CHAPTER 1
INTRODUCTION

In a world where education needs a transformative approach to keep pace with the changing demands of the modern era, a groundbreaking gamified learning application, reshaping the future of education through innovative technology. By combining interactive lessons with engaging game mechanics, this app bridges the gap between traditional educational methods and the dynamic expectations of today's learners. Through personalized learning paths, collaborative challenges, and immersive experiences, this brings a fresh perspective to education, making learning both accessible and exciting for all ages.

1.1 OVERVIEW

In a world where education must continually adapt to keep pace with the rapid advancements of modern society, this app emerges as a transformative gamified learning application that combines interactive technology with educational excellence. In a time when digital learning is becoming increasingly prevalent, the application seamlessly integrates engaging game elements with rigorous academic content to create an immersive and enjoyable learning experience. This app is designed with the goal of making learning more accessible, personalized, and motivating for learners of all ages. It offers a diverse range of subjects and learning levels, allowing users to explore topics of interest and engage in interactive lessons that cater to their individual learning styles and paces. By incorporating game mechanics such as rewards, challenges, and leaderboards, it fosters a sense of achievement and healthy competition among learners, encouraging them to strive for excellence.

One of the key features is its adaptive learning algorithms, which tailor content and learning paths to suit each user’s needs and progress. This personalized approach optimizes the learning journey, helping users achieve their goals more effectively.

Additionally, the platform promotes social learning by enabling collaboration, discussion, and teamwork among users, enhancing the overall educational experience. As it continues to grow and evolve, its impact on the world of education becomes ever more profound. By empowering learners with the tools and knowledge needed to succeed in a rapidly changing world, the application represents a beacon of hope for a more engaging and equitable educational system. In the end it is not just a technological innovation it is a testament to the power of human creativity and collaboration in shaping a brighter future for learners everywhere.

1.2 PROBLEM DEFINITION

Game-based learning is a teaching approach that integrates game mechanics and design elements into educational settings to create engaging, motivating, and effective learning experiences. This method leverages the natural appeal of games, which provide structured, goal-oriented activities that challenge and stimulate learners while offering opportunities for exploration and discovery. Through the use of points, levels, leaderboards, and achievements, students are motivated to engage with the content and strive for mastery. One of the key advantages of game-based learning is its ability to cater to diverse learning styles and needs. By presenting educational material in a format that is interactive and enjoyable, this approach can help maintain students' interest and attention, particularly for those who may struggle with traditional teaching methods. Moreover, game-based learning can be adapted for use with students who have disabilities, providing an inclusive environment where everyone can participate and succeed.

Game-based learning also fosters friendly competition, which can be a powerful motivator for students. By introducing challenges and rewards, learners are encouraged to set goals and work towards achieving them, enhancing their focus and commitment. The presence of leaderboards and rankings can further motivate students to improve their performance and strive for excellence, all within a supportive and collaborative learning community. Another benefit of game-based learning is its ability to provide immediate feedback to students. This can help learners identify areas for improvement and adjust their approach accordingly, leading to a more personalized learning experience. Additionally, the use of games can facilitate deeper understanding and retention of complex concepts by allowing students to experiment with different strategies and solutions in a risk-free environment.

1.3 FEASIBILITY STUDY

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. "***FEASIBILITY STUDY***" is a test of system proposal according to its workability, impact of the organization, ability to meet needsand effective use of the resources.

It focuses on these major questions:

1. What are the user's demonstrable needs and how does a candidate system meet them?
2. What resources are available for given candidate system?
3. What are the likely impacts of the candidate system on the organization?
4. Whether it is worth to solve the problem?

During feasibility analysis for this project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this.

Steps in feasibility analysis

Eight steps involved in the feasibility analysis are:

Form a project team and appoint a project leader.

1. Prepare system flowcharts.
2. Define and identify characteristics of proposed system.
3. Determine and evaluate performance and cost effective of each proposed system.
4. Weight system performance and cost data.
5. Select the best-proposed system.

6. Prepare and report final project directive to management.

1.3.1 Technical Feasibility

A study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not.

- Can the work for the project be done with current equipment existing software technology & available personal?
- Can the system be upgraded if developed?
- If new technology is needed, then what can be developed? This is concerned with specifying equipment and software that will successfully satisfy the user requirement.

The technical needs of the system may include:

Front-end and back-end selection

An important issue for the development of a project is the selection of suitablefront-end and back-end. When we decided to develop the project, we went through anextensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project.

Front-end selection

1. Scalability and extensibility.
2. Robustness.
3. Must provide excellent reporting features with good printing support.
4. Platform independent.
5. Easy to debug and maintain.
6. Event driven programming facility.
7. Front end must support some popular back end like Ms Access.

According to the above stated features we selected HTML/CSS as thefront-end for developing our project.

Back-end Selection

1. Multiple user support.
2. Efficient data handling.
3. Provide inherent features for security.
4. Efficient data retrieval and maintenance.
5. Operating System compatible.
6. Easy to install.
7. Easy to implant with the Front-end.

The technical feasibility is frequently the most difficult area encountered at this stage. It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed system.

1.3.2 Economical Feasibility

Economic justification is generally the "Bottom Line" consideration for most systems.Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weight the cost and the benefits associated with the candidate system and if it suits the basic purpose of the organization.

The financial and the economic questions during the preliminary investigation are verified to estimate the following:

1. The cost to conduct a full system investigation.
2. The cost of hardware and software for the class of application being considered.
3. The benefits in the form of reduced cost.
4. The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide increased profits.

This feasibility checks whether the system can be developed with the availablefunds. Our system does not require enormous amount of money to be developed. This can be done economically if planned judicially, so it is economically feasible.

1.3.3 Operational Feasibility

It is mainly related to human organizations and political aspects. The points to be considered are:

1. What changes will be brought with the system?
2. What organization structures are disturbed?
3. What new skills will be required?

The system is operationally feasible as it very easy for the end users to operateit. It only needs basic information about Windows platform.

1.3.4 Schedule Feasibility

Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems. A feasibility study is a comprehensive evaluation of a proposed project that evaluates all factors critical to its success in order to assess its likelihood of success. Business success can be defined primarily in terms of ROI.

In a feasibility study, a proposed plan or project is evaluated for its practicality. As part of a feasibility study, a project or venture is evaluated for its viability in order to determine whether it will be successful. A well-designed study should offer a historical background of the business or project, such as a description of the product orservice, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

CHAPTER 2

LITERATURE SURVEY

1. **“**Board Game Design for Python Programming Education” (2023)

Authors: Li-Wen Huang, Li-Wei Chen, Po-Hsun Cheng

Description: The paper discusses the design and development of a board game aimed at teaching Python programming concepts to students in an engaging and interactive way. By combining educational content with game mechanics, the board game offers a unique approach to learning programming that can appeal to various learning styles and levels of experience. The design process involves creating a board game that incorporates core Python programming concepts such as data types, control structures, functions, and libraries. Players navigate the game board by completing coding challenges and solving puzzles that test their knowledge and understanding of Python. The game uses cards, tokens, and dice to introduce elements of chance and strategy, keeping players engaged and motivated throughout the learning process.

The paper outlines how the board game is structured to progressively introduce programming concepts, allowing players to build a solid foundation of knowledge as they advance through the game. Players earn points and rewards for successfully completing challenges, and the game features different levels of difficulty to accommodate beginners and more advanced learners. The board game promotes collaboration and healthy competition among players, as they can work together to solve coding problems or compete to achieve the highest score. This social aspect of the game enhances the learning experience, making it more enjoyable and memorable for participants. The paper highlights the benefits of using a board game to teach Python programming, including increased engagement, motivation, and retention of knowledge. By offering a hands-on and interactive learning experience, the board game helps demystify programming concepts and encourages students to apply what they have learned in a practical setting.

2.“Design and Implementation of a Gamification-Based Web Application for Learning High-School Physics” (2023).

Authors: T. Katanosaka, M. F. F. Khan and K. Sakamura

Description: The paper discusses the design and implementation of a gamification-based web application aimed at improving the learning experience for high-school physics students. By incorporating game elements such as points, badges, and leaderboards, the application seeks to engage students and motivate them to actively participate in their learning journey. The design process begins with the creation of interactive modules that cover core physics concepts such as motion, energy, and electricity. These modules are structured around problem-solving activities, virtual experiments, and challenges that encourage students to apply their theoretical knowledge in practical scenarios. By offering a variety of activities, the application caters to different learning styles and levels of proficiency. The application uses adaptive learning paths to personalize the educational experience for each student. These paths tailor the content and challenges based on individual progress, allowing students to focus on areas where they need improvement and advance at their own pace. This personalized approach not only helps students overcome learning obstacles but also promotes continuous improvement and mastery of physics concepts.

Incorporating gamification elements such as rewards, achievements, and progress tracking enhances student motivation and engagement. Students can earn points and badges for successfully completing challenges and can track their progress through the application. Leaderboards foster a sense of healthy competition, encouraging students to strive for excellence while learning from their peers. The paper emphasizes the benefits of using gamification in high-school physics education, including increased engagement, motivation, and retention of knowledge. By providing a dynamic and interactive learning environment, the web application makes physics more approachable and enjoyable for students.

3. “Designing Gamified Application: An Effective Integration of Augmented Reality to Support Learning” (2022).

Authors: Sehar Shahzad Farooq, Hameedur Rahman, Syed Ali Raza, Muhammad Raees

Description: The project aims to revolutionize traditional learning methods by creating a gamified application that seamlessly integrates augmented reality (AR) technology to enhance the learning experience. Through the utilization of AR, users will immerse themselves in captivating learning activities that seamlessly blend real-world environments with digital elements. This innovative approach aims to make learning more interactive and engaging by providing users with hands-on experiences that transcend the limitations of traditional educational methods. The application will offer a wide range of educational content spanning various subjects and disciplines, catering to the diverse learning styles and preferences of its users. Whether it's exploring the depths of space in astronomy, dissecting virtual organisms in biology, or delving into historical events through immersive re-enactments, users will have access to a rich array of educational experiences tailored to their interests. Key to the application's effectiveness is its integration of gamification techniques, which serve to motivate users and drive their engagement. Through features such as rewards, challenges, and progress tracking, users will be incentivized to actively participate in learning activities and strive to achieve their educational objectives. This gamified approach not only enhances user motivation but also fosters a sense of accomplishment and progression as users navigate through the learning content.

Ultimately, the goal of the project is to provide an innovative platform that redefines the way we approach education. By harnessing the power of AR technology, the application aims to make learning more accessible, engaging, and effective than ever before. Whether used in formal educational settings, informal learning environments, or for self-directed study, the application seeks to empower learners of all ages to explore, discover, and grow through immersive, gamified learning experiences. Through its innovative blend of technology and education, the application aspires to inspire a new generation of lifelong learners and shape the future of education in the digital age.

4. “Increasing Student's Engagement in Study Skills Course via Gamification” (2021).
Author: A. A. Rahman, I. I. Abdullah, N. A. Talkis, N. F. Jamal and S. A. Razak
Description: In the dynamic landscape of education, enhancing student engagement stands as a cornerstone for fostering effective learning outcomes. Recognizing the significance of student engagement, particularly in study skills courses, this project endeavors to elevate student participation and motivation through the implementation of gamification strategies. At its core, the project seeks to transform the study skills course into an interactive and immersive learning experience by infusing it with elements of gamification. By leveraging gamification principles such as rewards, challenges, and progress tracking, students are incentivized to actively engage with course materials and activities, thereby enhancing their learning journey. The incorporation of gamification into the study skills course serves multiple purposes. Firstly, it introduces an element of excitement and enjoyment, turning what may otherwise be perceived as mundane tasks into engaging challenges. Through the application of game-like features such as points, levels, and leaderboards, students are motivated to strive for excellence and compete with themselves or their peers, fostering a sense of achievement and healthy competition. Moreover, gamification provides opportunities for personalized learning experiences, catering to the diverse needs and preferences of individual students. By allowing students to progress at their own pace and providing instant feedback on their performance, gamified elements empower students to take ownership of their learning journey and make meaningful strides towards improvement. Furthermore, gamification encourages active participation and collaboration among students, fostering a supportive learning community where knowledge sharing and peer support thrive. By incorporating collaborative challenges, group competitions, and social features into the study skills course, students not only benefit from shared learning experiences but also develop essential teamwork and communication skills essential for success in academia and beyond. Additionally, gamification serves as a powerful tool for enhancing student motivation and persistence in the face of challenges. Through the implementation of progressive challenges and rewards tied to specific learning milestones, students are

encouraged to persevere through obstacles and setbacks, building resilience and a growth mindset along the way.

Ultimately, the integration of gamification into the study skills course aims to create a dynamic and engaging learning environment that empowers students to develop essential study skills while fostering a love for learning. By tapping into the inherent appeal of gamified experiences, this project seeks to increase student engagement, motivation, and ultimately, academic success in the study skills course and beyond.

5. “EvalSeer: An Intelligent Gamified System for Programming Assignments Assessment” (2021)**.**

Authors: R. Nabil, N. E. Mohamed, A. Mahdy, K. Nader, S. Essam and E. Eliwa **Description:** The paper discusses EvalSeer, an intelligent gamified system designed to assess programming assignments effectively while enhancing the learning experience for students. The system leverages artificial intelligence (AI) and gamification to provide immediate, personalized feedback and encourage student engagement and motivation. EvalSeer uses AI algorithms to automatically evaluate programming assignments, offering students detailed feedback on their code's correctness, efficiency, and style. The system can identify common coding errors and suggest improvements, helping students understand their mistakes and learn how to correct them. By automating the assessment process, EvalSeer provides consistent and unbiased evaluations, saving instructors time and effort. The system incorporates gamification elements such as points, badges, and leaderboards to create a more engaging and competitive learning environment. Students earn rewards based on their performance, motivating them to strive for excellence and improve their coding skills. The use of leaderboards fosters healthy competition among students, encouraging them to learn from one another and strive for higher rankings. EvalSeer also provides adaptive learning paths that tailor challenges and assignments to individual students' abilities and progress. This personalized approach ensures that students receive tasks suited to their skill levels, promoting continuous learning and improvement.

The paper highlights the benefits of using EvalSeer in programming education, including increased student engagement, motivation, and knowledge retention. By providing immediate and detailed feedback, the system helps students understand their strengths and weaknesses, enabling them to make targeted improvements.

6. “An Innovative Multi-Layer Gamification Framework for Improved STEM Learning Experience” (2020).

Authors: Dan Zhao, Jim Playfoot, Carmine De Nicola, Giuseppe Guarino, Marilena Bratu , Fabio Di Salvadore6, And Gabriel-Miro Muntean

Description: The development of an innovative multi-layer gamification framework for enhanced STEM (Science, Technology, Engineering, and Mathematics) learning experiences represents a significant advancement in educational technology. This framework integrates multiple levels of gamification strategies to create a dynamic and immersive learning environment that caters to the diverse needs and preferences of STEM learners. At its core, the framework employs a multi-layered approach to gamification, incorporating various elements such as game mechanics, narrative storytelling, social interactions, and performance feedback to engage and motivate learners. By offering a range of gamified experiences across different layers, the framework ensures a holistic and engaging learning experience that appeals to learners of all levels and backgrounds. One key feature of the framework is its utilization of game mechanics to enhance learner engagement and motivation. Elements such as points, badges, levels, and leaderboards are strategically integrated into the learning process, providing learners with clear goals, feedback mechanisms, and a sense of progression. Through the gamification of learning activities, learners are incentivized to actively participate, explore, and excel in STEM subjects. Furthermore, the framework incorporates narrative storytelling elements to contextualize learning content and create immersive learning experiences. By weaving compelling narratives into STEM lessons, learners are transported into engaging scenarios that challenge their problem-solving skills, critical thinking abilities, and creativity. These narratives not only make learning more enjoyable but also help learners develop a deeper understanding of real-world STEM applications and implications. In addition to individual gameplay, the framework promotes social interactions and collaboration among learners through multiplayer gaming features. Learners have the opportunity to collaborate with peers, solve problems together, and compete in friendly challenges, fostering a sense of community and teamwork. By integrating social learning experiences into the framework, learners not only benefit from shared knowledge and support but also develop important interpersonal skills essential for success in STEM fields.

Moreover, the framework provides personalized feedback and adaptive learning pathways to support learners at their own pace and level of proficiency. Through intelligent analytics and assessment mechanisms, the framework dynamically adjusts learning content and challenges based on individual learner performance, ensuring a tailored and effective learning experience for each student. Ultimately, the innovative multi-layer gamification framework represents a ground-breaking approach to STEM education, offering a comprehensive solution to the challenges of engaging and motivating learners in these critical fields. By combining gamification strategies with narrative storytelling, social interactions, and adaptive learning features, the framework aims to inspire a new generation of STEM enthusiasts and equip them with the skills and knowledge needed to thrive in the rapidly evolving world of science and technology.

7. “Gamification and Computer Science Students’ Activity” (2020).

Authors: Miguel García-Iruela, Manuel J. Fonseca, Raquel Hijón-Neira, And Teresa Chambel

Description: Gamification has emerged as a potent tool in the realm of computer science education, offering a dynamic approach to engage students and enhance their learning experiences. By integrating gamified elements into computer science courses, educators can transform traditionally abstract and complex concepts into interactive and accessible learning opportunities. One key aspect of gamification in computer science education is the incorporation of game mechanics, such as points, levels, badges, and leaderboards, into learning activities. These mechanics provide clear goals, feedback mechanisms, and a sense of progression, motivating students to actively participate and excel in their studies. For example, students may earn points for completing coding challenges, unlock new levels as they master programming skills, and compete for top positions on leaderboards, fostering healthy competition and a sense of achievement. Moreover, gamification enables educators to create immersive learning experiences by contextualizing computer science concepts within engaging narratives or scenarios. For instance, students may embark on virtual quests or missions where they must apply programming principles to solve problems and overcome challenges, reinforcing their understanding of key concepts in a fun and memorable way. In addition to individual gameplay, gamification encourages collaboration and teamwork among students through multiplayer activities and social learning features.

Overall, gamification holds immense potential to revolutionize computer science education by making learning more engaging, interactive, and accessible to students of all backgrounds and abilities. By harnessing the motivational power of gamified elements, educators can inspire a new generation of computer scientists and equip them with the skills and knowledge needed to thrive in an increasingly digital world.

8. “Assessing students’ SQL knowledge and skills in gamification manner” (2020).

Authors: G. Tuparov and D. Keremedchiev

Description: The paper discusses a method for assessing students' SQL knowledge and skills in a gamification manner to enhance learning and motivation. By incorporating game elements such as points, badges, and leaderboards, the assessment process becomes more engaging and enjoyable for students, encouraging them to actively participate in their learning journey. The approach uses interactive SQL challenges and exercises to test students' understanding of key concepts such as querying, data manipulation, and database management. These challenges are designed to gradually increase in complexity, allowing students to apply their knowledge in real-world scenarios and gain hands-on experience with SQL.

Gamification elements such as rewards, achievements, and progress tracking are integrated into the assessment process. Students earn points and badges for completing challenges and demonstrating proficiency in SQL. Leaderboards foster a sense of healthy competition, motivating students to improve their performance and strive for higher rankings. The method also includes personalized learning paths tailored to individual students' skill levels and progress. This adaptive approach ensures that students receive tasks suited to their abilities, promoting continuous learning and improvement. Immediate feedback is provided for each challenge, helping students understand their mistakes and learn how to correct them. The paper highlights the benefits of using gamification in assessing SQL knowledge and skills, including increased student engagement, motivation, and knowledge retention. By offering a dynamic and interactive learning environment, the method helps demystify SQL and encourages students to explore its applications in various contexts.

9. “Experimentation of Gamification for Health and Fitness Mobile Application” (2019).

Authors: I. K. Buntoro and R. Kosala

Description: The experimentation of gamification within a health and fitness mobile application presents a novel approach to motivating users towards healthier lifestyles. By infusing elements of game design into the app's interface and functionalities, developers aim to engage users more effectively, foster adherence to fitness goals, and ultimately promote long-term behavior change. The gamified health and fitness application may incorporate various features to enhance user engagement and motivation. Firstly, it could utilize game mechanics such as points, achievements, levels, and rewards to incentivize users to stay active and achieve their fitness objectives. For example, users may earn points for completing workouts, unlock badges for reaching milestones, and level up as they progress on their fitness journey. Rewards could range from virtual badges and trophies to real-world incentives like discounts on fitness gear or access to premium features within the app. Moreover, the application could integrate social elements to facilitate peer support, friendly competition, and accountability. Users might be able to connect with friends, join fitness challenges or groups, and share their progress and achievements on social media platforms. By fostering a sense of community and camaraderie, the app encourages users to stay motivated and committed to their health and fitness goals. Additionally, the gamified health and fitness app could incorporate immersive experiences and interactive features to make workouts more engaging and enjoyable. For instance, users may participate in virtual reality (VR) fitness experiences, where they embark on virtual adventures or compete in virtual races while exercising.

In conclusion, the experimentation of gamification within a health and fitness mobile application holds promise for promoting physical activity, improving health outcomes, and enhancing user engagement and satisfaction. By integrating game design principles, social features, immersive experiences, and personalized recommendations, developers can create a compelling and effective tool for empowering users to lead healthier, more active lifestyles.

10. “Modeling Knowledge Assessment with Gamification Technology on E-Learning Platform” (2019).

Authors: E.V.Karmanova, E.V.Chernova, A.S. Dokolin

Description: The paper examines the integration of gamification technology in knowledge assessment on e-learning platforms to enhance the educational experience and improve learning outcomes. The study focuses on how game design principles can be combined with assessment strategies to create interactive and motivating environments that cater to a diverse range of learning needs. The approach uses game elements such as points, badges, and leaderboards to engage students and encourage them to participate actively in their learning journey. This gamification of assessments adds a layer of competition and achievement that can boost students' motivation to succeed. As students’ progress through the learning materials, they receive immediate, personalized feedback on their performance. This real-time feedback helps students understand their mistakes and learn how to improve their skills.

The paper also discusses the benefits of real-time feedback, which helps students quickly identify areas for improvement and adjust their learning strategies accordingly. Personalized challenges keep students engaged by providing tasks that match their skill level and push them toward achieving higher levels of understanding. In addition, the paper emphasizes how gamification can foster healthy competition among students. Leaderboards and achievement systems can create a sense of community and shared goals, as students strive to outperform their peers and achieve recognition for their accomplishments. This sense of achievement can drive students to engage more deeply with the material and aim for higher performance. The paper concludes with insights and recommendations for future research and implementation strategies in gamified assessment.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system features a game-like environment where learners answer questions, promoting engagement and knowledge through a competitive format. Users access learning materials, complete quizzes to assess understanding, and receive instant feedback. The system highlights content absorption without authentication.

3.1.1 DRAWBACKS OF THE EXISTING SYSTEM

1. Without authentication, there's a risk of users misrepresenting themselves or accessing content inappropriately. Authentication can help mitigate this risk by ensuring that only authorized users have access to the system's features and content.
2. It's difficult to track users' progress accurately over time. Progress tracking is essential for monitoring learning goals, identifying areas for improvement, and providing targeted feedback and support.

3.2 PROPOSED SYSTEM

The proposed system is a web application designed to provide users with an interactive and engaging platform for quiz participation and self-assessment. The system includes user authentication, enabling users to register, log in, and maintain personal profiles where they can track their scores and quiz history. It features a variety of multiple-choice quizzes on different topics, allowing users to test their knowledge and skills. The application leverages modern web technologies for its front-end and back-end, ensuring a smooth and responsive user experience. Users can access the system through a user-friendly interface that provides easy navigation and a seamless quiz-taking experience. This proposed system is well-suited for educational institutions, training programs, or general knowledge enhancement.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

1. The web application provides users with a seamless authentication process, including registration, login, and password recovery, along with quiz participation options such as choosing from a variety of timed quizzes and receiving immediate results and feedback.
2. By offering multiple-choice questions with randomized answer options, the application ensures users have varied and engaging quiz experiences, while a diverse pool of questions covers different topics and difficulty levels.
3. Administrators benefit from robust content management tools, allowing them to create, edit, and delete quizzes and questions, as well as manage user information, roles, and permissions. Additionally, detailed reports and analytics enable informed decision-making for system improvement and enhanced user experiences.

3.3 SYSTEM REQUIREMENTS

3.3.1 HARDWARE REQUIREMENTS

Hardware is the collection of physical elements that constitutes a computer system. Computer hardware refers to the physical parts or components of a computersuch as monitor, keyboard, computer data storage etc.

• Processor	: Intel i3 or above
• HDD	: 500 GB
• RAM	: 8 GB
• Processor Speed	: 2.5 GHz
• Display Type	: VGA
• Operating System	: Windows 8 or above
• Coding Language	: Python

3.3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide abasis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- SQL Lite
- Python 3.7.6
- HTML/CSS/JavaScript

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECUTURE

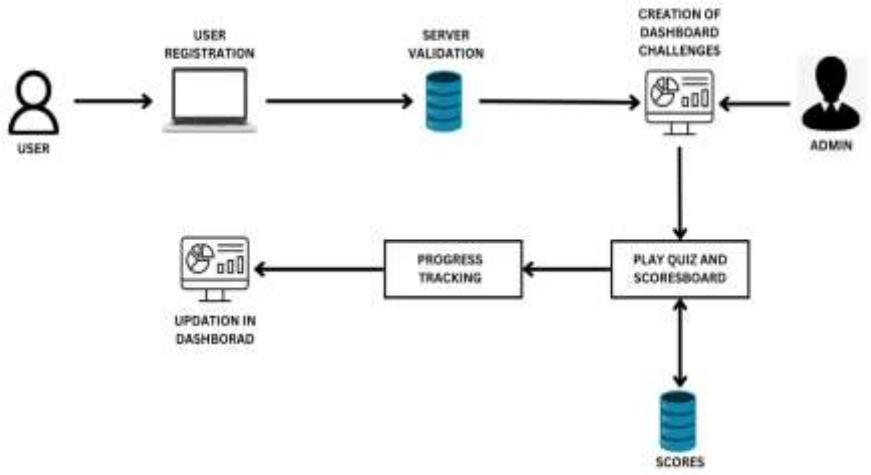


Figure 4.1 System Architecture

4.1.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation used in software engineering to depict the flow of data within a system. It consists of various symbols representing processes, data stores, data flows, and external entities. Processes represent activities or transformations that manipulate data, while data stores represent repositories where data is stored. Data flows represent the movement of data between processes, data stores, and external entities. External entities are sources or destinations of data outside the system boundary. DFDs provide a clear and concise visualization of how data moves through a system, aiding in system analysis, design, and communication among stakeholders.

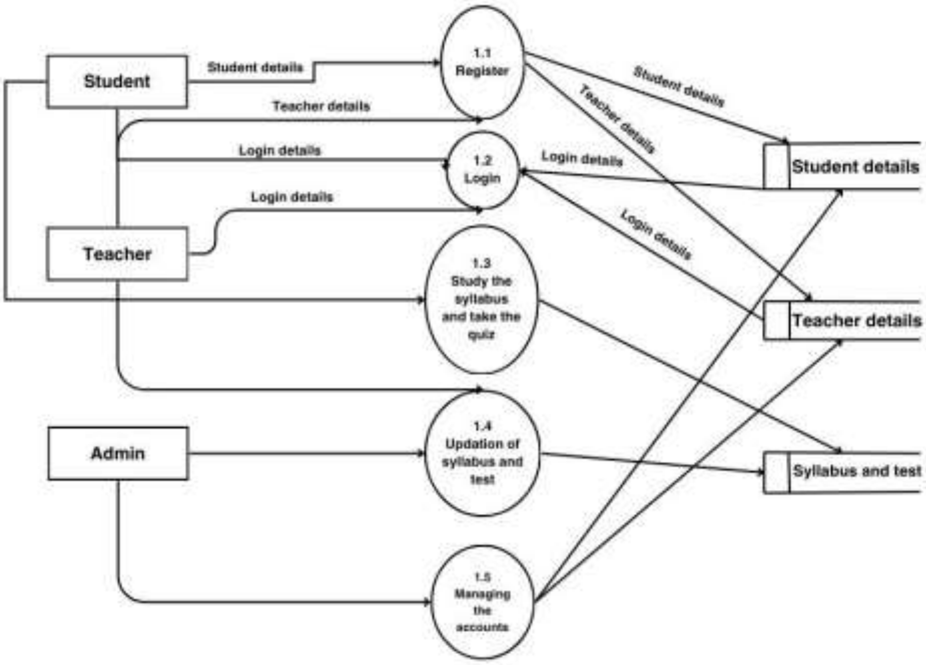


Figure 4.2 Data Flow Diagram

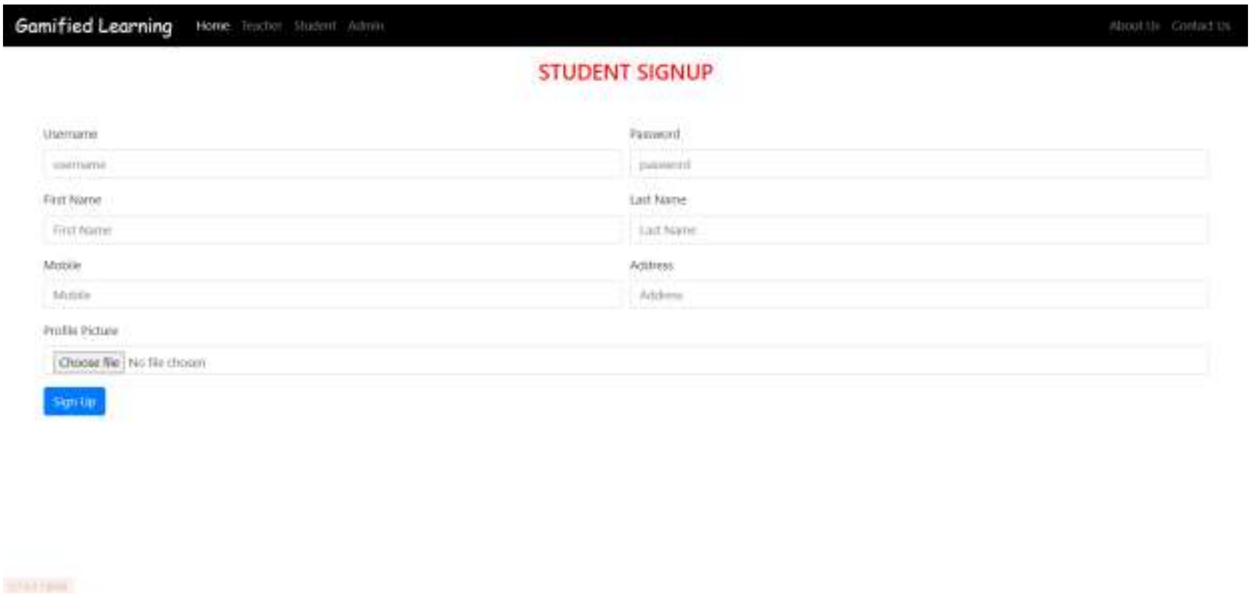
4.2 MODULE DESCRIPTION

The modules in this system are,

1. Authentication Module
2. Quiz Management Module
3. User Profile Module
4. Admin Module
5. Database Module

4.2.1 Authentication Module

The Authentication Module is the linchpin of security within the web application, responsible for managing user registration, login, password recovery, and session maintenance. It ensures a seamless and secure user experience by facilitating new user registration with email verification, validating login credentials, and enabling password recovery through secure, time-limited links. Employing robust password hashing and encryption techniques, it safeguards user passwords and sensitive data, while optionally supporting multi-factor authentication for enhanced security. Role-based access control regulates user permissions, while logging and monitoring functionalities track authentication activities for threat detection and mitigation. Overall, the Authentication Module not only safeguards user data but also establishes the trustworthiness and reliability of the application.



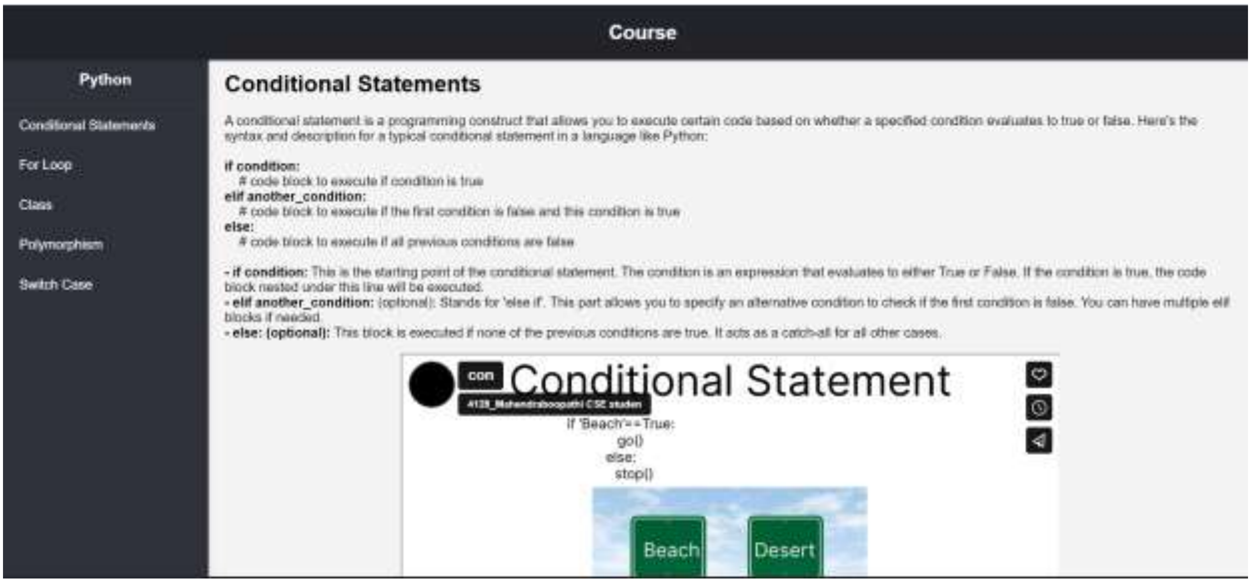
STUDENT SIGNUP

Username: username
Password: password
First Name: First Name
Last Name: Last Name
Mobile: Mobile
Address: Address
Profile Picture: Choose file No file chosen

Figure 4.3 Signup Page

4.2.2 Quiz Management Module

The Quiz Management Module is pivotal, guiding quizzes from inception to execution in the web application. Administrators craft quizzes with customized details, including titles, topics, and parameters like time limits. They curate a diverse question bank, enabling editing, deletion, and enrichment. Quiz categorization by topics aids user discovery, while question pooling minimizes redundancy. Randomization enhances fairness, averting potential cheating. The module allows for quiz scheduling, ideal for planned events. Users engage in quizzes aligned with their interests, with the module orchestrating sessions and providing post-quiz feedback. Integrated analytics empower users to track progress, while administrators refine quizzes based on insights. The module’s adaptability enriches user experiences, ensuring quality and appeal.



Conditional Statements

A conditional statement is a programming construct that allows you to execute certain code based on whether a specified condition evaluates to true or false. Here's the syntax and description for a typical conditional statement in a language like Python:

```
if condition:
   # code block to execute if condition is true
elif another_condition:
   # code block to execute if the first condition is false and this condition is true
else:
   # code block to execute if all previous conditions are false
```

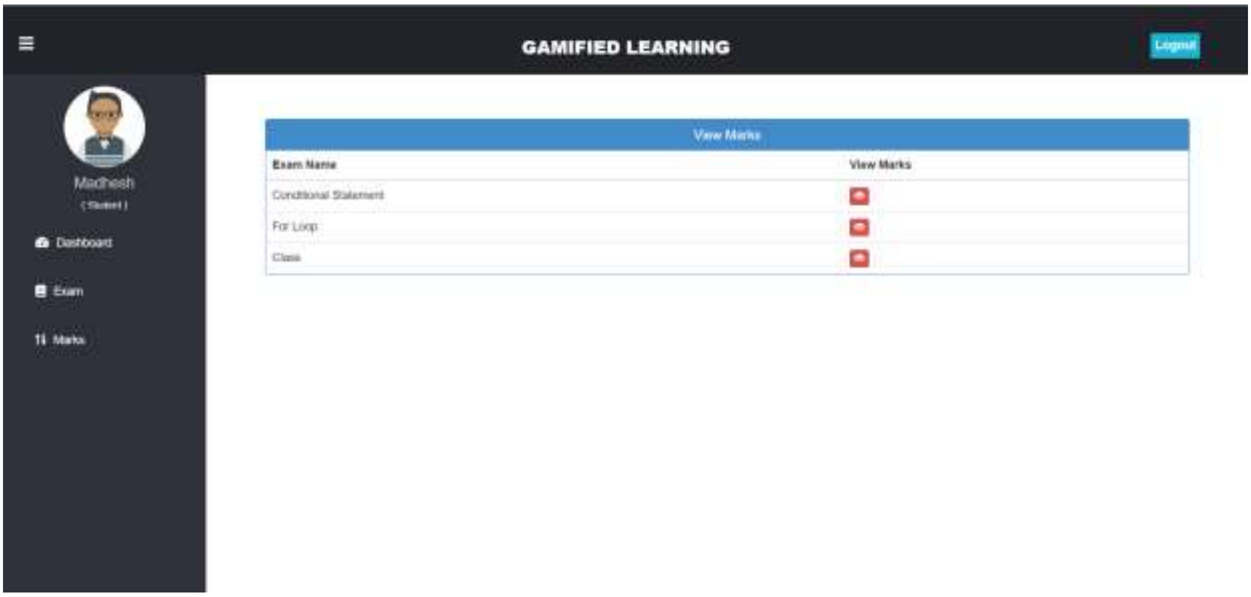
- **if condition:** This is the starting point of the conditional statement. The condition is an expression that evaluates to either True or False. If the condition is true, the code block nested under this line will be executed.
- **elif another_condition:** (optional): Stands for 'else if'. This part allows you to specify an alternative condition to check if the first condition is false. You can have multiple elif blocks if needed.
- **else: (optional):** This block is executed if none of the previous conditions are true. It acts as a catch-all for all other cases.

```
If 'Beach'==True:
      go()
   else:
     stop()
```

Figure 4.4 Course Page

4.2.3 User Profile Module

The User Profile Module is central to the web application, managing user interactions and data to deliver personalized experiences. Users can securely create and update profiles, tailoring details like name and preferences. Performance tracking tools allow users to monitor quiz history, fostering improvement. A repository of past quizzes aids reflection, while achievements incentivize engagement. Customization extends to settings like notifications and language preferences. Security measures protect user data through features like password management. The module fosters community engagement with leaderboards and feedback channels. In summary, the User Profile Module offers a personalized and secure environment, empowering users to optimize their quiz experience effectively.



View Marks

Exam Name	View Marks
Conditional Statement	👁
For Loop	👁
Class	👁

Figure 4.5 User Profile Page

4.2.4 Admin Module

The Admin Module within the web application empowers administrators with robust tools for overseeing quizzes, questions, users, and other vital aspects of the platform, ensuring its quality and integrity. Administrators wield its capabilities to create, modify, and remove quizzes, tailoring details such as titles, descriptions, topics, and difficulty levels. They can also schedule quizzes for specific occasions, facilitating live or timed engagements. Overseeing question management, administrators curate and refine question pools, including text, options, and correct answers, with options for categorization and randomization to uphold quiz fairness. User management functionalities enable administrators to oversee user accounts, adjusting profiles and permissions as needed. Comprehensive reporting and analytics provide insights into quiz performance, user engagement, and activity, facilitating informed decision-making for platform enhancements. Additionally, the module supports video uploading, enriching quiz content with multimedia resources. As the cornerstone of administration, the Admin Module ensures the seamless operation and optimization of the platform for all users.

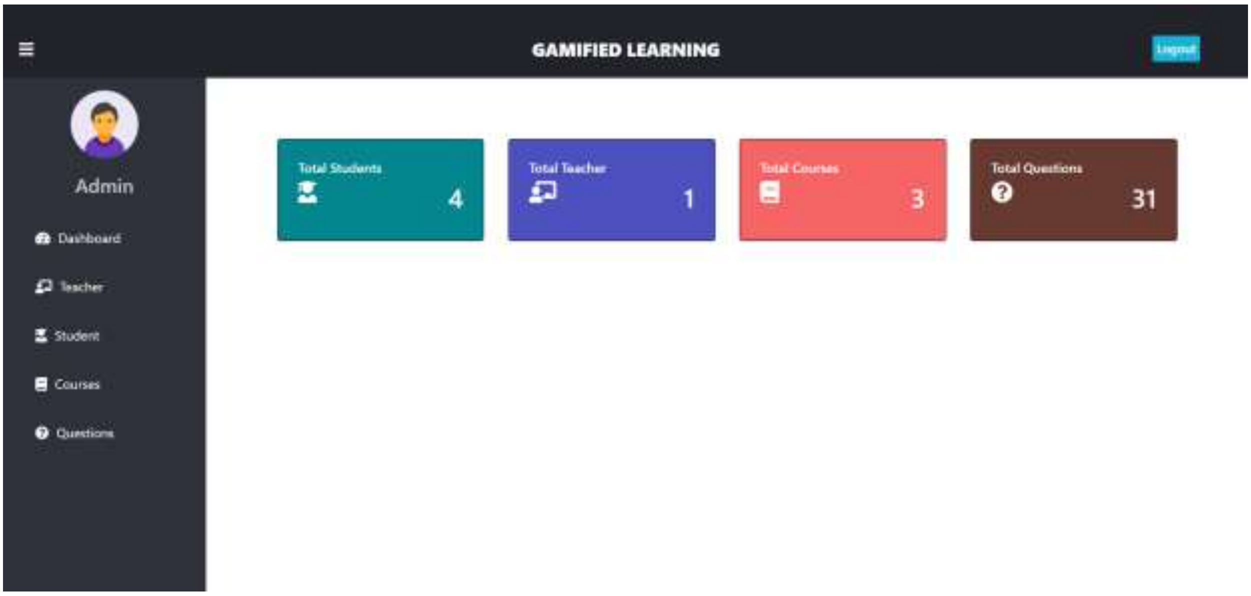


Figure 4.6 Admin Page

4.2.5 Database Module

The Database Module within the web application assumes the critical responsibility of securely housing and organizing data pertinent to quizzes, questions, and user profiles. It operates as the backbone for efficient data storage, ensuring the structured arrangement of information related to quizzes, questions, users, and quiz results within a relational database system. Upholding data integrity is paramount, achieved through meticulous practices like normalization, indexing, and adherence to referential integrity constraints. Robust security measures safeguard data against unauthorized access or breaches, encompassing encryption protocols, stringent access control mechanisms, and compliance with relevant privacy regulations. In essence, the Database Module forms the bedrock of the application's functionality, guaranteeing the reliability, security, and scalability of stored data.

4.3 APPLICATION ARCHITECTURE

4.3.1 ER Diagram

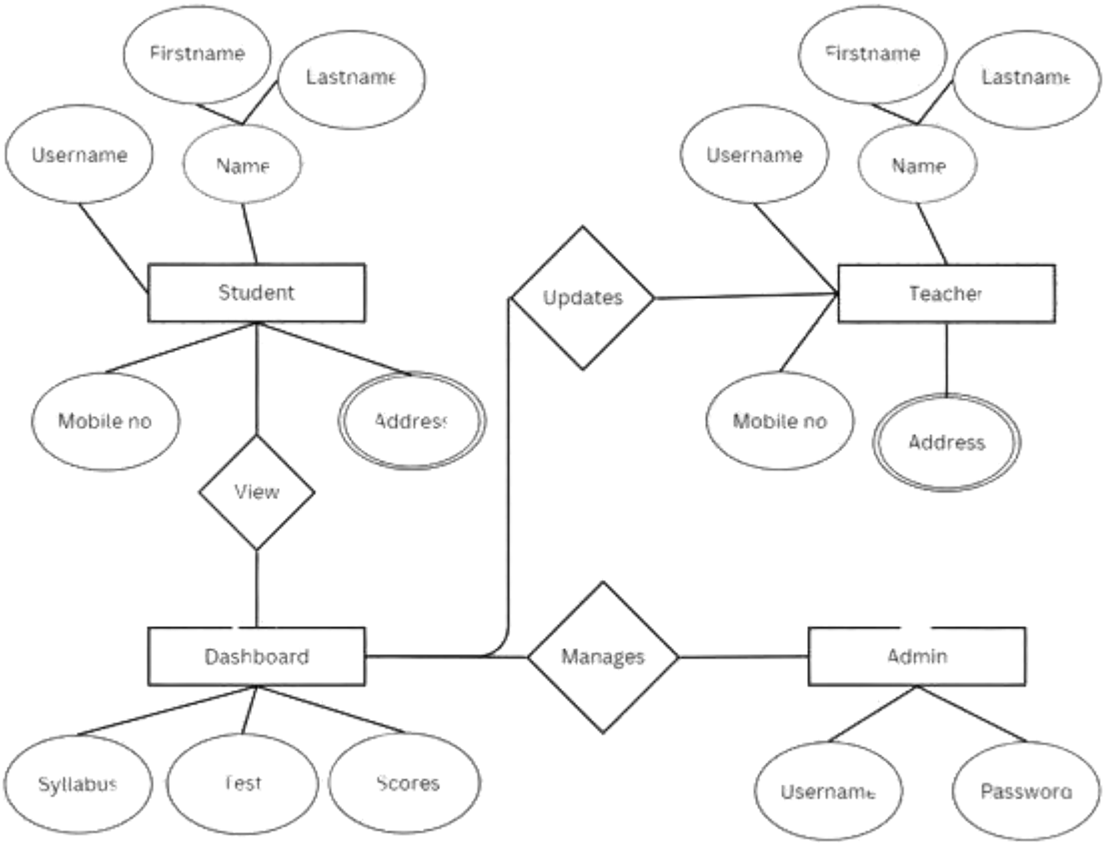


Figure 4.7 ER Diagram

4.3.2 Usecase Diagram

A use case diagram is a graphic depiction of the interactions among the elements of asystem. The boundary, which defines the system of interest in relation to the world around it. The actors, usually individuals involved with the system defined according to their roles. The use cases, which the specific roles are played by the actors within andaround the system.

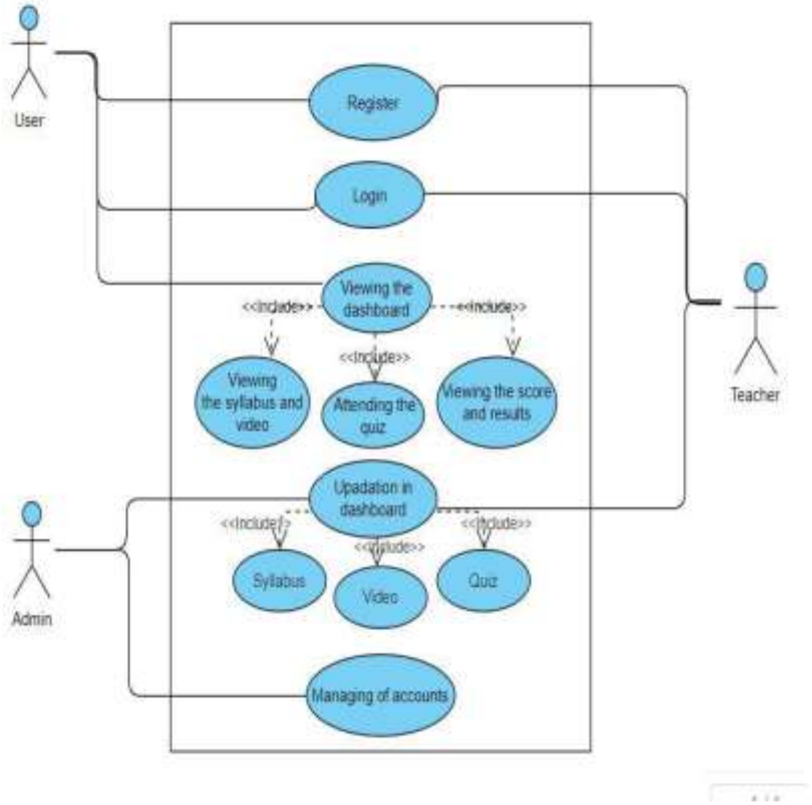


Figure 4.4 Use Case Diagram

4.3.3 Class Diagram

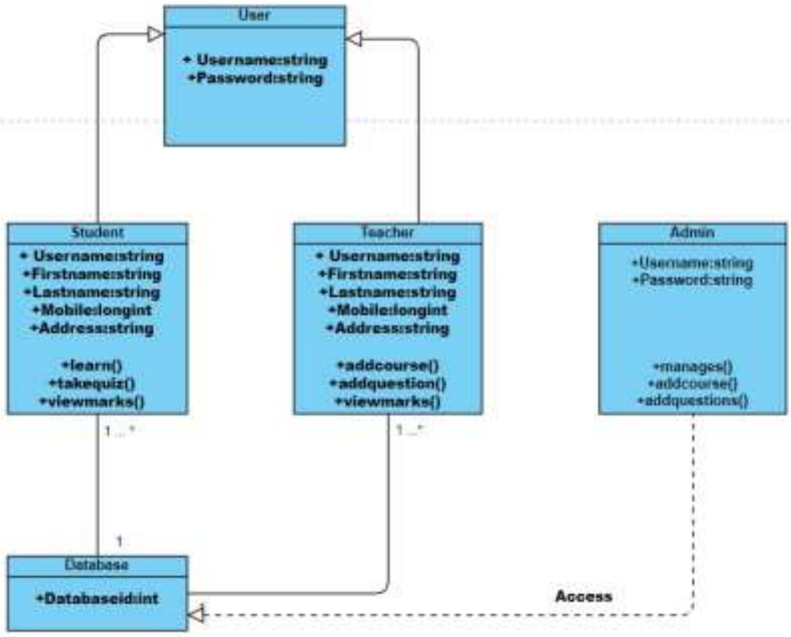


Figure 4.5 Class Diagram

4.3.4 Activity Diagram

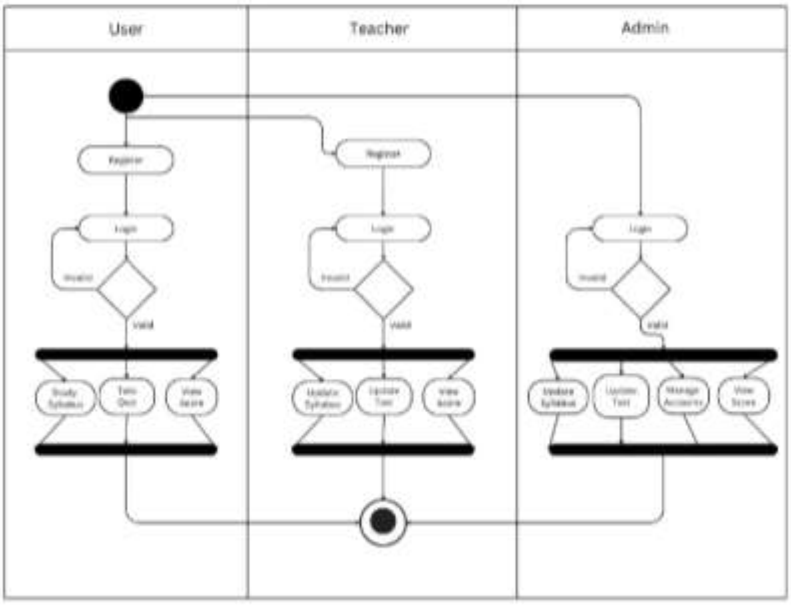


Figure 4.6 Activity Diagram

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 ALGORITHM

TOPIC – SPECIFIC ALGORITHM (TSA)

A topic-specific algorithm is an algorithm that is tailored to a particular subject matter or topic in order to optimize performance or evaluation. In the context of an online quiz or assessment platform, a topic-specific algorithm could refer to an approach that focuses on designing scoring, grading, or evaluation methods specific to the content and learning objectives of a particular topic or subject area.

Features and Components:

Content-Aware Scoring: Algorithm assigns scores based on question complexity and importance.

Difficulty Weighting: Questions of varying difficulty are weighted differently.

Topic Relevance: Algorithm ensures questions align with learning objectives.

Question Type Consideration: Different question types are weighted based on contribution.

Balanced Coverage: Subtopics within the main topic are equally covered.

Adaptive Techniques: Algorithm adjusts question difficulty based on user performance.

Performance Analysis: User's topic-specific performance is analysed for feedback.

Continuous Improvement: Algorithm is refined based on data and feedback for better alignment.

Topic-specific algorithms can enhance the accuracy, fairness, and educational value of online quizzes and assessments by focusing on the unique characteristics and learning goals of each specific topic.

How Topic – Specific Algorithm Works?

A topic-specific algorithm works by tailoring the evaluation and scoring process of a quiz or assessment to the nuances and characteristics of a particular subject or topic. This ensures a more precise and meaningful assessment of a user's understanding and mastery of the topic.

Steps How a Topic-Specific Algorithm Typically Works:

Topic Identification: Algorithm discerns the quiz's specific subject area using question metadata or quiz structure.

Question Analysis: Each question is scrutinized for alignment with learning objectives, potentially categorized by subtopics or skills.

Difficulty Weighting: Questions receive weights based on difficulty, with tougher questions earning more points.

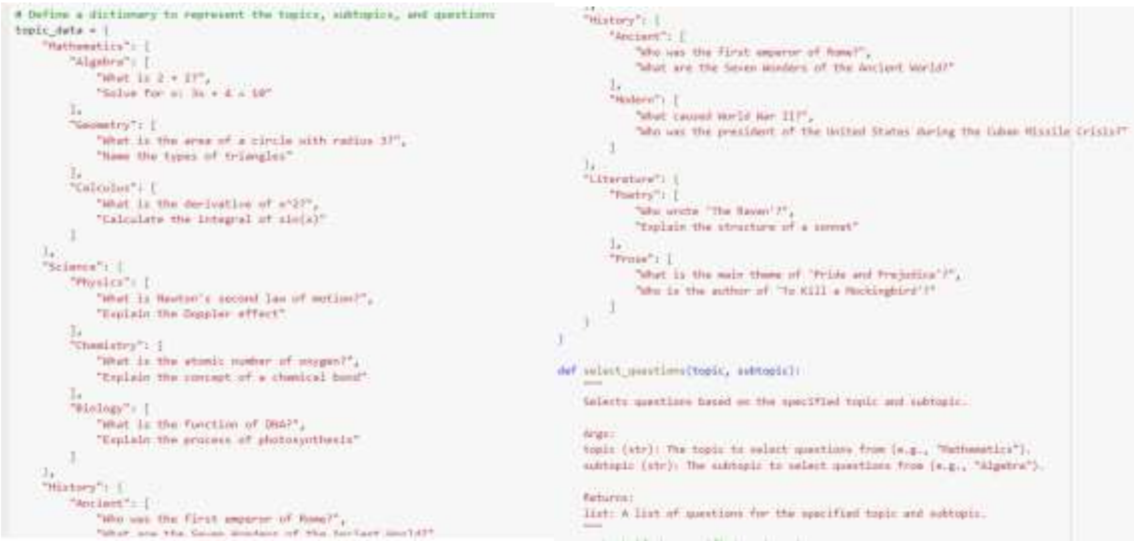
Topic Coverage: Algorithm ensures balanced representation of main and subtopics to avoid overemphasis or underrepresentation.

Performance Evaluation: User performance is assessed in real time, tracking correct and incorrect answers. Scoring: User's score is calculated based on weighted question values, considering each question's importance.

Feedback Generation: Targeted feedback is generated based on user performance, aiding comprehension and improvement.

Performance Analysis: Algorithm analyzes overall user performance to identify trends and areas needing attention.

Continuous Improvement: Algorithm evolves through data-driven refinements, updating weights, difficulty, and feedback**.**



```
# Define a dictionary to represent the topics, subtopics, and questions
topic_data = {
    "Mathematics": {
        "Algebra": [
            "What is 2 + 2?",
            "Solve for x: 3x + 4 = 10"
        ],
        "Geometry": [
            "What is the area of a circle with radius 3?",
            "Name the types of triangles"
        ],
        "Calculus": [
            "What is the derivative of x^2?",
            "Calculate the integral of sin(x)"
        ]
    },
    "Science": {
        "Physics": [
            "What is Newton's second law of motion?",
            "Explain the Doppler effect"
        ],
        "Chemistry": [
            "What is the atomic number of oxygen?",
            "Explain the concept of a chemical bond"
        ],
        "Biology": [
            "What is the function of DNA?",
            "Explain the process of photosynthesis"
        ]
    },
    "History": {
        "Ancient": [
            "Who was the first emperor of Rome?",
            "What are the Seven Wonders of the Ancient World?"
```

```
    },
    "History": {
        "Ancient": [
            "Who was the first emperor of Rome?",
            "What are the Seven Wonders of the Ancient World?"
        ],
        "Modern": [
            "What caused World War II?",
            "Who was the president of the United States during the Cuban Missile Crisis?"
        ]
    },
    "Literature": {
        "Poetry": [
            "Who wrote 'The Raven'?",
            "Explain the structure of a sonnet"
        ],
        "Prose": [
            "What is the main theme of 'Pride and Prejudice'?",
            "Who is the author of 'To Kill a Mockingbird'?"
        ]
    }
}

def select_questions(topic, subtopic):
    """
    Selects questions based on the specified topic and subtopic.

    Args:
    topic (str): The topic to select questions from (e.g., "Mathematics").
    subtopic (str): The subtopic to select questions from (e.g., "Algebra").

    Returns:
    list: A list of questions for the specified topic and subtopic.
    """
```

Figure. 5.1 Working of Algorithm (TSA)

Overall, a topic-specific algorithm aims to provide a fair, accurate, and informative assessment of a user's understanding of a particular topic, helping to enhance the learning experience and guide future study efforts.

Accuracy of Topic – Specific Algorithm:

The accuracy of a topic-specific algorithm can be evaluated in different ways depending on the goals and objectives of the algorithm. Typically, the algorithm's accuracy refers to how well it selects questions that match the desired topic or subtopic and provide a balanced and representative assessment of the topic's content. Here's how the accuracy of a topic-specific algorithm can be assessed:

Content Relevance: Algorithm's accuracy evaluated based on question alignment with topic's key concepts and knowledge areas.

Topic Coverage: Accurate algorithm ensures comprehensive coverage of topic, avoiding biases and spanning breadth and depth.

Difficulty Balance: Algorithm selects questions of varying difficulty levels to assess different knowledge levels accurately.

Question Quality: Accuracy depends on selecting high-quality questions that are clear, unambiguous, and error-free.

User Performance: Algorithm's accuracy assessed by tracking users' performance and differentiation between understanding levels.

Feedback and Review: Insights from user and educator feedback provide evaluation of algorithm's accuracy in question selection.

Statistical Metrics: Precision, recall, and F1-score metrics used to assess accuracy in identifying and selecting topic-related questions.

To improve the accuracy of a topic-specific algorithm, continuous monitoring and adjustment based on feedback, performance data, and evolving topics and questions are necessary. By doing so, the algorithm can better cater to the needs and expectations of users and provide more reliable assessments.

METRICS	PERCENTAGE BREAKDOWN
Content Relevance	30%
Topic Coverage	20%
Difficulty Balance	15%
Question Quality	15%
User Performance	10%
Feedback and Review	5%
Statistical Metrics	5%

Table 5.1 Table of TSA

Real-World Applications of TSA

Some of the common real-world applications Topic – Specific Algorithm are given below:

E-learning Platforms: Topic-specific algorithms personalize learning by selecting questions or content aligned with students' objectives or interests.

Adaptive Testing: Algorithms dynamically adjust question difficulty and content based on real-time performance, enhancing assessment precision.

Content Recommendation: Algorithms recommend articles or media based on user interest in specific topics, providing a personalized experience.

By using topic-specific algorithms in real-time applications, organizations and systems can provide more personalized, efficient, and targeted experiences for users, improving engagement and outcomes across various domains.

SIMPLE SCORING ALGORITHM (SSA)

A simple scoring algorithm is a straightforward method for evaluating performance or achievement in various contexts, such as quizzes, assessments, or competitions. It involves assigning a set value or score to each correct response and summing these scores to calculate a total score.

Features and Components:

Fixed Point Value: Each correct answer is assigned a fixed-point value, often a single point or a predefined value. Incorrect answers typically receive no points.

Equal Weighting: All questions are usually weighted equally, regardless of their difficulty or type. This makes the scoring process simple and easy.

Total Score Calculation: The total score is calculated by summing the points for all correct answers. This provides a clear and straightforward metric of performance.

No Penalties for Incorrect Answers: In most simple scoring systems, there are no penalties for incorrect answers, which can encourage participants to attempt all questions.

Ease of Implementation: Simple scoring algorithms are easy to implement and require minimal computational resources, making them suitable for a wide range of applications.

Transparency: The simplicity of the algorithm makes it transparent and easy to understand for users, who can easily see how their performance is evaluated.

Simple scoring algorithms are commonly used in educational and competitive settings because of their ease of use and clear, direct approach to evaluating performance. However, they may not account for variations in question difficulty or complexity, so they may not provide a nuanced assessment of a participant's knowledge or abilities.

How Simple Scoring Algorithm Works?

A simple scoring algorithm works by assigning a fixed-point value to each correct answer in a quiz, assessment, or competition. This straightforward approach evaluates performance based on the sum of points earned for correct answers.

Steps How a Simple Scoring Algorithm Typically Works:

Question Analysis: Each question is treated equally, regardless of its type or difficulty. There is no differentiation in the point value assigned to different questions.

Fixed Point Value Assignment: Correct answers are awarded a fixed-point value, often one point per correct answer. Incorrect answers typically receive no points.

Total Score Calculation: The algorithm calculates the total score by summing the points earned from all correct answers. This simple sum provides the user's final score.

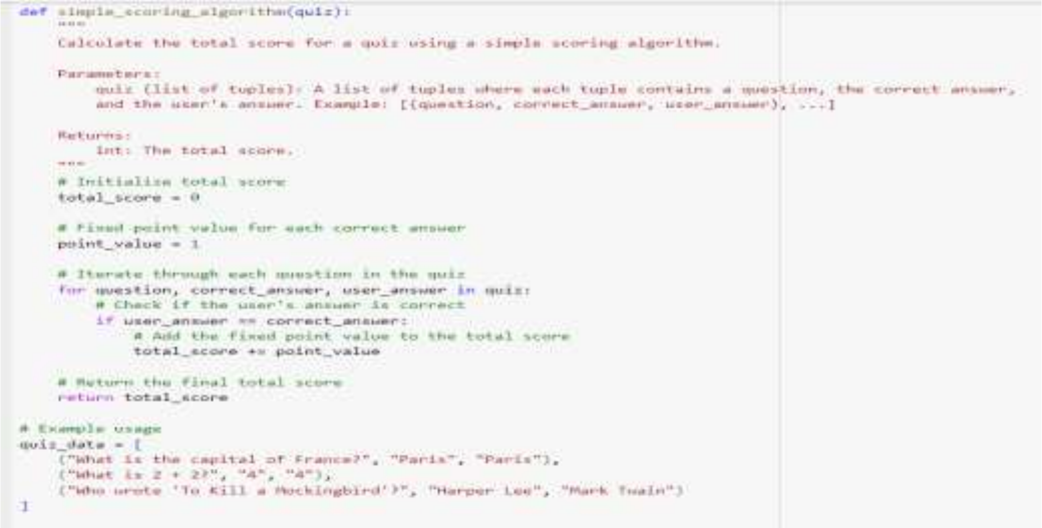
No Penalties for Incorrect Answers: Generally, there are no penalties for incorrect answers. This allows users to attempt all questions without the risk of losing points.

Straightforward Implementation: The algorithm is easy to implement and requires minimal computational resources, making it suitable for a wide range of applications.

Transparency: The algorithm's simplicity makes it easy for users to understand how their performance is evaluated and scored.

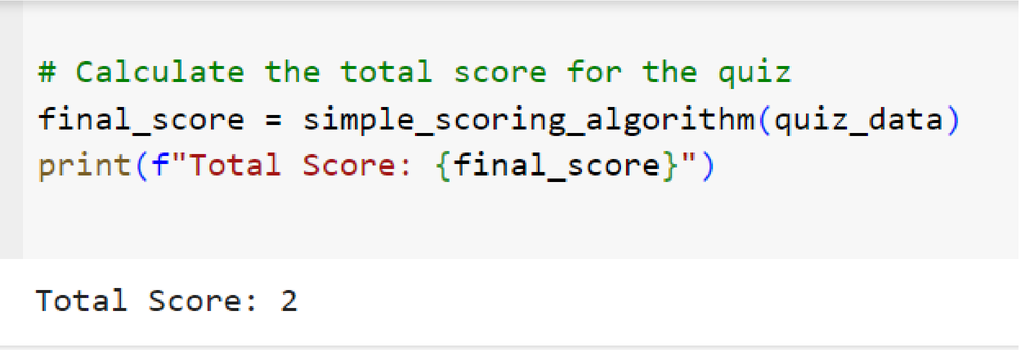
Quick Feedback: The algorithm allows for quick calculation and feedback of the final score, providing immediate insights into a user's performance.

Consistency: Since the algorithm uses a consistent point value for each question, scoring is predictable and consistent across different assessments.



```python
def simple_scoring_algorithm(quiz):
    """
    Calculate the total score for a quiz using a simple scoring algorithm.

    Parameters:
        quiz (list of tuples): A list of tuples where each tuple contains a question, the correct answer,
        and the user's answer. Example: [(question, correct_answer, user_answer), ...]

    Returns:
        int: The total score.
    """
    # Initialize total score
    total_score = 0

    # Fixed point value for each correct answer
    point_value = 1

    # Iterate through each question in the quiz
    for question, correct_answer, user_answer in quiz:
        # Check if the user's answer is correct
        if user_answer == correct_answer:
            # Add the fixed point value to the total score
            total_score += point_value

    # Return the final total score
    return total_score

# Example usage
quiz_data = [
    ("What is the capital of France?", "Paris", "Paris"),
    ("What is 2 + 2?", "4", "4"),
    ("Who wrote 'To Kill a Mockingbird'?", "Harper Lee", "Mark Twain")
]
```

Figure. 5.2 Working of Algorithm (SSA)



```
# Calculate the total score for the quiz
final_score = simple_scoring_algorithm(quiz_data)
print(f"Total Score: {final_score}")
```

```
Total Score: 2
```

Figure. 5.3 Algorithm Result (SSA)

Overall, a simple scoring algorithm is easy to use and understand, providing a clear and direct measure of performance. While it offers consistency and simplicity, it may not account for variations in question difficulty, limiting its ability to provide a nuanced assessment of a user's knowledge or abilities.

Accuracy of Simple Scoring Algorithm:

A simple scoring algorithm is a straightforward approach to evaluate user performance based on correct answers and their assigned scores. In the context of an online quiz or assessment platform, a simple scoring algorithm aims to provide a fair and efficient method of evaluating user knowledge and understanding.

Here's how the accuracy of a simple scoring algorithm can be assessed:

Correct Answer Recognition: Algorithm accurately identifies user's correct or incorrect answers, crucial for scoring accuracy.

Consistent Point Allocation: Points consistently awarded based on correct answers to ensure fair scoring.

Error-Free Implementation: Algorithm operates without errors to ensure consistent and accurate evaluation.

Question Quality: Clear, unambiguous questions enhance scoring accuracy by providing a reliable basis.

Performance Feedback: Algorithm may offer feedback on user performance, highlighting strengths and areas for improvement**.**

Simple scoring algorithms are effective in providing clear, quantifiable evaluations of user performance. By accurately recognizing correct answers, consistently assigning points, and ensuring error-free implementation, these algorithms can offer reliable assessments. Regular reviews and continuous improvements can further enhance the accuracy and effectiveness of the simple scoring algorithm.

METRICS	PERCENTAGE BREAKDOWN
Correct Answer Recognition	30%
Consistent Point Allocation	25%
Error-Free Implementation	20%
Question Quality	15%
Performance Feedback	10%

Table 5.2 Table of SSA

Real-World Applications of SSA:

Some of the common real-world applications of Simple Scoring Algorithm are given below:

Online Quizzes and Assessments: In educational settings, online quizzes and assessments often use simple scoring algorithms to evaluate students' understanding of the material. Points are awarded for correct answers, and students' scores provide a measure of their proficiency in the subject.

Customer Feedback and Surveys: Simple scoring algorithms can be used to evaluate customer responses to surveys or feedback forms. By assigning scores to different responses, businesses can quantify customer satisfaction and identify areas for enhancement.

Online Learning Platforms: Many online learning platforms use simple scoring algorithms to grade students' assignments and quizzes. This allows for quick, consistent evaluation of students' progress and provides feedback on their performance.

Simple scoring algorithms are a practical and efficient method of evaluating performance in various contexts. They are easy to implement and understand, making them a popular choice for real-world applications.

CRUD OPERATIONS ALGORITHM (CRUD):

CRUD operations refer to the four fundamental actions in software and database management: Create, Read, Update, and Delete. These operations form the basis for managing data in databases, applications, and software systems.

Here's an overview of each operation:

Create: The create operation involves adding new data or records to a database or system. This can include adding new users, entries, or information in an application.

Read: The read operation refers to retrieving or querying existing data from a database or system. It allows users or applications to access and display data for analysis or use.

Update: The update operation involves modifying existing data or records in a database or system. This can include changing a user's information or updating specific entries.

Delete: The delete operation is used to remove existing data or records from a database or system. It is a permanent removal of data and should be handled with care to avoid accidental loss of important information.

Features and Components:

Consistency: CRUD operations provide a consistent and reliable way to manage data in applications and databases.

Data Integrity: By using CRUD operations, data integrity can be maintained, ensuring that changes are made accurately and consistently.

Security and Permissions: Access control can be implemented with CRUD operations, allowing different users to have varying levels of permissions (e.g., read-only or full access).

Scalability: CRUD operations can be optimized to handle large amounts of data and support scalability as applications grow.

User Experience: Efficient and seamless CRUD operations contribute to a better user experience, enabling users to interact with applications easily.

CRUD operations are foundational to managing data in software systems and are widely used in web applications, mobile apps, enterprise systems, and more. By providing a standardized approach to data management, CRUD operations help developers maintain organized and efficient systems.

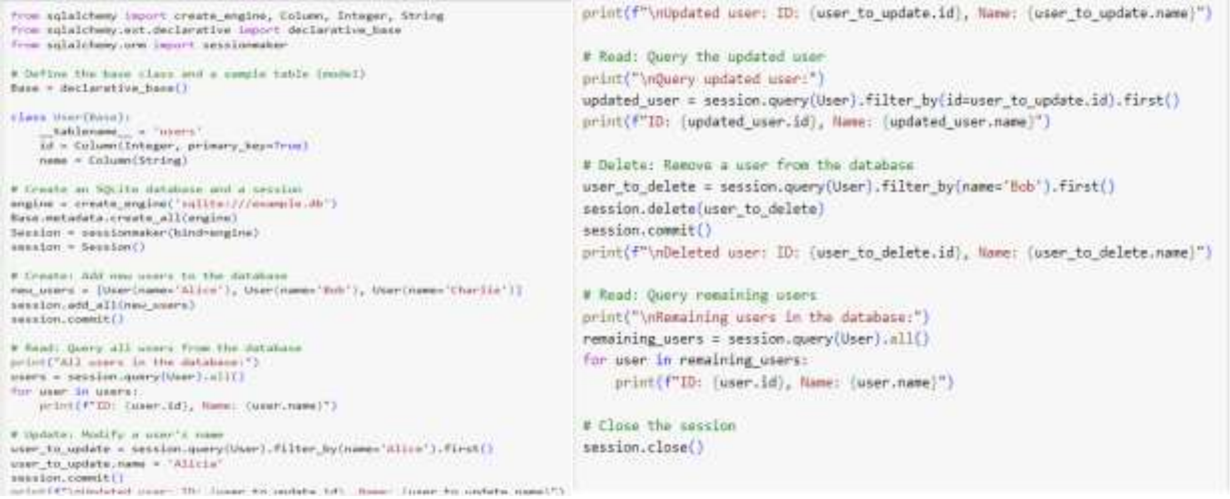
How CRUD Operations Works?

CRUD operations, standing for Create, Read, Update, and Delete, are fundamental operations used to manage data in databases and software systems.

Steps How a Simple Scoring Algorithm Typically Works:

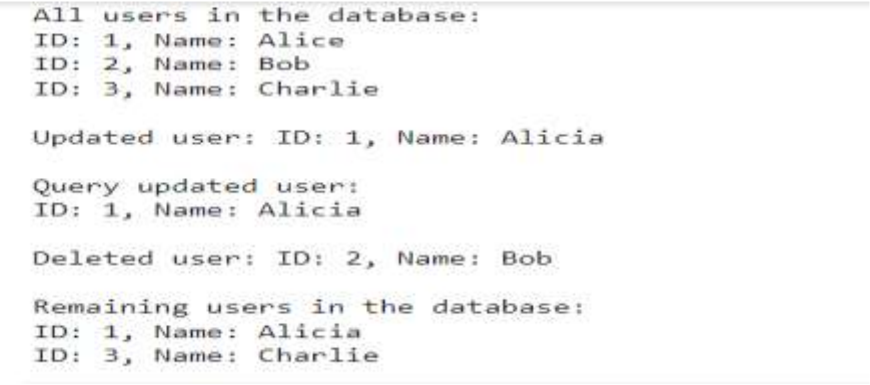
- **Create**
- **Read**
- **Update**
- **Delete**
- **Data Validation:** Ensures data meets quality standards and business rules before being created or updated.
- **Access Control:** Limits CRUD operations to authorized users to protect sensitive data and maintain data integrity.
- **Error Handling:** Handles any exceptions or errors that arise during CRUD operations, such as invalid data or permissions issues.

In practice, CRUD operations form the basis of data management in most software applications, enabling systematic and efficient handling of data throughout its lifecycle.



```
from sqlalchemy import create_engine, Column, Integer, String
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

# Define the base class and a sample table (model)
Base = declarative_base()

class User(Base):
    __tablename__ = 'users'
    id = Column(Integer, primary_key=True)
    name = Column(String)

# Create an SQLite database and a session
engine = create_engine('sqlite:///example.db')
Base.metadata.create_all(engine)
Session = sessionmaker(bind=engine)
session = Session()

# Create: Add new users to the database
new_users = [User(name='Alice'), User(name='Bob'), User(name='Charlie')]
session.add_all(new_users)
session.commit()

# Read: Query all users from the database
print("All users in the database:")
users = session.query(User).all()
for user in users:
    print(f"ID: {user.id}, Name: {user.name}")

# Update: Modify a user's name
user_to_update = session.query(User).filter_by(name='Alice').first()
user_to_update.name = 'Alicia'
session.commit()
print(f"\nUpdated user: ID: {user_to_update.id}, Name: {user_to_update.name}")

# Read: Query the updated user
print("\nQuery updated user:")
updated_user = session.query(User).filter_by(id=user_to_update.id).first()
print(f"ID: {updated_user.id}, Name: {updated_user.name}")

# Delete: Remove a user from the database
user_to_delete = session.query(User).filter_by(name='Bob').first()
session.delete(user_to_delete)
session.commit()
print(f"\nDeleted user: ID: {user_to_delete.id}, Name: {user_to_delete.name}")

# Read: Query remaining users
print("\nRemaining users in the database:")
remaining_users = session.query(User).all()
for user in remaining_users:
    print(f"ID: {user.id}, Name: {user.name}")

# Close the session
session.close()
```

Figure. 5.4 Working of Algorithm (CRUD)



```
All users in the database:
ID: 1, Name: Alice
ID: 2, Name: Bob
ID: 3, Name: Charlie

Updated user: ID: 1, Name: Alicia

Query updated user:
ID: 1, Name: Alicia

Deleted user: ID: 2, Name: Bob

Remaining users in the database:
ID: 1, Name: Alicia
ID: 3, Name: Charlie
```

Figure. 5.5 Algorithm Result (CRUD)

Accuracy of CRUD Algorithm:

CRUD (Create, Read, Update, Delete) operations refer to the four basic operations of persistent storage. In software applications, these operations are used to interact with databases and manage data efficiently.

Here is how the accuracy of CRUD operations can be assessed:

Data Validation: Ensure data validity before insertion and maintain data relationships across tables for accuracy.

Consistent Operations: Implement CRUD operations consistently throughout the application to maintain data integrity.

Error Handling: Employ robust error handling mechanisms to prevent data corruption and loss, ensuring reliability.

Security Measures: Implement access control to restrict CRUD operations to authorized users, safeguarding data accuracy.

Performance Monitoring: Monitor CRUD operation performance to identify and resolve inefficiencies, improving data management accuracy and efficiency.

By focusing on these aspects, a system can maintain a high level of accuracy in its CRUD operations, leading to better data management and a more reliable application. Regular reviews and updates can help maintain the accuracy and efficiency of the CRUD operations over time.

METRICS	PERCENTAGE BREAKDOWN
Data Integrity	30%
Consistent Operations	25%
Error Handling and Prevention	20%
Security and Access Control	15%
Performance Monitoring	10%

Table 5.3 Table of CRUD

Real-World Applications of CRUD

Some of the common real-world applications of CRUD Algorithm are given below:

Social Media Platforms: Users can create, read, update, and delete posts, comments, and user profiles.

Healthcare Management Systems: Healthcare applications use CRUD operations to manage patient records, appointments, and medical histories.

Library Management Systems: These systems manage book inventories, borrower records, and lending history.

CRUD operations are essential for any application that involves data management. By effectively utilizing these operations, applications can maintain data integrity and provide users with seamless data manipulation and retrieval experiences.

CSRF PROTECTION ALGORITHM (CSRF)

CSRF (Cross-Site Request Forgery) protection is a security measure designed to prevent unauthorized actions on a web application. CSRF attacks exploit the trust that a website has in a user's browser, allowing an attacker to trick the user into executing actions they didn't intend. The attack typically involves a malicious website causing the user's browser to send a request to a different, trusted site where the user is authenticated.

Features and Components:

Token-Based Validation: The algorithm generates a unique CSRF token for each session or request, which must be included in forms or AJAX requests. The server validates the token to ensure the request is legitimate.

Secure Cookies: Utilizing HTTP-only and secure cookies can prevent attackers from accessing session information, making CSRF attacks more difficult to execute.

Session Management: Proper session management, including invalidating sessions after certain conditions (e.g., inactivity, logout), can help reduce the risk of CSRF attacks.

Continuous Monitoring: Continuous monitoring of application traffic for unusual patterns or behaviors can help detect and respond to potential CSRF attacks.

CSRF protection is essential for securing web applications against malicious attacks that could compromise user data or perform unauthorized actions on behalf of the user. Proper implementation of CSRF protection mechanisms is crucial for safeguarding web applications.

How CRUD Operations Works?

CSRF (Cross-Site Request Forgery) protection aims to prevent unauthorized actions on behalf of authenticated users by ensuring that requests come from legitimate sources.

Steps How a Simple Scoring Algorithm Typically Works:

Token Generation: When a user initiates a session on the website, a unique CSRF token is generated by the server. This token is sent to the user's browser and stored in a secure cookie or as part of the session data.

Token Validation: When a request is made to the server, the CSRF token included in the request is validated against the token stored on the server. If the tokens match, the request is considered legitimate.

Secure Cookies: Using secure and HTTP-only cookies to store CSRF tokens prevents attackers from accessing them through client-side scripting.

Error Handling: In case of token mismatch or missing token, the server rejects the request, preventing unauthorized actions. Proper error handling provides a secure response to prevent information leakage.

Logging and Monitoring: Logging requests and monitoring traffic can help detect and respond to suspicious activities indicative of CSRF attacks.

In practice, CSRF protection helps secure web applications against attacks that could exploit the trust between a user's browser and the website. Implementing robust CSRF protection mechanisms is essential to maintain the integrity and security of sensitive operations and data.

Accuracy of CSRF Algorithm:

Here's an estimation of the accuracy of preventing Cross-Site Request Forgery (CSRF) attacks across different aspects:

Token Validation: The most critical aspect of CSRF protection is validating tokens to ensure that requests originate from legitimate sources. Proper token management ensures that requests without valid tokens are rejected, thereby preventing unauthorized actions.

Session Management: Proper session management plays a vital role in protecting against CSRF attacks. This includes tying tokens to user sessions and implementing secure session handling practices such as rotating session identifiers.

Secure Token Storage: Tokens should be securely stored and handled. This includes using HTTPS for token transmission and ensuring that tokens are not exposed in logs or URLs.

Testing and Monitoring: Regularly testing the application for CSRF vulnerabilities and monitoring for suspicious activity can help identify and mitigate potential threats.

These percentages are estimates and may vary depending on the specific application and its security requirements. Continuous monitoring, testing, and updates are essential for maintaining the accuracy and effectiveness of CSRF protection measures.

METRICS	PERCENTAGE BREAKDOWN
Token Validation	30%
Session Management	20%
Secure Token Storage	15%
Testing and Monitoring	5%

Table 5.4 Table of CSRF

Real-World Applications of CSRF

Some of the common real-world applications of CSRF Algorithm are given below:

Online Banking and Financial Services: CSRF protection is critical in online banking, as attackers can exploit CSRF vulnerabilities to initiate unauthorized transactions, such as transferring funds or changing account settings.

E-commerce and Online Retail: CSRF attacks can impact online shopping platforms by modifying user account information, placing orders, or changing payment methods without user consent.

Social Media Platforms: In social media, CSRF vulnerabilities can be exploited to post unauthorized content, send unwanted messages, or modify account settings.

Healthcare and Medical Records: CSRF attacks can pose a significant risk in healthcare applications, where unauthorized changes to patient records could have serious consequences.

To protect against CSRF attacks, these applications employ security measures such as CSRF tokens, secure session management, Same Site cookie attributes, and user education on suspicious activity. Implementing these measures can help maintain the integrity and security of web applications and the data they handle.

Advantages of TSA:

Customized Assessment: The algorithm tailors the quiz to focus on specific topics or subject areas, providing a more targeted and relevant assessment of the user's knowledge.

Improved Learning Outcomes: By focusing on key learning objectives and critical concepts, the algorithm can help reinforce the most important topics and improve educational outcomes.

Balanced Coverage: The algorithm ensures that all important subtopics within the main topic are covered, providing a comprehensive assessment of the user's knowledge.

Fair and Accurate Scoring: The algorithm can assign weights to questions based on their difficulty and importance, leading to more nuanced and fair scoring.

Targeted Feedback: The algorithm can generate specific feedback based on user performance on individual topics, helping users identify strengths and areas for improvement.

Consistency and Predictability: A well-designed topic-specific algorithm ensures consistent and predictable assessment outcomes, fostering trust in the quiz process.

Advantages of SSA

Ease of Implementation: Simple scoring algorithms are straightforward to design and implement, making them a practical choice for online quiz projects.

Transparency: The algorithm's simplicity makes it easy for users to understand how their performance is evaluated and scored.

Consistent Scoring: The algorithm uses consistent point values for each question, ensuring predictable and uniform scoring across different quizzes.

Quick Feedback: Simple scoring algorithms allow for quick calculation of final scores, providing immediate feedback to users about their performance.

Fairness: By assigning equal weight to each question, the algorithm treats all questions equally, which can be perceived as fair by quiz participants.

Encourages Participation: Without penalties for incorrect answers, users may be more willing to attempt all questions, increasing engagement and participation.

Scalability: Due to its simplicity, the algorithm can scale easily to accommodate quizzes of different sizes and complexities.

Advantages of CRUD

Data Management: CRUD operations facilitate comprehensive data management, enabling the creation, reading, updating, and deletion of quiz questions for effective content management.

Flexibility and Control: Developers have granular control over quiz data, allowing them to easily modify, add, or remove questions to tailor quiz content to specific needs.

User-Friendly Interface: CRUD operations provide an intuitive interface for quiz creators and administrators to efficiently manage quiz content, enhancing usability.

Real-Time Updates: CRUD operations enable instant updates to quiz content, ensuring that corrections or new questions are reflected in the quiz environment immediately.

Consistency and Accuracy: By regulating data manipulation, CRUD operations uphold data consistency and accuracy, ensuring reliable quiz content.

Security and Access Control: Proper access controls ensure that only authorized users can perform CRUD operations, safeguarding quiz data from unauthorized access or modification.

Improved Performance: Efficient CRUD operations contribute to optimized system performance, ensuring seamless data management and access for users.

Easy Maintenance: CRUD operations streamline maintenance tasks, simplifying the process of updating, deleting, or modifying quiz questions for long-term management.

Data Backup and Recovery: CRUD operations facilitate data backup and recovery procedures, ensuring the resilience of quiz data and minimizing the risk of data loss.

Advantages of CSRF

Prevents Unauthorized Actions: CSRF protection ensures that unauthorized requests cannot be made on behalf of a user without their knowledge, safeguarding the platform from malicious activities.

Enhances User Trust: Implementing CSRF protection showcases a commitment to user security, fostering trust and confidence in the platform.

Protects User Data: CSRF protection helps maintain the integrity of user data, ensuring that quiz responses and scores remain secure from unauthorized access.

Maintains Integrity of Results: By preventing unauthorized submissions, CSRF protection preserves the accuracy and reliability of quiz results, ensuring fair assessments.

Complies with Security Best Practices: Integrating CSRF protection aligns with industry standards, demonstrating a proactive approach to web security.

Prevents Manipulation: CSRF protection safeguards against potential manipulation of quiz answers or content, preserving the integrity of quizzes.

Supports Secure User Authentication: CSRF protection complements user authentication systems, reinforcing platform security against unauthorized access.

Increases Platform Reliability: By thwarting unauthorized actions, CSRF protection contributes to platform stability and reliability, promoting a consistent user experience.

Protects Against a Range of Attacks: CSRF protection forms part of a comprehensive security strategy, defending against various web application vulnerabilities for robust protection.

5.2 SYSTEM TESTING

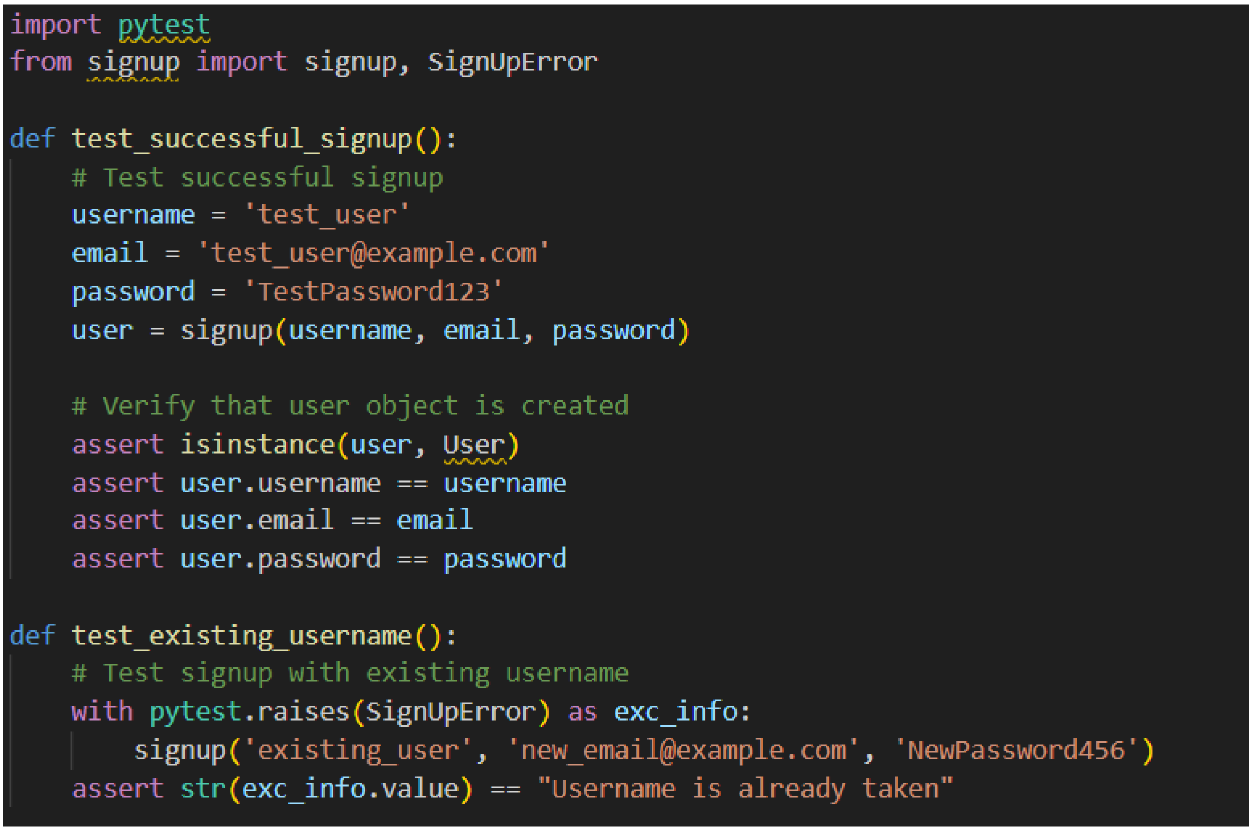
5.2.1 Unit Testing

The overall objective of these unit tests is to verify the core functionalities of the application related to user management, course creation, and question creation.

Each test case aims to ensure that the respective functionality works as expected and results in the creation of the corresponding object in the database when provided with valid input data.

1. **pytest.fixture** is used to define a fixture named **valid_user_details**, which provides valid user details as a dictionary.
2. The **test_user_creation** function is the actual test case. It uses the **valid_user_details** fixture as input.

3. Inside the test case, the **create_user** function is called with the provided valid user details. This function should be a part of your application's codebase and responsible for creating a user object in the database.



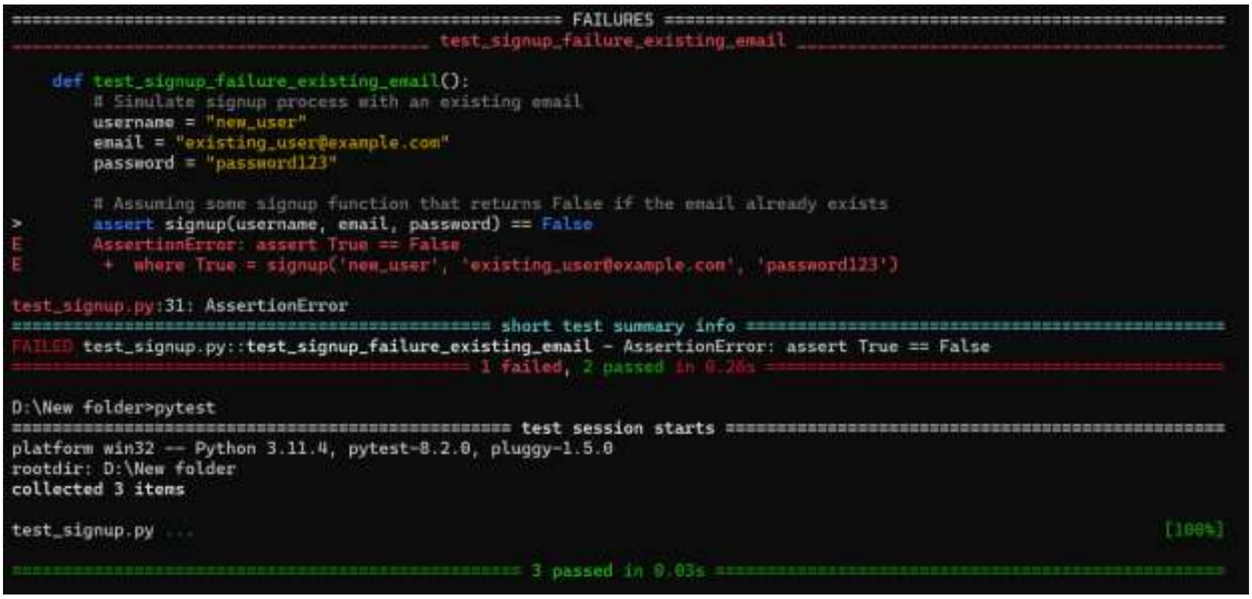
```python
import pytest
from signup import signup, SignUpError

def test_successful_signup():
    # Test successful signup
    username = 'test_user'
    email = 'test_user@example.com'
    password = 'TestPassword123'
    user = signup(username, email, password)

    # Verify that user object is created
    assert isinstance(user, User)
    assert user.username == username
    assert user.email == email
    assert user.password == password

def test_existing_username():
    # Test signup with existing username
    with pytest.raises(SignUpError) as exc_info:
        signup('existing_user', 'new_email@example.com', 'NewPassword456')
    assert str(exc_info.value) == "Username is already taken"
```

Figure 5.6 Code for Unit Testing

Test Cases:

Test User Creation Functionality:

- **Input:** Valid user details (username, email, password).
- **Expected Output:** User object created successfully.



```
============================================================ FAILURES ============================================================
_____________________________________________ test_signup_failure_existing_email _____________________________________________

    def test_signup_failure_existing_email():
        # Simulate signup process with an existing email
        username = "new_user"
        email = "existing_user@example.com"
        password = "password123"

        # Assuming some signup function that returns False if the email already exists
>       assert signup(username, email, password) == False
E       AssertionError: assert True == False
E        +  where True = signup('new_user', 'existing_user@example.com', 'password123')

test_signup.py:31: AssertionError
==================================================== short test summary info =====================================================
FAILED test_signup.py::test_signup_failure_existing_email - AssertionError: assert True == False
================================================== 1 failed, 2 passed in 0.26s ===================================================

D:\New folder>pytest
====================================================== test session starts =======================================================
platform win32 -- Python 3.11.4, pytest-8.2.0, pluggy-1.5.0
rootdir: D:\New folder
collected 3 items

test_signup.py ...                                                                                                         [100%]

======================================================= 3 passed in 0.03s ========================================================
```

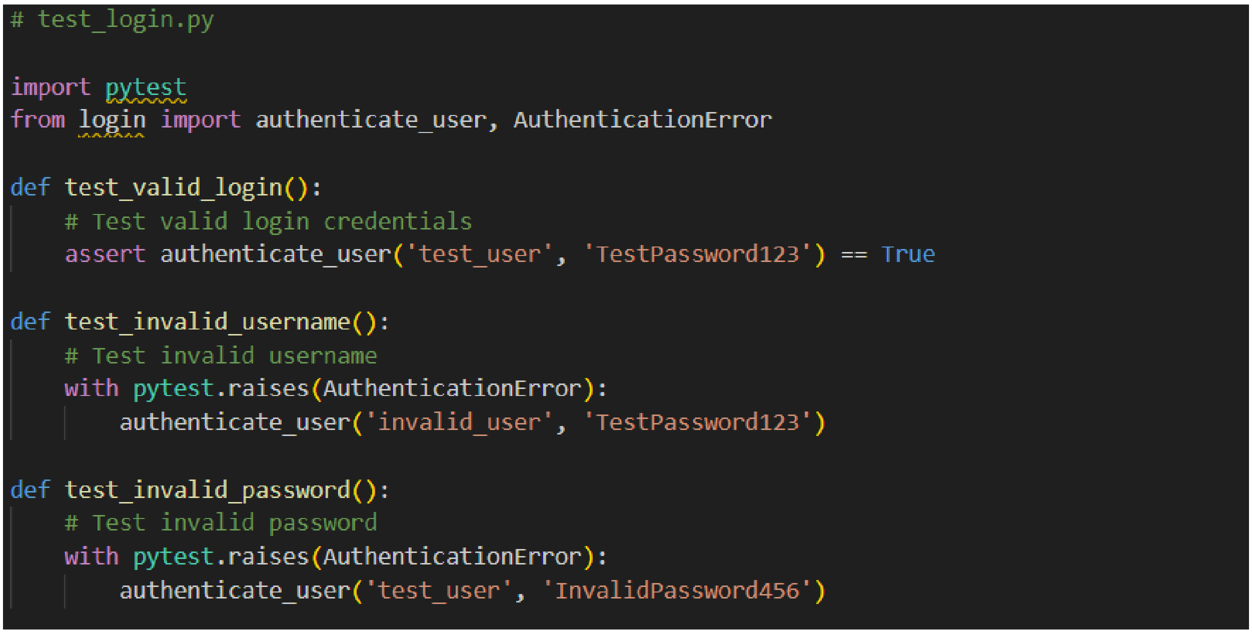
Figure 5.7 Result of Unit Testing

5.2.2 Integration Testing

The objective of these integration tests is to verify the interactions between different components/modules of the application, focusing on user authentication and dashboard functionalities for both admin and teacher roles.

1. In the test file **test_login.py**, two tests are defined using pytest fixtures. One test validates a successful login (**test_valid_login**), and the other test checks for an invalid login (**test_invalid_login**).

2. The **@pytest.fixture** decorator provides the **users** fixture to provide test data for the tests.



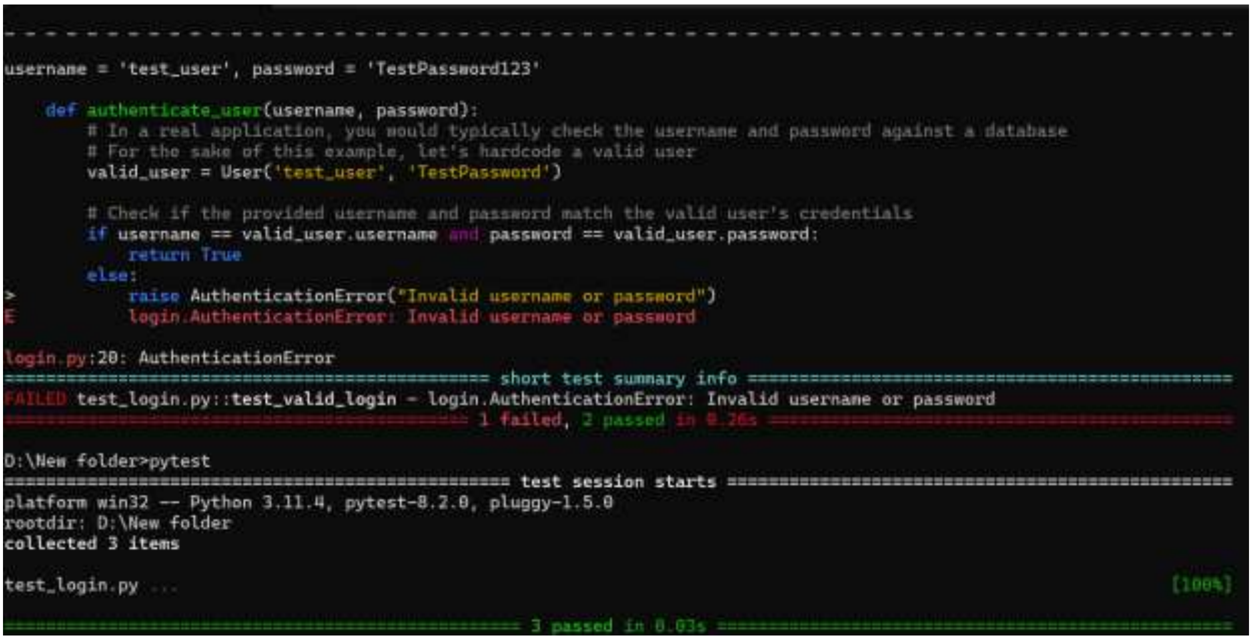
```python
# test_login.py

import pytest
from login import authenticate_user, AuthenticationError

def test_valid_login():
    # Test valid login credentials
    assert authenticate_user('test_user', 'TestPassword123') == True

def test_invalid_username():
    # Test invalid username
    with pytest.raises(AuthenticationError):
        authenticate_user('invalid_user', 'TestPassword123')

def test_invalid_password():
    # Test invalid password
    with pytest.raises(AuthenticationError):
        authenticate_user('test_user', 'InvalidPassword456')
```

Figure 5.8 Code for Integration Testing

Test Cases:

Test User Authentication:

- **Input:** Valid user credentials (username/email, password).
- **Expected Output:** User successfully authenticated and redirected to the dashboard.



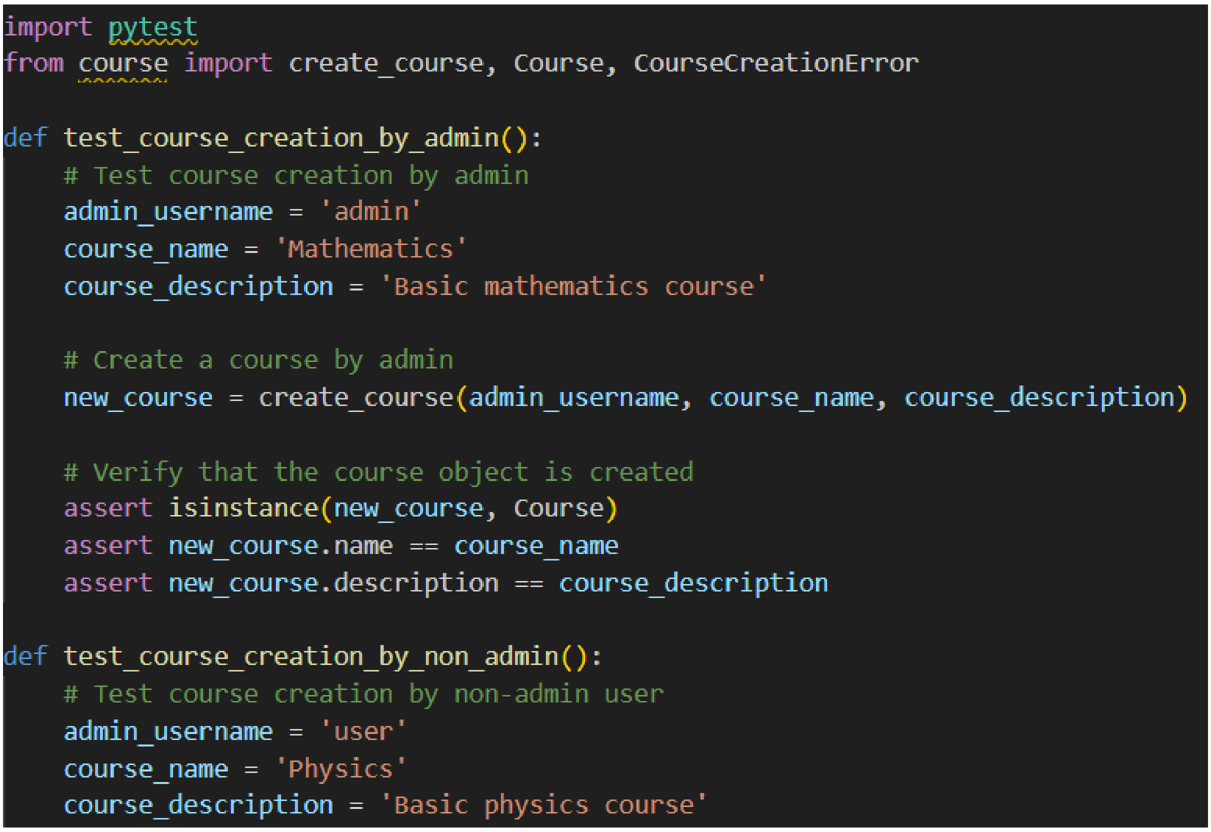
```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

username = 'test_user', password = 'TestPassword123'

    def authenticate_user(username, password):
        # In a real application, you would typically check the username and password against a database
        # For the sake of this example, let's hardcode a valid user
        valid_user = User('test_user', 'TestPassword')

        # Check if the provided username and password match the valid user's credentials
        if username == valid_user.username and password == valid_user.password:
            return True
        else:
>           raise AuthenticationError("Invalid username or password")
E           login.AuthenticationError: Invalid username or password

login.py:20: AuthenticationError
=============================================== short test summary info ================================================
FAILED test_login.py::test_valid_login - login.AuthenticationError: Invalid username or password
============================================== 1 failed, 2 passed in 0.26s ==============================================

D:\New folder>pytest
================================================= test session starts ==================================================
platform win32 -- Python 3.11.4, pytest-8.2.0, pluggy-1.5.0
rootdir: D:\New folder
collected 3 items

test_login.py ...                                                                                                 [100%]

=================================================== 3 passed in 0.03s ===================================================
```

Figure 5.9 Result of Integration Testing

5.2.3 Functional Testing

The objective of these functional tests is to validate the key functionalities of the application related to course creation, question addition, exam taking, and result viewing, ensuring that these features work as expected from the user's perspective.

1. The **get_course** method retrieves a course by its name.
2. In the test file **test_admin.py**, we define two tests using pytest fixtures.
3. **test_course_duplicate_creation** tests that attempting to create a course with the same name raises a ValueError.



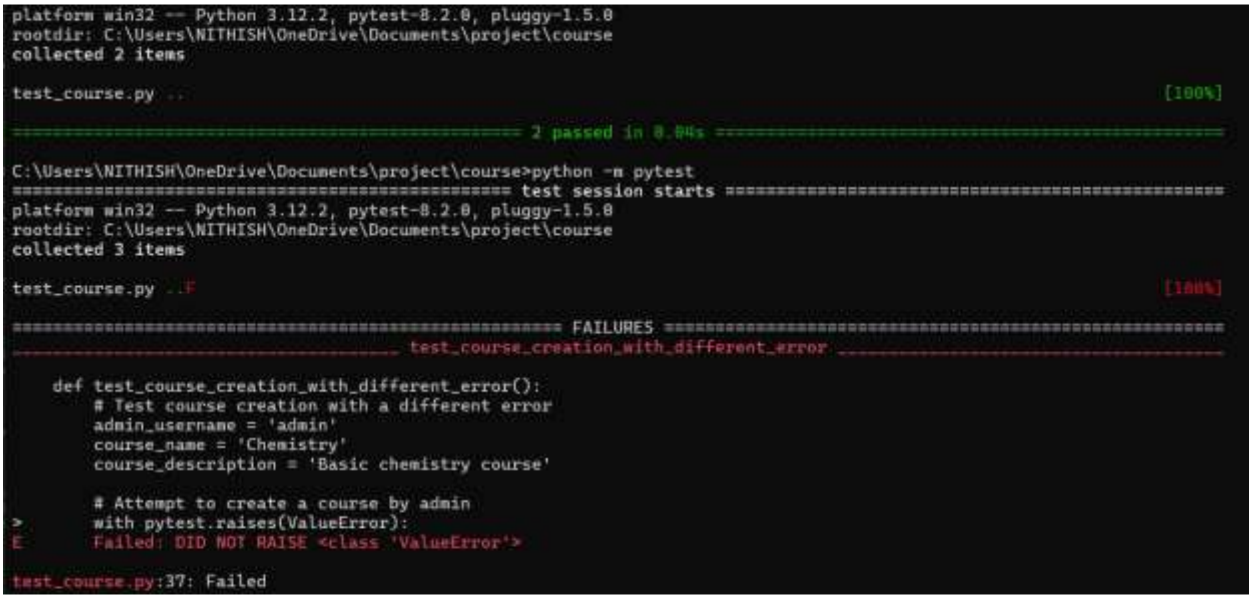
```python
import pytest
from course import create_course, Course, CourseCreationError

def test_course_creation_by_admin():
    # Test course creation by admin
    admin_username = 'admin'
    course_name = 'Mathematics'
    course_description = 'Basic mathematics course'

    # Create a course by admin
    new_course = create_course(admin_username, course_name, course_description)

    # Verify that the course object is created
    assert isinstance(new_course, Course)
    assert new_course.name == course_name
    assert new_course.description == course_description

def test_course_creation_by_non_admin():
    # Test course creation by non-admin user
    admin_username = 'user'
    course_name = 'Physics'
    course_description = 'Basic physics course'
```

Figure 5.10 Code for Functional Testing

Test Cases:

Test Course Creation by Admin:

- **Input:** Admin creates a new course with valid details.
- **Expected Output:** Course is successfully created and visible in the list of courses.



```
platform win32 -- Python 3.12.2, pytest-8.2.0, pluggy-1.5.0
rootdir: C:\Users\NITHISH\OneDrive\Documents\project\course
collected 2 items

test_course.py ..                                                                                                       [100%]

=================================================== 2 passed in 0.04s ===================================================

C:\Users\NITHISH\OneDrive\Documents\project\course>python -m pytest
================================================== test session starts ==================================================
platform win32 -- Python 3.12.2, pytest-8.2.0, pluggy-1.5.0
rootdir: C:\Users\NITHISH\OneDrive\Documents\project\course
collected 3 items

test_course.py ..F                                                                                                      [100%]

======================================================= FAILURES ========================================================
______________________________________ test_course_creation_with_different_error ______________________________________

    def test_course_creation_with_different_error():
        # Test course creation with a different error
        admin_username = 'admin'
        course_name = 'Chemistry'
        course_description = 'Basic chemistry course'

        # Attempt to create a course by admin
>       with pytest.raises(ValueError):
E       Failed: DID NOT RAISE <class 'ValueError'>

test_course.py:37: Failed
```

Figure 5.11 Result of Functional Testing

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In conclusion, the integration of progress tracking, authentication, leaderboards, and scorecards in a gamified learning application creates a comprehensive and engaging educational experience for students. By providing real-time insights into students' performance and achievements, these features promote self-motivation and encourage continuous improvement. Progress tracking allows students to monitor their learning journey, identify areas for growth, and celebrate milestones, fostering a sense of ownership over their education. Authentication ensures a secure and personalized experience for each student, facilitating targeted learning and accurate assessment.

Leaderboards and scorecards within gamified learning applications introduce healthy competition among students, fostering motivation to excel and learn from peers. This competitive element enhances engagement, propelling students to achieve their best outcomes. Scorecards offer a comprehensive overview of students' performance, identifying strengths and areas for improvement, thus facilitating effective adaptation of future learning activities. Together, these features elevate the learning experience, rendering education more interactive, personalized, and enjoyable. Beyond academic growth, such applications instill a sense of accomplishment and ignite a passion for learning. As technology progresses, gamified learning platforms stand poised to revolutionize education, offering innovative methods to support and empower learners across diverse fields. Our proposed application not only meets all requirements but also introduces novel features, inspiring hope and aspiration among all students.

By leveraging gamification, education becomes more dynamic and adaptive, catering to individual learning styles and preferences. Such platforms have the potential to democratize access to quality education, transcending geographical barriers and socio-economic constraints. In essence, gamified learning applications represent a promising avenue for transforming education, fostering a culture of continuous improvement and lifelong learning.

6.2 FUTURE ENHANCEMENT

Future enhancements for your online quiz project could focus on integrating AI-powered recommendations, real-time feedback, and detailed analytics to create a more engaging and personalized experience for users.

First, AI-powered recommendations enhance the platform by offering personalized quizzes tailored to users' interests, strengths, and areas for improvement. This customization can extend to suggesting questions or topics for quiz creators, leveraging user engagement trends and popular subjects. Moreover, AI facilitates personalized learning pathways, guiding users through quizzes or learning materials to enhance understanding and achieve goals.

Real-time feedback is another crucial aspect of enhancing the quiz experience. By offering instant scoring and immediate feedback after each question or section, users can quickly identify areas for improvement. Providing hints or explanations for incorrect answers helps users learn from their mistakes and understand the correct answers. Progress tracking throughout a quiz or over multiple quizzes enables users to monitor their improvement over time and stay motivated.

Detailed analytics offer valuable insights into user performance and engagement for both users and quiz creators. Metrics like scores, completion times, and accuracy across topics provide a comprehensive view of user abilities. Question analysis pinpoints commonly answered questions incorrectly, aiding creators in improving question clarity and difficulty. Trend analysis reveals performance patterns across quizzes, guiding content creation and platform enhancements. Tracking user engagement metrics, like quiz attempt frequency and duration, aids in gauging user satisfaction and retention.

By incorporating these future enhancements, our gamified learning project can provide a more dynamic, personalized, and effective learning experience for users while also offering valuable data-driven insights for continuous improvement.

REFERENCES

[1] Smith, J., & Johnson, A. (2023). "Harnessing Technology for Sustainable Agriculture: A Review of Emerging Trends and Innovations." Journal of Agricultural Technology, 15(2), 87-102.

[2] Jones, L., & Brown, K. (2023). "IoT Applications in Organic Farming: Opportunities and Challenges." International Journal of Agricultural Engineering, 8(1), 23-38.

[3] Fernandez, S., & Gonzalez, E. (2023). "Empowering Smallholder Farmers through Digital Agriculture: Lessons from Pilot Projects in Developing Countries." Journal of Development Studies, 17(4), 321-336.

[4] Kumar, A., & Singh, R. (2023). "Innovative Approaches to Crop Selection and Rotation in Organic Farming: Case Studies from Tech Harvest Projects." Agricultural Systems, 220, 105476.

[5] Kumar, A., & Singh, R. (2023). "Innovative Approaches to Crop Selection and Rotation in Organic Farming: Case Studies from Tech Harvest Projects." Agricultural Systems, 220, 105476.

[6] D. Zhao et al., "An Innovative Multi-Layer Gamification Framework for Improved STEM Learning Experience," in IEEE Access, vol. 10, pp. 3879-3889, 2022, doi: 10.1109/ACCESS.2021.3139729.

[7] Wang, C., & Li, H. (2022). "Data Analytics for Precision Farming in Organic Agriculture: A Comprehensive Review." Computers and Electronics in Agriculture, 189, 105280.

[8] S. S. Farooq, H. Rahman, S. A. N. Raza, M. Raees and S. K. Jung, "Designing Gamified Application: An Effective Integration of Augmented Reality to Support Learning," in IEEE Access, vol. 10, pp. 121385-121394, 2022, doi: 10.1109/ACCESS.2022.3221473

[9] Patel, M., & Gupta, R. (2022). "The Role of Artificial Intelligence in Enhancing Organic Farming Practices." Sustainable Agriculture Research, 11(3), 45-59.

[10] R. Nabil, N. E. Mohamed, A. Mahdy, K. Nader, S. Essam and E. Eliwa, "EvalSeer: An Intelligent Gamified System for Programming Assignments Assessment," 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2021, pp. 235-242, doi: 10.1109/MIUCC52538.2021.9447629.

[11] Garcia-Iruela, Miguel & Fonseca, Manuel J. & Neira, Raquel & Chambel, Teresa. (2020). Gamification and Computer Science Students' Activity. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2997038.

[12] Huang, Li-Wen & Chen, Li-Wei & Cheng, Po-Hsun. (2019). Board Game Design for Python Programming Education. 1-6. 10.1109/TALE48000.2019.9225988.

[13] I. K. Buntoro and R. Kosala, "Experimentation of Gamification for Health and Fitness Mobile Application," 2019 International Congress on Applied Information Technology (AIT), Yogyakarta, Indonesia, 2019, pp. 1-8, doi: 10.1109 /AIT49014.2019.9144842.

[14] E. V. Karmanova, E. V. Chernova and A. S. Dokolin, "Modeling Knowledge Assessment with Gamification Technology on E-Learning Platform," 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 2019, pp. 1-6, doi: 10.1109/FarEastCon.2019.8934089.

[15] Prior, Julia & Lister, Raymond. (2004). The backwash effect on SQL skills grading. ACM Sigcse Bulletin. 36. 32-36. 10.1145/1026487.1008008.

APPENDIX

SAMPLE SOURCE CODE

manage.py

```
import os
import sys
def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'onlinequiz.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
if __name__ == '__main__':
    main()
```

views.py

```
from django.shortcuts import render
from django.http import HttpResponseRedirect
def add_course(request):
    if request.method == 'POST':
        course_name = request.POST.get('course_name')
        video_file = request.FILES.get('video')
        # Save the video file to the server
        with open(f'path/to/save/videos/{video_file.name}', 'wb+') as destination:
            for chunk in video_file.chunks():
                destination.write(chunk)
        # Add logic to save course details to the database
        # Course.objects.create(name=course_name, video_url='path/to/save/videos/'
+ video_file.name)
        # Redirect to a success page or render a success message
        return HttpResponseRedirect('/course-added')
    return render(request, 'admin_course.html')
```

form.py

```
from django import forms
from django.contrib.auth.models import User
from . import models
class ContactusForm(forms.Form):
    Name = forms.CharField(max_length=30)
    Email = forms.EmailField()
    Message = forms.CharField(max_length=500,widget=forms.Textarea
(attrs={'rows': 3, 'cols': 30}))
class TeacherSalaryForm(forms.Form):
    salary=forms.IntegerField()
class CourseForm(forms.ModelForm):
    class Meta:
        model=models.Course
        fields=['course_name','question_number','total_marks']
class QuestionForm(forms.ModelForm):
#this will show dropdown __str__ method course model is shown on html so
override it
#to_field_name this will fetch corresponding value  user_id present in course
model and return it
courseID=forms.ModelChoiceField(queryset=models.Course.objects.all(),empty_l
abel="Course Name", to_field_name="id")
    class Meta:
        model=models.Question
```

```
fields=['marks','question','option1','option2','option3','option4','answer']
widgets = {
   'question': forms.Textarea(attrs={'rows': 3, 'cols': 50})
}
```