentication & Registration:

* Users first go through User Authentication.
* New users can register via New User Registration.
* Existing users can log in via one of three portals:
* Admin Login
* Lecturer Login
* Student Login

### 3.2.3. Admin Workflow:

* After Admin Login, the admin can:
* View User Profile
* Add New User

### 3.2.4. Lecturer Workflow:

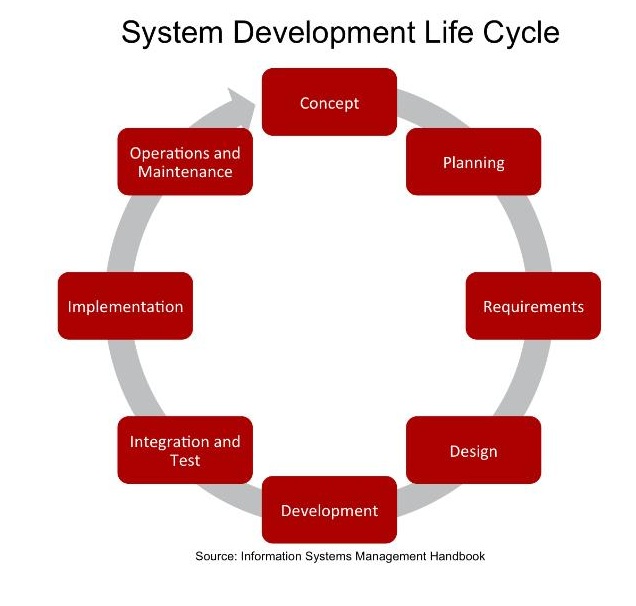
* After Lecturer Login, lecturers have multiple functions:
  + Manage Users
  + Manage Resources
  + Manage Courses
* From here, they can:
  + Update Profile
  + Answer Queries
  + Upload/Download Files

### 3.2.5. Student Workflow:

* After Student Login, students can:
  + Provide Question Set
  + Update Profile
  + Post Questions
  + Submit Answers
  + View Questions
  + Download Files

### 3.2.6. Key Functionalities Across Roles:

* Admins focus on user management and system oversight.
* Lecturers handle academic content, student support, and course administration.
* Students interact with learning materials, ask questions, and submit answers.
  1. **Flowcharts/ SDLC**

****

### 

### Fig 3.2: System Development Life Cycle

### Assumptions

### 3.4.1 System Assumption

### User Roles Are Clearly Defined

### There are three distinct user roles: Admin, Lecturer, and Student.

### Internet Access is Available

### All users are assumed to have stable internet access to use the web-based system.

### Single Device Login

### Users will log in from one device at a time.

### No Biometric or Advanced Authentication

### Basic authentication (username/password) is sufficient.

### Question Types Are Objective

### The system will support only Multiple Choice Questions (MCQs) in the initial version.

### No Plagiarism or Cheating Detection

### The system does not prevent cheating through browser restrictions or camera monitoring.

### Quiz Duration Is Predefined

### Each quiz will have a set time limit, configured by the lecturer/admin.

### No Real-Time Video/Audio Integration

### Lectures or video sessions are outside the current scope.

### 3.4.2 Technical Assumptions

### Using Django for Backend Development

### Django is the primary framework, with SQLite or PostgreSQL for data storage.

### System Hosted on Local Server or Cloud

### It will be accessible via local or cloud-hosted web server.

### Email Services Are Configured

### For password resets or notifications, email service is assumed to be set up.

### System Tested on Modern Browsers

### Chrome, Firefox, and Edge are supported; older browsers are not guaranteed to work properly.

### 3.4.3 User Assumptions

### Users Have Basic Digital Literacy

### Users know how to use web forms, navigate interfaces, and submit answers online.

### Lecturers Will Upload Questions Manually

### No automated question bank import functionality in this version.

### Students Attempt Quizzes Honestly

### The system assumes a trust-based model for academic integrity.

### Hardware and software required

**Hardware Requirements for hosting**

* RAM – 8GB/16GB
* CPU – AMD Ryzen 7 4000 series
* SSD – 1TB

**Software Requirements**

* Operating System - Windows
* Programming Language – Python, CSS, HTML, Javascript
* IDE – Visual Studio, PyCharm
* DBMS - MySQL

**Development and Deployment**

* Frontend: HTML, CSS,JS
* Backend: Python (Django)

**Multimedia**

* Video
* Audio

**Additional Features**

* AI/Powered Recommendation
* Web App

### 

### 3.6 Implementation Plan

Here's a clear and structured **Implementation Plan** for your **E-Learning with Quiz Generator** project:

**3.6.1. Requirement Analysis (Week 1)**

**Activities:**

* Identify and finalize features based on user roles (Admin, Lecturer, Student).
* Define quiz functionality: question types, scoring, time limits.
* Plan database schema and system flow.

**Deliverables:**

* Software Requirements Specification (SRS)
* Use-case diagrams
* Data flow diagrams (DFD)

**3.6.2. System Design (Week 2)**

**Activities:**

* Design ER (Entity-Relationship) diagrams for database structure.
* Create wireframes or mock-ups for key user interfaces.
* Define URL routes and view structure in Django.

**Deliverables:**

* ER diagram
* UI wireframes
* Design documentation

**3.6.3. Development / Implementation (Week 3–4)**

**Backend (Django):**

* Set up project structure and apps (e.g., users, quizzes, courses).
* Implement user authentication and role-based access.
* Create models: User, Quiz, Question, Result, Course.
* Build views and logic for:
  + Quiz generation and submission
  + Admin panel for course/question management
  + Score calculation and feedback display

**Frontend:**

* Use Django Templates with HTML/CSS for UI.
* Add forms for registration, login, quiz taking, and result view.

**Database:**

* Start with SQLite for development.
* Use Django ORM to handle database queries.

**3.6.4. Testing & Debugging (Week 5)**

**Activities:**

* Unit testing for models and views.
* Functional testing for all user flows.
* User Acceptance Testing (UAT) with sample users.
* Debug and fix errors.

**Tools:**

* Django’s built-in test framework (Testcase)
* Browser-based testing and feedback

**3.6.5. Deployment & Documentation (Week 6)**

**Activities:**

* Deploy project on Render/Heroku or local server.
* Finalize user manual and technical documentation.
* Prepare demo and project report.

**Deliverables:**

* Live system URL (if hosted online)
* Deployment guide
* Final project documentation
* Project report & demo presentation

### 

### Chapter 4: Expected Outputs

* The **E-Learning Platform with Quizzes** is expected to deliver a fully functional web application with role-based access, interactive quizzes, and user performance tracking. Below are the specific outputs categorized by user roles and system functionalities:

**4.1. Student Side Outputs:**

* User Registration & Login  
  A student can register with valid credentials and log in to access the platform.
* Quiz Dashboard  
  Students will see available quizzes based on enrolled courses.
* Quiz Participation  
  Students can take quizzes, answer multiple-choice questions, and submit them
* Instant Feedback  
  After submission, the system displays:
  + Score
  + Correct/Incorrect answers
  + Explanation (if provided)
* Performance HistoryStudents can view past quiz attempts, scores, and progress.

**4.2. Admin Side Outputs:**

* Login and Admin Dashboard  
  Admins or lecturers can log in to a dashboard to manage system data.
* Course and Quiz Management  
  Admins can:
  + Add/update/delete courses
  + Create and manage quizzes
  + Add questions (with options and correct answer)
* User Management  
  Admins can view and manage students or lecturers registered in the system.
* Result Analytics  
  Admins can view quiz results and download data for analysis.

**4.3. System-Level Outputs:**

* Role-Based Access Control  
  Secure and isolated functionalities for each type of user (Admin, Lecturer, Student).
* Database Records  
  All user activity, quiz data, and results are stored and retrievable from the database.
* Error-Free Navigation & Validations  
  Forms include validation checks to ensure data integrity and prevent errors.
* Responsive UI

The system should work properly on desktops, laptops, and tablets.

### 

### Chapter 5: PROJECT SCHEDULE

### Schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tasks** | **1 week** | **1 week** | **2 week** | **1 week** | **1 week** |
| **Requirement Analysis** |  |  |  |  |  |
| **Design** |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |
| **Testing** |  |  |  |  |  |
| **Final Report & Demo** |  |  |  |  |  |

**Table 5.1 Gantt Chart**

### Chapter 6: FEASIBILITY ANALYSIS

### Technical Feasibility

The project is technically feasible using widely adopted tools and frameworks:

* **Backend:** Python with Django (high-level, scalable, and well-documented).
* **Frontend:** HTML, CSS, Django Templates (easy integration and customization).
* **Database:** SQLite for development; can scale to PostgreSQL or MySQL.

All technologies required are open-source and free, reducing licensing costs. Django’s built-in admin interface and security features enhance development speed and safety.

**Conclusion:** The system is technically feasible with available open-source tools and student-level development experience.

### Operational Feasibility

* Users (students, lecturers, and admins) will find the system easy to use with a simple web interface.
* The system requires basic digital literacy — registration, taking quizzes, viewing results.
* The operations (quiz creation, score calculation, result storage) are straightforward and automated.

**Conclusion:** The system is operationally feasible and user-friendly.

#### Economic Feasibility

* No significant financial investment is required:
  + Hosting can be done for free using platforms like Render or GitHub Pages (static part).
  + No need for paid software or hardware.
* The cost of development is limited to time and effort, making it suitable for a minor project.

**Conclusion:** Economically feasible, especially for academic purposes.

### Time Feasibility

* The entire system can be implemented within 4–6 weeks:
  + Requirement Analysis & Design: 1–2 weeks
  + Implementation: 2–3 weeks
  + Testing & Deployment: 1 week
* This fits within a typical academic minor project timeline.

**Conclusion:** Time-wise, the project is feasible for successful completion within a semester.

10

### Chapter 7: CONCLUSION

The **E-Learning Platform with Quizzes** aims to enhance the online assessment experience by providing a reliable, scalable, and user-friendly platform for students and educators. Through this project, we seek to automate and streamline the process of quiz creation, delivery, and evaluation, making education more accessible and interactive.

By leveraging Django as the backend framework, the system ensures rapid development, secure data handling, and maintainability. With features like role-based access, instant feedback, and performance tracking, the system will support flexible learning environments and contribute to improved educational outcomes.

This proposal outlines a practical and achievable plan to develop a functional web application within the project timeline, utilizing open-source tools and following best software development practices. The successful implementation of this system will demonstrate the potential of technology-driven learning and provide a foundation for future enhancements in digital education platforms.

## REFERENCES

[1] K. P. Nuci, R. Tahir, A. I. Wang and A. S. Imran, "Game-Based Digital Quiz as a Tool for Improving Students’ Engagement and Learning in Online Lectures," in IEEE Access, vol. 9, pp. 91220-91234, 2021, Doi\*/\*: 10.1109/ACCESS.2021.3088583.

[2] Alin, Zamfiroiu & Petre, Ionut & Boncea, Radu. (2019). Quiz logs analyze on Moodle E-Learning platform.

[3] Zamzami Zainuddin, Muhammad Shujahat, Hussein Haruna, Samuel Kai Wah Chu Year: 2020 Container: Computers and Education Volume: 145 Issue: 1 Page: 103729 DOI: 10.1016/j.compedu.2019.103729

[4] Buhari, Bello & Roko, Abubakar. (2017). An Improved E-Learning System. Saudi Journal of Engineering and Technology. 2. 114-118. 10.21276/sjeat.2017.2.2.5.

[5] L. Grindei and M. Bulgaru, "Design of an e-Learning platform in the field of Innovation Management for engineers," 2011 Proceedings of the 22nd EAEEIE Annual Conference (EAEEIE), Maribor, Slovenia, 2011, pp. 1-6

12

**BIBLIOGRAPHY**

Websites and Online Documentation

1. Python Software Foundation. (n.d.). *Python Documentation*. Retrieved from <https://docs.python.org/3/>
2. SQLite. (n.d.). *SQLite Documentation*. Retrieved from https://www.sqlite.org/docs.html

Software Tools

1. Visual Studio Code – Source code editor by Microsoft.
2. Git – Version control system.
3. GitHub – Platform for code hosting and collaboration.

13