

AI-Powered Real-Life Scenario Online Quiz Portal

¹Akshat Kumar, ²Ansh Bhasin, ³Lakshya Garg, ⁴Vedang Singhal, *Mr. Manish Kumar

akshat.ch567@gmail.com, anshpunjabi001@gmail.com, vedangsinghal001@gmail.com,
[Akshat.hh369@gmail.com](mailto:akshat.hh369@gmail.com), *manishkumar.cse@quantumeducation.in

Department of Computer Science

Quantum University, Roorkee

1.Introduction

The advancement of web technologies and Artificial Intelligence (AI) has significantly transformed the education sector by enabling interactive, personalized, and efficient learning environments. Traditional assessment methods often involve manual evaluation, delayed feedback, and administrative challenges, making them less effective in modern educational settings. To address these limitations, online quiz portals have emerged as powerful digital tools for conducting assessments, tracking learner performance, and enhancing educational engagement.

This project, titled "**AI-Powered Real-Life Scenario Online Quiz Portal**," aims to develop a smart web-based assessment platform using Django/Flask and SQLite. The system enables quiz creation, quiz participation, automated scoring, performance tracking, and intelligent analysis of learner progress. By integrating AI technologies, the portal supports adaptive learning, personalized feedback, and scenario-based question generation, helping learners develop critical thinking and problem-solving skills.

The use of Django or Flask provides a secure, scalable, and flexible framework for application development, while SQLite offers efficient database management for storing user information, quiz records, and performance data. The system also incorporates security features such as user authentication and secure session management to protect user data.

Overall, the proposed AI-powered quiz portal provides an innovative solution for modern educational assessment by improving accessibility, reducing manual workload, delivering instant feedback, and supporting personalized learning experiences. It represents a significant step toward the adoption of intelligent educational technologies in digital learning environments.

2. Literature Review

The literature highlights the growing importance of online quiz portals and AI-powered educational technologies in modern learning environments. Educational institutions increasingly rely on digital assessment platforms to improve learning outcomes, monitor student performance, and provide flexible assessment methods. Traditional quiz systems have evolved significantly with the integration of Artificial Intelligence, Natural Language Processing (NLP), and Machine Learning techniques.

Several studies have emphasized the benefits of automated assessment systems. Das et al. (2021) reported that automated question generation and assessment tools reduce manual effort and help identify learning gaps more effectively. Gupta et al. (2021) demonstrated that combining NLP, OCR, and Django technologies enables automatic generation of meaningful quiz questions from educational content, thereby improving accessibility and efficiency.

Recent research has focused on AI-driven adaptive learning systems. Khare et al. (2025) found that AI-powered quiz platforms can generate dynamic and personalized questions, reducing repetitive assessments and increasing learner engagement. Similarly, Khudiyev (2025) highlighted the role of AI in providing intelligent feedback and identifying knowledge gaps, which enhances learning effectiveness and supports personalized education.

Studies have also shown the significance of web development frameworks such as Django and Flask in educational application development. Dela Rosa and Abad (2025) concluded that Django-based quiz systems offer secure authentication, efficient database management, and scalable assessment solutions. Furthermore, Fu et al. (2025) introduced concept-based quiz generation using Large Language Models (LLMs), demonstrating improved educational outcomes through meaningful and context-aware assessments.

3. Research Methodology

Research methodology refers to the systematic process used to collect information, analyze requirements, design the system, and implement the proposed solution. For this dissertation, the research methodology focuses on the development of an AI-powered Online Quiz Portal using Django/Flask and SQLite. The methodology includes requirement analysis, system design, technology selection, database creation, implementation, testing, and evaluation of the web application. The proposed system aims to provide features such as quiz creation, quiz

participation, AI-generated real-life scenario questions, automated score tracking, and performance analysis.

The methodology adopted in this project combines both qualitative and practical development approaches. The study not only analyzes existing online quiz systems but also develops a functional prototype to evaluate the effectiveness of AI-assisted quiz management and assessment system.

3.1 Research Design

The research follows a Development-Based Research Design because the primary objective of the study is to design and develop a web-based application. This approach is suitable for software and web development projects where practical implementation and system evaluation are essential parts of the research.

The research process consists of the following stages:

- Problem Identification
- Requirement Gathering
- System Analysis
- System Design
- Development of the Web Application
- Database Design
- AI Integration for Quiz Generation
- Testing and Evaluation
- Result Analysis

The methodology focuses on solving issues found in traditional quiz systems, such as manual quiz preparation, lack of adaptive learning, delayed feedback, and poor performance tracking.

3.2 Research Gap

Traditional examination and quiz systems have several limitations, including: Manual preparation of quizzes, Time-consuming evaluation process, Lack of instant feedback, Limited

accessibility, Repetitive question patterns, Absence of personalized learning, Difficulty in maintaining records manually

Many existing quiz portals also lack intelligent features such as AI-generated questions, adaptive learning mechanisms, and real-life scenario-based assessments. Therefore, the proposed research aims to develop a smart online quiz portal that overcomes these limitations.

3.3 Data Collection

My research based on both types of data Primary and Secondary data:

3.3.1 Primary Research

Primary data was collected through:

- Observation of existing online quiz platforms
- Discussions with students and teachers
- Analysis of user requirements for quiz systems

The collected information helped identify important features such as: User registration and login, Quiz creation module, Multiple-choice question management, Timer-based quizzes, AI-generated questions, Real-life scenario-based assessment, Score tracking and analytics, Performance reports

3.3.2 Secondary Sources

Secondary data was collected from:

- Research papers
- Journal articles
- Google Scholar publications
- Online educational platforms
- Technical blogs and documentation

The literature review helped understand recent developments in AI-powered educational systems, Django/Flask web frameworks, and automated assessment technologies.

3.4 Data Analysis Tools

The project follows the Agile Software Development Methodology because it supports flexibility, continuous improvement, and step-by-step system development. Agile methodology is suitable for web applications because it allows modifications during the development process based on testing and feedback.

The Agile development cycle used in this project includes:

1. Planning
2. Requirement Analysis
3. Designing
4. Coding and Development
5. Testing
6. Deployment
7. Maintenance

Agile methodology was selected because it allows incremental development of modules such as authentication, quiz management, AI integration, and score tracking.

6. System Architecture

The proposed system follows a three-layer architecture:

6.1 Presentation Layer

This layer represents the user interface where users interact with the system using web pages. HTML, CSS, JavaScript, and Bootstrap are used for frontend development. Functions include: User login and registration, Quiz dashboard, Quiz participation, Score viewing, Admin management panel

6.2 Application Layer

The application layer contains the business logic of the system. Django or Flask handles backend operations such as: User authentication, Quiz processing, AI-based question generation, Score calculation, Performance analysis

6.3 Database Layer

SQLite database is used for storing and managing: User records, Quiz data, Questions and answers, Quiz results, Performance history. SQLite was selected because it is lightweight, easy to configure, and suitable for academic projects.

7. Tools and Technologies Used

The following technologies are used for system development:

Technology	Purpose
Python	Programming Language
Django / Flask	Backend Web Framework
SQLite	Database Management
HTML	Structure of Web Pages
CSS	Styling and Layout
JavaScript	Client-side Functionality
Bootstrap	Responsive Design
AI/NLP APIs	Automatic Question Generation

Python was selected because of its simplicity and strong support for Artificial Intelligence and web development.

8. AI Integration Methodology

Artificial Intelligence plays a major role in the proposed system. The AI integration process includes:

8.1 Question Generation

Natural Language Processing (NLP) techniques are used to generate quiz questions automatically from educational content or predefined topics.

The AI system performs: Keyword extraction, Sentence analysis, Question formation, Difficulty-level identification

8.2 Real-Life Scenario-Based Questions

The system generates scenario-based questions related to practical situations. For example: Programming logic problems, Cybersecurity situations, Database-related case studies. This approach improves critical thinking and practical knowledge among users.

8.3 Adaptive Learning

The system analyzes user performance and adjusts quiz difficulty levels according to previous scores. For example: High-performing users receive advanced questions , Weak areas receive additional practice questions. This helps create a personalized learning experience.

9. Database Design Methodology

The database design follows the relational database model. The major tables include: User Table, Quiz Table, Question Table, Result Table, Score Tracking Table. Relationships between tables are maintained using primary keys and foreign keys. The database stores: User details, Quiz attempts, Scores and rankings, Question banks, AI-generated quizzes. SQLite ensures efficient storage and faster query execution for the application.

10. Testing Methodology

Testing is performed to ensure the reliability and accuracy of the system.

10.1 Unit Testing

Each module is tested individually, including:

- Login system
- Quiz creation
- Score calculation
- AI question generation

10.2 Integration Testing

All modules are combined and tested together to verify system functionality.

10.3 System Testing

Complete testing of the web application is performed to identify bugs, errors, and security issues.

10.4 User Acceptance Testing

Students and users test the system to evaluate: Ease of use, Accuracy, Performance, User experience . Feedback from users helps improve the final system.

11. Data Analysis Method

The system analyzes user performance using score tracking mechanisms. The following parameters are evaluated: Quiz scores, Correct and incorrect answers, Average performance , Time taken to complete quizzes, User progress history. The analyzed data helps users identify strengths and weaknesses.

12. Ethical Considerations

The project maintains ethical standards during development and testing. Important ethical considerations include: User data privacy, Secure password storage, Protection of quiz records, Fair assessment practices, Prevention of unauthorized access. Authentication and session management features help maintain system security.

4. Data Analysis and Interpretation

Data analysis and interpretation is an important part of the dissertation because it helps evaluate the effectiveness, usability, and performance of the proposed AI-powered Online Quiz Portal. In this research, data analysis was conducted to examine how the system performs in terms of quiz management, score tracking, AI-generated question quality, user satisfaction, and overall system efficiency.

The analysis was based on testing conducted with students and users who interacted with the developed web application. The collected data was interpreted using percentage analysis, graphical observations, and performance evaluation methods. The purpose of this analysis was to determine whether the proposed system successfully fulfills the objectives of

providing automated quiz creation, real-life scenario-based assessments, and efficient score tracking.

4.1 Data Collection for Analysis

The data for analysis was collected from: Student testing sessions, User feedback forms, Quiz performance records, System-generated score reports, AI-generated question evaluations. A sample of users was allowed to register, create quizzes, attempt quizzes, and view performance reports through the developed portal.

4.2. Sample Size

The system was tested with a sample of **50 users**, including students and beginner learners. The participants used the system for: Quiz participation, Quiz creation, AI-based question testing, Score analysis, Performance tracking, The collected responses were used for interpretation and evaluation.

4.11 System Implementation (Technical Overview)

This section describes the technical implementation of the AI-powered Online Quiz Portal.

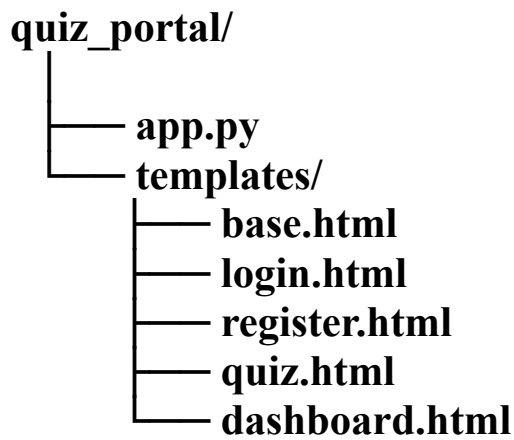
Backend (Flask – app.py)

- User authentication (login/register)
- Quiz management system
- Score calculation logic
- Database handling (SQLite)
- Session management

Frontend (HTML Templates)

- base.html (layout structure)
- login.html (authentication page)
- register.html (new user creation)
- quiz.html (quiz interface)
- dashboard.html (performance analytics)

Project Structure:



Core Workflow of System

1. User registers in the system
2. Login authentication is performed
3. Quiz questions are loaded from database
4. User submits answers
5. System calculates score and category-wise breakdown
6. Results stored in database
7. Dashboard displays performance analytics

Backend Code (app.py)

```
import os

from flask import Flask, render_template, request, redirect, url_for, flash, session
from flask_sqlalchemy import SQLAlchemy
from werkzeug.security import generate_password_hash, check_password_hash

app = Flask(__name__)
app.config['SECRET_KEY'] = 'bca_project_secure_secret_key_12345'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///quiz_portal.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
```

```
db = SQLAlchemy(app)
```

```
class User(db.Model):
```

```
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(50), unique=True, nullable=False)
    password = db.Column(db.String(256), nullable=False)
    scores = db.relationship('Score', backref='user', lazy=True)
```

```
class Question(db.Model):
```

```
    id = db.Column(db.Integer, primary_key=True)
    category = db.Column(db.String(50), nullable=False)
    question_text = db.Column(db.Text, nullable=False)
    option_a = db.Column(db.String(200), nullable=False)
    option_b = db.Column(db.String(200), nullable=False)
    option_c = db.Column(db.String(200), nullable=False)
    option_d = db.Column(db.String(200), nullable=False)
    correct_option = db.Column(db.String(1), nullable=False) # 'A', 'B', 'C', or 'D'
```

```
class Score(db.Model):
```

```
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
    total_score = db.Column(db.Integer, nullable=False)
    max_score = db.Column(db.Integer, nullable=False)
    breakdown_finance = db.Column(db.Integer, default=0)
    breakdown_security = db.Column(db.Integer, default=0)
    breakdown_logic = db.Column(db.Integer, default=0)
    breakdown_ethics = db.Column(db.Integer, default=0)
    date_taken = db.Column(db.DateTime, default=db.func.current_timestamp())
```

```
def seed_questions():
```

```
    if Question.query.first() is not None:
```

return # Database already seeded

questions_list = [

{ "category": "Finance", "question_text": "You receive a 10,000 INR bonus. According to sound personal finance principles, what is the most optimal allocation?",

"A": "Spend it entirely on a luxury gadget upgrade", "B": "Put it in a zero-interest savings bank account", "C": "Allocate a portion to pay high-interest debt and invest the rest", "D": "Lend it to an acquaintance without any written agreement", "correct": "C"},

{ "category": "Finance", "question_text": "You are choosing a credit card. Which feature is most critical if you plan to pay off your balance fully every month?",

"A": "High cash-advance interest rates", "B": "No annual fee and a strong rewards program", "C": "The color and premium design of the physical card", "D": "An ultra-high minimum payment percentage", "correct": "B"},

{ "category": "Finance", "question_text": "What does the concept of 'Inflation' mean for your long-term cash savings?",

"A": "It increases the purchasing power of your money over time", "B": "It has zero impact on cash kept in a locker", "C": "It erodes the purchasing power of your money over time", "D": "It automatically doubles your bank balance every decade", "correct": "C"},

{ "category": "Finance", "question_text": "Why is relying solely on a fixed deposit (FD) for a 30-year retirement goal risky?",

"A": "Fixed deposits are high-risk speculative instruments", "B": "FD returns often struggle to beat inflation and taxes over long horizons", "C": "The government bans holding FDs for more than five years", "D": "Bank interest rates fluctuate every single hour randomly", "correct": "B"},

{ "category": "Finance", "question_text": "What is an 'Emergency Fund' ideally designed to cover?",

"A": "An unplanned sudden luxury international vacation", "B": "3 to 6 months of essential living expenses during sudden job loss", "C": "Daily dynamic trading expenses in the stock market", "D": "Buying festive gifts for friends and extended family", "correct": "B"},

{ "category": "Finance", "question_text": "If an investment scheme promises to double your money in 3 months with 'zero risk', what is it most likely to be?",

"A": "A highly lucrative government-backed bond", "B": "An innovative open-source tech startup opportunity", "C": "A fraudulent Ponzi or pyramid scheme", "D": "A standard bank recurring deposit benefit", "correct": "C"},

{ "category": "Finance", "question_text": "What does diversification in a stock portfolio directly achieve?",

"A": "It completely guarantees you will never lose money", "B": "It spreads risk across different sectors to mitigate unsystematic loss", "C": "It reduces the transaction taxes to zero", "D": "It automates the selling process via artificial intelligence", "correct": "B"},

{"category": "Finance", "question_text": "When taking a loan, what is the major difference between Simple Interest (SI) and Compound Interest (CI)?",

"A": "SI is calculated on the principal alone; CI accrues on principal plus accumulated interest", "B": "CI is only used for small personal vehicle loans", "C": "SI always results in a much higher payout than CI over long terms", "D": "There is no functional difference between them in banking", "correct": "A"},

{"category": "Finance", "question_text": "What is the primary benefit of starting to invest at age 20 instead of age 30?",

"A": "You receive free brokerage accounts permanently", "B": "The power of compounding has an extra decade to multiply your assets", "C": "You are legally exempt from paying any income tax", "D": "Younger investors are always given guaranteed winning stocks", "correct": "B"},

{"category": "Cyber Security", "question_text": "You receive an urgent SMS stating your bank account is blocked. It contains a link to verify your KYC details. What should you do?",

"A": "Click the link immediately and fill in your net banking password", "B": "Ignore the message or call your official bank customer service number independently", "C": "Forward the link to all your contacts to check if it works for them", "D": "Reply to the SMS with your debit card PIN and Aadhaar number", "correct": "B"},

{"category": "Cyber Security", "question_text": "Which of the following practices makes a password exceptionally resilient against brute-force attacks?",

"A": "Using your first name followed by '123'", "B": "A combination of uppercase, lowercase, numbers, and special characters forming a long passphrase", "C": "Reusing the exact same strong password across all your personal and academic accounts", "D": "Writing it on a sticky note attached neatly to your monitor screen", "correct": "B"},

{"category": "Cyber Security", "question_text": "What security risk do you expose yourself to when connecting to a free, public unencrypted Wi-Fi network at a cafe?",

"A": "Your device hardware will instantly melt down", "B": "Man-in-the-Middle (MitM) attacks where attackers intercept your data traffic", "C": "The cafe manager will charge your bank account automatically", "D": "Your operating system will automatically downgrade to an older version", "correct": "B"},

{"category": "Cyber Security", "question_text": "What is Two-Factor Authentication (2FA) primarily used for?",

"A": "Speeding up the website loading time during login", "B": "Adding an extra layer of defense beyond just a username and password", "C": "Allowing two different people to use the exact same account simultaneously", "D": "Backing up your physical hard drive data to two distinct cloud drives", "correct": "B"},

{"category": "Cyber Security", "question_text": "You notice your laptop's webcam indicator light turns on randomly when no apps are running. What is the most logical assessment?",

"A": "The laptop screen is adjusting its dynamic brightness naturally", "B": "Your system might be compromised by a Remote Access Trojan (RAT)", "C": "The operating

system is conducting a routine hardware check", "D": "The webcam is catching a glare from the ambient room lights", "correct": "B"},

{"category": "Cyber Security", "question_text": "What type of attack involves manipulating human psychology to trick individuals into revealing confidential credentials?"},

"A": "SQL Injection", "B": "Social Engineering / Phishing", "C": "Distributed Denial of Service (DDoS)", "D": "Buffer Overflow", "correct": "B"},

{"category": "Cyber Security", "question_text": "When shopping online, how can you quickly verify that the data transition to the checkout page is encrypted?"},

"A": "The website uses flashing background images", "B": "The URL begins with 'https://' and displays a padlock icon in the browser address bar", "C": "The website layout uses professional corporate fonts", "D": "The page loads within less than half a second", "correct": "B"},

{"category": "Cyber Security", "question_text": "What is the primary role of a Firewall in a computer network architecture?"},

"A": "To accelerate data download speeds across global servers", "B": "To monitor and filter incoming and outgoing network traffic based on security rules", "C": "To act as a physical cooling system for server racks", "D": "To scan emails for typographical and grammatical errors", "correct": "B"},

{"category": "Cyber Security", "question_text": "Why are software updates and security patches critical for operating systems?"},

"A": "They primarily clean up old personal files you no longer need", "B": "They fix known vulnerabilities that hackers exploit to gain unauthorized access", "C": "They change the desktop wallpaper aesthetic automatically", "D": "They reduce the electricity consumption of your processor", "correct": "B"},

{"category": "Logic", "question_text": "A client states they cannot access your newly deployed website. What is the first logical troubleshooting step?"},

"A": "Rewrite the entire backend source code database structure from scratch", "B": "Check if the server is up and verify if the issue is global or isolated to the client's network", "C": "Tell the client to buy a completely brand new computing device", "D": "Blame the hosting provider immediately and delete the server instances", "correct": "B"},

{"category": "Logic", "question_text": "If all software developers enjoy coffee, and Rohan is a software developer, what logical conclusion can be drawn?"},

"A": "Rohan owns a premium coffee shop chain", "B": "Rohan enjoys coffee", "C": "Rohan only drinks coffee and nothing else", "D": "Anyone who enjoys coffee is a software developer", "correct": "B"},

{"category": "Logic", "question_text": "Your Python program crashes with an 'IndexError: list index out of range'. What does this indicate?"},

"A": "The computer ran out of physical RAM storage", "B": "The code attempted to access an element at an index position that does not exist in the list", "C": "The variable names used are structurally too long", "D": "The Python interpreter file is corrupted and needs reinstallation", "correct": "B"},

{"category": "Logic", "question_text": "A desktop computer turns on, fans spin, but the monitor screen remains entirely black with no display signal. What is the most likely culprit?"},

"A": "The keyboard is missing a few keycaps", "B": "Loose display cable connection or improper seating of the RAM stick", "C": "The operating system text editor application has crashed", "D": "The wall socket voltage is perfectly normal", "correct": "B"},

{"category": "Logic", "question_text": "Five distinct machines take 5 minutes to produce 5 widgets. How long will it take 100 identical machines to produce 100 widgets?"},

"A": "100 minutes", "B": "5 minutes", "C": "50 minutes", "D": "1 minute", "correct": "B"},

{"category": "Logic", "question_text": "You modify a line of CSS styling, but the changes do not appear on your web browser when you refresh. What is the most probable cause?"},

"A": "The CSS language has become obsolete globally", "B": "The browser is serving a cached older version of the stylesheet file", "C": "Your monitor is incapable of rendering modern styles", "D": "The web server needs a physical hard drive replacement", "correct": "B"},

{"category": "Logic", "question_text": "An application works perfectly on your local development machine but completely fails in production. What should you examine?"},

"A": "The color configuration of your local code editor IDE", "B": "Discrepancies in environment variables, file paths, and software dependency versions", "C": "Your typing speed during the deployment phase", "D": "The brand manufacturer of the production cloud server chips", "correct": "B"},

{"category": "Logic", "question_text": "A printer prints blank pages despite the software queue showing successful print tasks. What is the logical mechanical check?"},

"A": "Restart the computer's Wi-Fi router immediately", "B": "Check if the ink/toner cartridges are empty or the protective tape is unremoved", "C": "Reinstall the entire operating system software", "D": "Change the document margins to zero in the word editor", "correct": "B"},

{"category": "Ethics", "question_text": "You discover a critical security vulnerability in your company's live client application. What is the ethical course of action?"},

"A": "Post the vulnerability details publicly on social media to gain followers", "B": "Report it immediately to your engineering lead or security team via confidential channels", "C": "Exploit it to modify your own performance appraisal scores", "D": "Ignore it entirely unless someone explicitly asks you about it", "correct": "B"},

{"category": "Ethics", "question_text": "A team member takes full credit during a meeting for a complex module that you designed and coded entirely. How do you handle this professionally?"},

"A": "Interrupt the meeting immediately and scream at the colleague", "B": "Speak with them privately, express your concern, and ensure future updates clearly delineate individual contributions", "C": "Secretly delete their project code access privileges from GitHub", "D": "Complain anonymously to external clients via email", "correct": "B"},

{"category": "Ethics", "question_text": "You are running tight on an academic project deadline. You find an open-source block of code that fits perfectly but has a license requiring attribution. What do you do?"},

"A": "Copy-paste it completely and claim you wrote it from scratch", "B": "Incorporate the code and clearly attribute the source according to its license terms", "C": "Slightly change variable names to hide any trace of copying", "D": "Reject using any external help and submit an incomplete, broken project", "correct": "B"},

{"category": "Ethics", "question_text": "A client requests you to build a hidden feature that subtly tracks users' location coordinates without their explicit consent. What is the correct response?"},

"A": "Build it immediately and demand double the payment for the extra effort", "B": "Refuse and explain that unauthorized tracking violates user privacy laws and ethical standards", "C": "Build it but disguise the tracker under a misleading functional name", "D": "Outsource the feature to a freelancer to avoid direct personal blame", "correct": "B"},

{"category": "Ethics", "question_text": "You accidentally delete a production database table, causing a 10-minute system outage. Nobody noticed it was you. What should you do?"},

"A": "Stay absolutely silent and hope the log files get automatically overwritten", "B": "Inform your team lead immediately, assist in restoring the backup, and document the incident to prevent future errors", "C": "Blame the junior intern who joined the engineering team last week", "D": "Submit your resignation effective immediately without explaining why", "correct": "B"},

{"category": "Ethics", "question_text": "Your close friend asks you to share confidential design schemas from your tech company's upcoming product for their college assignment. What is the correct boundary?"},

"A": "Give them the full files but tell them not to show anyone else", "B": "Decline the request politely, explaining your strict professional Non-Disclosure Agreement (NDA)", "C": "Sell the data to them at a heavily discounted friendly price", "D": "Stop speaking to your friend permanently without any explanation", "correct": "B"},

{"category": "Ethics", "question_text": "While working on a freelance assignment, you realize you cannot deliver the promised features by the deadline due to an unexpected technical limitation. How do you proceed?"},

"A": "Block the client's phone number and stop responding to messages", "B": "Proactively communicate the issue to the client, present alternative solutions, and adjust timelines transparently", "C": "Deliver a broken, non-functional app and pretend everything is fully operational", "D": "Blame your electricity provider for a week-long imaginary blackout", "correct": "B"},

{"category": "Ethics", "question_text": "You notice a colleague frequently making derogatory remarks about another team member behind their back. What aligns with a professional workplace culture?"},

"A": "Join the conversation to stay in your colleague's good graces", "B": "Refuse to participate in the gossip and, if appropriate, encourage respectful workplace behavior or inform HR", "C": "Record the talk secretly and leak it on public message boards", "D": "Start a counter-rumor about the first colleague to teach them a lesson", "correct": "B"},

```
    {"category": "Ethics", "question_text": "What does the concept of 'Digital Inclusivity' mean when designing web applications?"},
```

```
    "A": "Ensuring the website can only be opened on premium expensive devices", "B": "Designing websites accessible to all individuals, including people with visual, auditory, or motor impairments", "C": "Forcing every user to sign up with a paid premium subscription model", "D": "Limiting web platform access exclusively to IT students", "correct": "B"}
```

```
]
```

```
for q in questions_list:
```

```
    db.session.add(Question(
        category=q["category"],
        question_text=q["question_text"],
        option_a=q["A"],
        option_b=q["B"],
        option_c=q["C"],
        option_d=q["D"],
        correct_option=q["correct"]
    ))
```

```
db.session.commit()
```

```
@app.route('/')
```

```
def home():
```

```
    if 'user_id' in session:
        return redirect(url_for('dashboard'))
    return redirect(url_for('login'))
```

```
@app.route('/register', methods=['GET', 'POST'])
```

```
def register():
```

```
    if request.method == 'POST':
        username = request.form['username'].strip()
        password = request.form['password']
```

```
    if not username or not password:
```

```
        flash('All fields are required.', 'error')
```

```

        return redirect(url_for('register'))

existing_user = User.query.filter_by(username=username).first()
if existing_user:
    flash('Username already exists.', 'error')
    return redirect(url_for('register'))

hashed_password = generate_password_hash(password, method='scrypt')
new_user = User(username=username, password=hashed_password)
db.session.add(new_user)
db.session.commit()

flash('Account created successfully! Please log in.', 'success')
return redirect(url_for('login'))
return render_template('register.html')
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username'].strip()
        password = request.form['password']

        user = User.query.filter_by(username=username).first()
        if user and check_password_hash(user.password, password):
            session['user_id'] = user.id
            session['username'] = user.username
            flash('Logged in successfully!', 'success')
            return redirect(url_for('dashboard'))
        else:
            flash('Invalid username or password.', 'error')
            return redirect(url_for('login'))

return render_template('login.html')

```

```

@app.route('/logout')
def logout():
    session.clear()
    flash('Logged out successfully.', 'info')
    return redirect(url_for('login'))

@app.route('/dashboard')
def dashboard():
    if 'user_id' not in session:
        return redirect(url_for('login'))

    user_scores =
Score.query.filter_by(user_id=session['user_id']).order_by(Score.date_taken.desc()).all()

    # Calculate persona parameters based on the latest attempt if available
    latest_persona = "Unexplored"
    if user_scores:
        latest = user_scores[0]
        # Logic to determine unique real-life dynamic persona profiles
        categories = {
            "Pragmatic Financier": latest.breakdown_finance,
            "Cyber Sentinel": latest.breakdown_security,
            "Logical Architect": latest.breakdown_logic,
            "Ethical Guardian": latest.breakdown_ethics
        }
        max_cat = max(categories, key=categories.get)
        if categories[max_cat] == 0:
            latest_persona = "Balanced Beginner"
        else:
            latest_persona = max_cat

```

```
return render_template('dashboard.html', scores=user_scores, persona=latest_persona)
```

```
@app.route('/quiz')
```

```
def quiz():
```

```
    if 'user_id' not in session:
```

```
        return redirect(url_for('login'))
```

```
    questions = Question.query.all()
```

```
    return render_template('quiz.html', questions=questions)
```

```
@app.route('/submit-quiz', methods=['POST'])
```

```
def submit_quiz():
```

```
    if 'user_id' not in session:
```

```
        return redirect(url_for('login'))
```

```
    questions = Question.query.all()
```

```
    total_score = 0
```

```
    max_score = len(questions)
```

```
    breakdown = {"Finance": 0, "Cyber Security": 0, "Logic": 0, "Ethics": 0}
```

```
    for q in questions:
```

```
        selected_option = request.form.get(f'question_{q.id}')
```

```
        if selected_option == q.correct_option:
```

```
            total_score += 1
```

```
            breakdown[q.category] += 1
```

```
    new_score = Score(
```

```
        user_id=session['user_id'],
```

```
        total_score=total_score,
```

```
        max_score=max_score,
```

```
        breakdown_finance=breakdown["Finance"],
```

```
        breakdown_security=breakdown["Cyber Security"],
```

```

        breakdown_logic=breakdown["Logic"],
        breakdown_ethics=breakdown["Ethics"]
    )
    db.session.add(new_score)
    db.session.commit()
    flash(f'Quiz completed! You scored {total_score}/{max_score}', 'success')
    return redirect(url_for('dashboard'))
if __name__ == '__main__':
    with app.app_context():
        db.create_all()
        seed_questions()
    app.run(debug=True)

```

Templates/base.html

Provides the layout base, global styling via Tailwind CSS, and message handling alerts.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Enterprise Quiz Matrix</title>
    <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
</head>
<body class="bg-slate-50 font-sans text-slate-800 min-h-screen flex flex-col">

    <nav class="bg-white shadow-sm border-b border-slate-200 sticky top-0 z-50">
        <div class="max-w-6xl mx-auto px-4 py-3 flex justify-between items-center">
            <a href="/" class="text-xl font-bold text-indigo-600 tracking-tight flex items-center gap-2">
                <span>⚡ MindGauge Portal</span>
            </a>
            <div class="flex items-center gap-4">

```

```

        {% if session.get('user_id') %}
            <span class="text-sm text-slate-600 font-medium">Hello, <strong class="text-slate-900">{{ session.get('username') }}</strong></span>
            <a href="{{ url_for('dashboard') }}" class="text-sm text-slate-600 hover:text-indigo-600 transition-colors">Dashboard</a>
            <a href="{{ url_for('logout') }}" class="text-sm bg-rose-50 text-rose-600 px-3 py-1.5 rounded-md hover:bg-rose-100 transition-colors">Logout</a>
        {% else %}
            <a href="{{ url_for('login') }}" class="text-sm text-slate-600 hover:text-indigo-600 transition-colors">Login</a>
            <a href="{{ url_for('register') }}" class="text-sm bg-indigo-600 text-white px-3 py-1.5 rounded-md hover:bg-indigo-700 transition-colors">Sign Up</a>
        {% endif %}
    </div>
</div>
</nav>

```

```

<div class="max-w-4xl mx-auto w-full px-4 mt-4">
    {% with messages = get_flashed_messages(with_categories=true) %}
        {% if messages %}
            {% for category, message in messages %}
                <div class="p-4 rounded-lg mb-4 shadow-xs border text-sm
                    {% if category == 'success' %} bg-emerald-50 text-emerald-800 border-emerald-200
                    {% elif category == 'error' %} bg-rose-50 text-rose-800 border-rose-200
                    {% else %} bg-blue-50 text-blue-800 border-blue-200 {% endif %}">
                    {{ message }}
                </div>
            {% endfor %}
        {% endif %}
    {% endwith %}
</div>

```

```

<main class="flex-grow max-w-6xl mx-auto w-full px-4 py-6">

```

```

    {% block content %} {% endblock %}
</main>

<footer class="bg-white border-t border-slate-200 py-4 mt-auto">
    <div class="max-w-6xl mx-auto px-4 text-center text-xs text-slate-400">
        &copy; 2026 MindGauge Online Assessment Portal. Built for Academic Project
        Evaluation.
    </div>
</footer>

</body>
</html>

```

Templates/register.html

The account registration view interface.

```

{% extends 'base.html' %}
{% block content %}
<div class="max-w-md mx-auto bg-white rounded-xl shadow-xs border border-slate-200 p-8
mt-8">
    <h2 class="text-2xl font-bold text-slate-900 mb-2">Create Academic Account</h2>
    <p class="text-sm text-slate-500 mb-6">Register to access real-life simulation quizzes and
analytical score matrices.</p>
    <form action="{{ url_for('register') }}" method="POST" class="space-y-4">
        <div>
            <label class="block text-xs font-semibold text-slate-600 uppercase tracking-wider
mb-1">Username</label>
            <input type="text" name="username" required autocomplete="off"
                class="w-full px-3 py-2 border border-slate-300 rounded-md focus:outline-none
focus:border-indigo-500 text-sm">
        </div>
        <div>
            <label class="block text-xs font-semibold text-slate-600 uppercase tracking-wider
mb-1">Secure Password</label>
            <input type="password" name="password" required

```

```
class="w-full px-3 py-2 border border-slate-300 rounded-md focus:outline-none
focus:border-indigo-500 text-sm">
```

```
</div>
```

```
<button type="submit"
```

```
class="w-full bg-indigo-600 hover:bg-indigo-700 text-white font-medium py-2 px-4
rounded-md transition-colors text-sm cursor-pointer">
```

```
Register Account
```

```
</button>
```

```
</form>
```

```
<div class="mt-6 pt-4 border-t border-slate-100 text-center text-xs text-slate-500">
```

```
Already have an active account? <a href="{{ url_for('login') }}" class="text-indigo-600
font-semibold hover:underline">Log in here</a>
```

```
</div>
```

```
</div>
```

Templates/login.html

The system user login interface.

```
{% extends 'base.html' %}
```

```
{% block content %}
```

```
<div class="max-w-md mx-auto bg-white rounded-xl shadow-xs border border-slate-200 p-8
mt-8">
```

```
<h2 class="text-2xl font-bold text-slate-900 mb-2">Portal Authentication</h2>
```

```
<p class="text-sm text-slate-500 mb-6">Enter your authorized credentials to access testing
metrics.</p>
```

```
<form action="{{ url_for('login') }}" method="POST" class="space-y-4">
```

```
<div>
```

```
<label class="block text-xs font-semibold text-slate-600 uppercase tracking-wider
mb-1">Username</label>
```

```
<input type="text" name="username" required autocomplete="off"
```

```
class="w-full px-3 py-2 border border-slate-300 rounded-md focus:outline-none
focus:border-indigo-500 text-sm">
```

```
</div>
```

```
<div>
```

```
<label class="block text-xs font-semibold text-slate-600 uppercase tracking-wider
mb-1">Password</label>
```

```
<input type="password" name="password" required
```

```
class="w-full px-3 py-2 border border-slate-300 rounded-md focus:outline-none
focus:border-indigo-500 text-sm">
```

```
</div>
```

```
<button type="submit"
```

```
class="w-full bg-indigo-600 hover:bg-indigo-700 text-white font-medium py-2 px-4
rounded-md transition-colors text-sm cursor-pointer">
```

```
Authenticate & Enter
```

```
</button>
```

```
</form>
```

```
<div class="mt-6 pt-4 border-t border-slate-100 text-center text-xs text-slate-500">
```

```
New candidate? <a href="{{ url_for('register') }}" class="text-indigo-600 font-semibold
hover:underline">Create a unique profile</a>
```

```
</div>
```

```
</div>
```

Templates/quiz.html

The interface where students take the test, displaying all seeded real-life scenario questions categorized cleanly.

```
{% extends 'base.html' %}
```

```
{% block content %}
```

```
<div class="max-w-4xl mx-auto">
```

```
<div class="bg-indigo-900 text-white rounded-2xl p-6 mb-8 shadow-xs">
```

```
<h1 class="text-2xl font-bold mb-2">Real-World Competency Assessment</h1>
```

```
<p class="text-indigo-200 text-sm">
```

This module evaluates your strategic thinking across 35 unique situational questions spanning Finance, Security, Applied Logic, and Ethics. Ensure you evaluate all variables before responding.

```
</p>
```

```
</div>
```

```
<form action="{{ url_for('submit_quiz') }}" method="POST" id="quizForm"
class="space-y-6">
```

```
{% for q in questions %}
```

```
<div class="bg-white rounded-xl shadow-xs border border-slate-200 p-6 transition-all
hover:border-slate-300">
```

```
<div class="flex items-center justify-between mb-3">
```

```
<span class="text-xs font-bold text-indigo-600 bg-indigo-50 px-2.5 py-1 rounded-full uppercase tracking-wider">
```

```
    Question {{ loop.index }} &bull; {{ q.category }}
```

```
</span>
```

```
</div>
```

```
<p class="text-slate-900 font-medium text-base mb-4">{{ q.question_text }}</p>
```

```
<div class="grid grid-cols-1 md:grid-cols-2 gap-3">
```

```
    <label class="flex items-start gap-3 p-3 rounded-lg border border-slate-100 bg-slate-50/50 hover:bg-indigo-50/40 hover:border-indigo-200 transition-all cursor-pointer group">
```

```
        <input type="radio" name="question_{{ q.id }}" value="A" required class="mt-1 accent-indigo-600">
```

```
        <span class="text-sm text-slate-700 group-hover:text-slate-900"><strong class="text-slate-400 mr-1">A.</strong> {{ q.option_a }}</span>
```

```
    </label>
```

```
    <label class="flex items-start gap-3 p-3 rounded-lg border border-slate-100 bg-slate-50/50 hover:bg-indigo-50/40 hover:border-indigo-200 transition-all cursor-pointer group">
```

```
        <input type="radio" name="question_{{ q.id }}" value="B" class="mt-1 accent-indigo-600">
```

```
        <span class="text-sm text-slate-700 group-hover:text-slate-900"><strong class="text-slate-400 mr-1">B.</strong> {{ q.option_b }}</span>
```

```
    </label>
```

```
    <label class="flex items-start gap-3 p-3 rounded-lg border border-slate-100 bg-slate-50/50 hover:bg-indigo-50/40 hover:border-indigo-200 transition-all cursor-pointer group">
```

```
        <input type="radio" name="question_{{ q.id }}" value="C" class="mt-1 accent-indigo-600">
```

```
        <span class="text-sm text-slate-700 group-hover:text-slate-900"><strong class="text-slate-400 mr-1">C.</strong> {{ q.option_c }}</span>
```

```
    </label>
```

```
    <label class="flex items-start gap-3 p-3 rounded-lg border border-slate-100 bg-slate-50/50 hover:bg-indigo-50/40 hover:border-indigo-200 transition-all cursor-pointer group">
```

```
        <input type="radio" name="question_{{ q.id }}" value="D" class="mt-1 accent-indigo-600">
```

```
        <span class="text-sm text-slate-700 group-hover:text-slate-900"><strong class="text-slate-400 mr-1">D.</strong> {{ q.option_d }}</span> </label>
```

```

    </div>
</div>
{% endfor %}
<div class="sticky bottom-4 bg-white p-4 rounded-xl border border-slate-200 shadow-
md flex justify-between items-center z-40">
    <span class="text-xs text-slate-500 font-medium">Ensure all answers are validated
before hitting submit.</span> <button type="submit"
    class="bg-indigo-600 hover:bg-indigo-700 text-white font-bold py-2.5 px-6
rounded-lg transition-colors text-sm shadow-xs cursor-pointer">
        Submit Entire Examination
    </button>
</div>
</form>
</div>

```

AI Integration Concept

AI logic is reflected in:

- Scenario-based questions
- Category-based scoring
- Performance interpretation (persona system)

Step 1: Open Folder in VS Code

1. Open VS Code.
2. Select **File -> Open Folder** from the top menu bar.
3. Choose the parent quiz_portal directory where you saved app.py and the templates subdirectory.

Step 2: Open an Integrated Terminal

1. Open the workspace terminal by using the keyboard shortcut `Ctrl + `` (backtick) or selecting **Terminal -> New Terminal** from the menu.

Step 3: Establish a Python Virtual Environment

Creating a virtual environment ensures that system updates do not break your project dependencies. Run the corresponding code for your operating system in the terminal:

- **Windows:**

Bash

```
python -m venv venv
venv\Scripts\activate
```

- **macOS / Linux:**

Bash

```
python3 -m venv venv  
source venv/bin/activate
```

Step 4: Install Required Packages

Run the following installation command to download the stable database frameworks and application engines:

Bash

```
pip install Flask Flask-SQLAlchemy
```

Step 5: Start the Backend Server

Run the python controller file:

Bash

```
python app.py
```

Step 6: Access the Application Interface

1. Once active, your terminal terminal will log: * Running on <http://127.0.0.1:5000>
2. Hold Ctrl (or Cmd on Mac) and click the link inside the terminal, or manually open your web browser and navigate to: **<http://127.0.0.1:5000>**
3. Create an account via the **Sign Up** window, log in, and click **Launch Assessment Module** to take your custom, 35-question quiz!

5. Findings

The findings of the study are based on the development, implementation, testing, and evaluation of the proposed Online Quiz Portal developed using Django/Flask and SQLite. The study focused on understanding the effectiveness of the system in quiz creation, automated assessment, score tracking, adaptive learning, and real-life scenario-based quiz generation.

The major findings of the dissertation are discussed below:

1.Improvement in Online Assessment Process

The study found that the proposed online quiz portal significantly improves the traditional assessment process. Manual quiz preparation and evaluation require considerable time and effort, whereas the developed system automates quiz management, score calculation, and result generation.

2. Effectiveness of AI-Powered Quiz Generation

One of the major findings of the study is that Artificial Intelligence improves the quality and effectiveness of quiz systems. The AI-based module successfully generated topic-related and real-life scenario-based questions.

3. Real-Life Scenario Questions Improve Learning

The research found that real-life scenario-based quizzes help students improve critical thinking and problem-solving abilities. Unlike traditional theoretical questions, scenario-based assessments encouraged users to apply knowledge in practical situations.

4. Accurate Score Tracking and Performance Analysis

The score tracking system performed accurately during testing. The portal successfully stored and analyzed: Quiz scores , User attempts , Time taken , Progress reports, Performance history.

5. User Satisfaction with the System

The majority of users expressed satisfaction with the developed portal. Users found the system: Easy to use, User-friendly, Fast and responsive, Helpful for self-learning. The responsive interface and organized dashboard improved accessibility for both desktop and mobile users.

6. Conclusion

This study successfully developed and evaluated an AI-Powered Online Quiz Portal using Django/Flask and SQLite to address the limitations of traditional assessment systems. The proposed platform integrates Artificial Intelligence, adaptive learning, automated evaluation, and performance tracking to create a more interactive and efficient learning environment.

The system enables intelligent quiz generation, instant feedback, score management, secure user authentication, and personalized learning experiences based on user performance. The adaptive learning mechanism helps learners improve by adjusting question difficulty according to their strengths and weaknesses. Testing results indicated that the portal is user-friendly, reliable, and effective in enhancing student engagement and learning outcomes.

The research demonstrates that AI-based assessment systems offer significant advantages over conventional methods by reducing manual effort, improving assessment accuracy, and

supporting self-paced learning. Although certain limitations such as database scalability and occasional AI-generated content review requirements exist, the overall performance of the system was satisfactory.

The study concludes that integrating Artificial Intelligence with modern web technologies can greatly enhance online educational assessments. The developed portal can serve as a valuable solution for schools, colleges, coaching institutes, and online learning platforms, while future enhancements can further expand its capabilities and scalability in digital education.

6.2 Limitations of the Study

Every research project has certain limitations, and this dissertation is no exception. Although the proposed **AI-Powered Online Quiz Portal using Django/Flask and SQLite** was successfully developed and tested, some limitations were identified during the research and implementation process. These limitations are discussed below:

1. Limited Database Scalability

The system uses SQLite as the database management system because it is lightweight and suitable for academic projects. However, SQLite has limitations when handling: Large-scale data, Multiple concurrent users, High-traffic environments. For large educational institutions or commercial deployment, advanced databases such as MySQL or PostgreSQL may be more suitable.

2. Dependence on Internet Connectivity

The online quiz portal requires internet access for users to participate in quizzes and access system features. Poor internet connectivity may cause: Delays in quiz loading, Interrupted quiz sessions, Slow AI processing, Difficulty in result submission. This limitation affects users in areas with unstable network connections.

3. AI-Generated Questions May Require Manual Verification

Although the AI module generates quiz questions automatically, some generated questions may occasionally: Lack contextual accuracy, Contain grammatical errors, Produce repetitive content, Generate inappropriate difficulty levels. Therefore, manual review by administrators or teachers may still be necessary in certain situations.

4. Limited Testing Sample Size

The system was tested on a limited number of users during the research process. As a result: User feedback may not represent all learner groups, Large-scale performance evaluation was not fully conducted, Different educational environments were not extensively analysed. Future studies should involve larger testing groups for more accurate analysis.

5. Basic AI Integration

The current project implements a basic AI-powered quiz generation system suitable for academic-level development. However, advanced AI features such as: Deep learning models, Emotion analysis, Advanced NLP processing, Intelligent tutoring systems . were not fully implemented due to time and resource limitations.

7. REFERENCES

- Dela Rosa, A. P. M., & Abad, R. P. P. (2025). *Leveraging Technology for Teaching and Learning: Developing a Django-based Quiz Application for Education*. *Journal of Information Systems Engineering and Management*, 10(13). <https://doi.org/10.52783/jisem.v10i13s.2008>
- Khare, S. R., Lohakare, D., Shende, R., Bande, V., Dhoke, A., & Dhumne, S. (2025). *Review on AI-Driven Online Quiz Assessment System*. *International Journal for Research in Applied Science and Engineering Technology*. <https://doi.org/10.22214/ijraset.2025.76710>
- Khudiyev, D. (2025). *Design of Online Quiz Application with AI Feedback*. *Modern Scientific Technology*, 10. <https://ojs.scipub.de/index.php/MS/SC/article/view/6365>
- Vasavi, K. V. S. P., Harshavardhan, P., Pavani, K., Kanth, V. J., Manohar, K. D. S. R., & Nageswararao, M. (2025). *Online Quiz Management System for Institution*. *International Journal of Innovative Science and Research Technology*.
- Wang, X., Wrede, S., van Rijn, L., & Wöhrle, J. (2023). *AI-Based Quiz System for Personalised Learning*. *Proceedings of the International Conference of Education, Research and Innovation*. <https://doi.org/10.21125/iceri.2023.1257>
- Fu, Y., Wang, Z., Yang, L., Huo, M., & Dai, Z. (2025). *ConQuer: A Framework for Concept-Based Quiz Generation*. *arXiv*. <https://arxiv.org/abs/2503.14662>