**1.INTRODUCTION**

* 1. **Purpose**

The purpose of this project is to create a quiz game using HTML, CSS, and JavaScript. The quiz game will serve as an interactive and engaging way for users to test their knowledge on a specific topic.

The project aims to achieve the following goals:

1. To create a user-friendly and visually appealing interface for the quiz game that will engage users.

2. To provide a fun and interactive way for users to learn about a specific topic and test their knowledge.

3. To demonstrate our skills in front-end web development using HTML, CSS, and JavaScript.

4. To gain experience in designing and implementing interactive applications.

5. To learn best practices for designing and developing engaging user interfaces.

Overall, the purpose of this project is to create an educational and enjoyable experience for users while providing an opportunity for us to learn and practice our skills in web development.

* 1. **Scope**

The scope of this project includes the development of a quiz game using HTML, CSS, and JavaScript. The quiz game will have multiple-choice questions related to a specific topic, and users will be able to select an answer and receive feedback on whether their answer is correct or incorrect. The game will keep track of the user's score and display it at the end of the quiz.

The project scope includes the following features:

1. User-friendly and visually appealing interface for the quiz game.

2. Multiple-choice questions related to a specific topic.

3. Feedback on user answers, indicating whether they are correct or incorrect.

4. Keeping track of the user's score and displaying it at the end of the quiz.

5. Responsive design that can be viewed on multiple devices.

6. Ability to reset the quiz and start over.

7. Code that is well-organized, documented, and follows best practices.

The project scope does not include the development of a back-end system for storing and processing data related to the quiz questions and user responses. Additionally, the project does not include the implementation of any advanced features such as a timer, leader board, or social media sharing.

* 1. **References**

1. SRS Document Template.

2. San Jose State University.pdf

3. Christian Service University College.

4. DAITM.pdf

5. Nakul Natu. (2011). JavaScript Game Engine using HTML, CSS and JavaScript.

* 1. **Overview**

As far as we know, this project involves the development of a quiz game using HTML, CSS, and JavaScript. The quiz game is designed to test the knowledge of users on a specific topic and provide a fun and interactive way for users to learn and engage with the content.

The project may involve the following steps:

1. Identifying the topic and difficulty level for the quiz game.

2. Developing the quiz questions and answers.

3. Designing the user interface for the quiz game using HTML and CSS.

4. Implementing the quiz game functionality using JavaScript.

5. Testing and debugging the quiz game to ensure that it functions as intended.

6. Deploying the quiz game to a web server or hosting platform.

The development of the quiz game will require a strong understanding of HTML, CSS, and JavaScript, as well as an ability to design an intuitive and engaging user interface. The project may also involve working with stakeholders or subject matter experts to ensure that the quiz questions and content are accurate and effective in achieving the educational objectives.

Overall, the goal of the project is to create a high-quality and engaging quiz game that provides value to the target audience and achieves the desired educational outcomes.

**2.OVERALL DESCRIPTION**

**2.1 Product Perspective**

This project is meant to serve as a common platform where users can test their knowledge and it can be a part of fun in their busy schedule. Our goal is to develop a user-friendly game which includes subjective questions, practical questions etc.

**2.1.1 System Interface**

The user inputs data via the web server using HTML forms. The actual program that will perform the operations is written in JAVASCRIPT.

**2.1.2 User Interface**

The new system shall provide a very intuitive and simple interface to the user and the administrator, so that the user can easily navigate through questions, result, score board.

**2.1.3 Hardware Interface**

**a) Server side:** The web application will be hosted on a web server which is listening on the web standard port, port 80.

**b) Client-side:** Monitor screen – the software shall display information to the user via the monitor screen

Mouse – the software shall interact with the movement of the mouse and the mouse buttons. The mouse shall activate areas for data input, command buttons and select options from menus. Keyboard – the software shall interact with the keystrokes of the keyboard. The keyboard will input data into the active area of the database.

**2.1.3 Software Interface**

**a) Server side:** Which can run HTML, CSS.

**b) Client side:** An OS which can run a modern web browser which supports JavaScript.

**2.1.4 Communication Interface**

The HTPP or HTTPS protocol(s) will be used to facilitate communication between the client and server.

**2.1.6 Memory Constraints**

Memory constraints will come into play when the size of HTML grows to a considerable size.

**2.1.7 Operations**

The product shall have operations to protect the data from being corrupted or accidentally altered during a system failure.

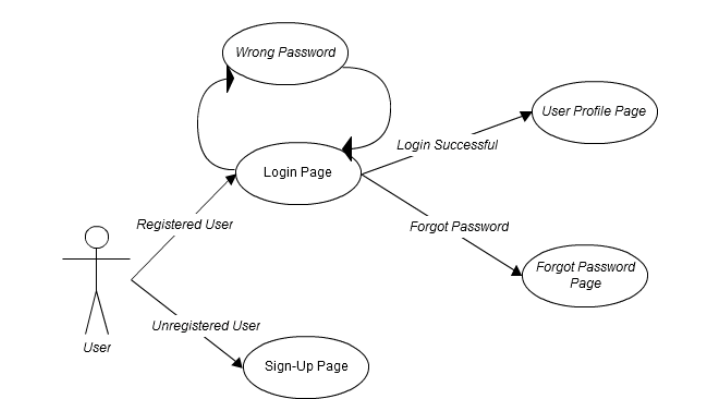
**2.1.8 Site adaption requirements**

Not applicable.

**2.2 Product Function**

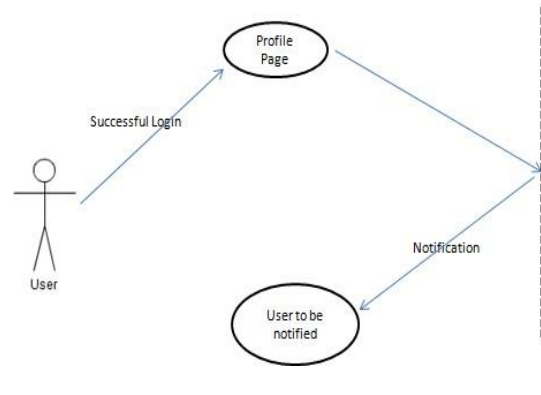
**2.2.1 Use case diagrams**

**2.2.1.1 User Login**

****

**Figure 1: User Login**

**2.2.1.2 Discussion Threads**

****

**Figure 2: Discussion Threads**

**2.3 User Characteristics**

**2.3.1 Students**

Students are the primary consumers of an academic portal. They are accessing information posted by System administrators, professors, which are used to gain knowledge about the quiz.

**2.3.2 Professors**

Professors are the primary content administrators of an academic portal. They are uploading questions for the students to prepare for the quiz.

**2.3.3 System Administrators**

System administrators are primarily responsible for maintaining the academic portal. They contribute minimally to the courses themselves but spend more time modifying the system’s configuration and making appropriate updates.

**2.4 Constraints**

**2.4.1 User interface constraints**

Using this system, it is simple and intuitive. A user familiar with basic browser navigation skills should be able to understand all functionality provided by the system.

**2.4.2 Hardware constraints**

The system should work on most home desktop and laptop computers which support JavaScript, CSS, and HTML.

**2.4.3 Software constraints**

The system will be intended to run on the visual studio code, browser, google chrome, Firefox etc.

**2.4.4 Data Management constraints**

System shall be able to interface with other components according to their specifications.

**2.4.5 Operational Constraints**

The system should be implemented in html, CSS, and JavaScript.

**2.5 Assumptions and Dependencies**

Most of the academic portals have a lot of redundant features which are rarely used in an academic session. Our new system focuses on the features which are most important to the users of an academic institute along with introduction of some new features which other portals lacks.

**3.SPECIFIC REQUIREMENTS**

**3.1 External Interface**

* + 1. **Web Server**
* Apache will be used as web server:
* The user inputs data via the web server using HTML forms.
* The web server displays a HTML page as result to the end-user.
  + 1. **PHP Application**

The actual program that will perform the operations is written in PHP. All data will be stored in a database.

**3.1.3 MySQL Database**

It’s an open-source SQL database to store all data which communicates with the application on the server

* 1. **Functional Requirements**
     1. **Use Case Scenario**

**3.2.1.1 Use Case Scenario 1- User Login**

|  |  |
| --- | --- |
| **Purpose** | User logs in to system using existing profile. |
| **User** | A user with an existing profile |
| **Input data** | Profile username and password. |
| **Output Data** | Corresponding page data |
| **Invariants** | Profile table data and user information. |
| **Pre- Conditions** | User is not logged in to a profile, input profile exists in data base, user password matches profile |
| **Post- Conditions** | User's computer has been supplied with appropriate cookie, page data is appropriate for selected profile |
| **Basic Flow** | Webpage looks up profile data and returns the matching cookie. Webpage is updated to match new user data. |
| **Alternative Flows** | Invalid password, invalid username, or mismatched username and password redirect to error message and previous page. |
| **Business Rules** | This allows users to log in to their profile from anywhere. |

**3.2.1.2 Use Case Scenario-2 – Content Sharing(Upload questions for the students)**

|  |  |
| --- | --- |
| **Purpose** | User wants to share some data |
| **User** | A user with an existing profile to edit content |
| **Input data** | The files to be shared. |
| **Output Data** | Files are ready for the students to gain knowledge. |
| **Invariants** | The file |
| **Pre- Conditions** | User is logged in to modify or upload information. |
| **Post- Conditions** | Other than user any one can access the data which is shared |
| **Basic Flow** | Webpage looks up profile data and returns the matching cookie. Webpage is updated to match new user data. |

**3.2.1.3 Use Case Scenario 3-Content Sharing**

|  |  |
| --- | --- |
| **Purpose** | Students can access information. |
| **User** | A user can play quiz with the information shared. |
| **Input data** | Requests the server to get quiz questions. |
| **Output Data** | User can access quiz |
| **Invariants** | The User and the File |
| **Pre- Conditions** | User is logged in and the file must be shared with him or with the group which he is a member of. |
| **Post- Conditions** | The user can access full information. |
| **Basic Flow** | User logins and selects the type of quiz they want to play. |

**3.2.1.4 Use Case Scenario 4-Search Result**

|  |  |
| --- | --- |
| **Purpose** | Students can access information what they searched. |
| **User** | Any user of academic portal. |
| **Input data** | The keyword. |
| **Output Data** | Search results. |
| **Invariants** | The User and the portal itself. |
| **Pre- Conditions** | User is logged in. |
| **Post- Conditions** | Search results. |
| **Basic Flow** | User logins enters the keyword in search box, clicks the search button and gets search results. |

* 1. **Performance Requirements**

The system should support more users without any blocking of information. This statement provides a general sense of reliability when the system is under load. It is important that a substantial number of users be able to access the system at the same time, since an academic portal is important to the courses that employ it. The times when the system will be under the most stress are likely during assignment submissions.

* 1. **Logical database requirements**

All data will be saved in the database: user accounts and profiles, discussion data, messages etc. (except files which are stored on the disk.) The database allows concurrent access and will be kept consistent at all times, requiring a good database design.

* 1. **Design Constraints**

1. The communication between the portal software and the database will be in SQL.

2. The portal layout will be produced with HTML/CSS.

3. The product will be written in HTML.

4. System administrators must have access to comprehensive documentation.

* 1. **Software System Attributes**

The software consists of the following elements:

1. The Apache web server

2. The PHP application

3. The MySQL database

4. The database should remain consistent at all times in case of an error

**3.6.1 Reliability**

The reliability of the overall program depends on the reliability of the separate components.

**3.6.2 Availability**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved with the MySQL server and saved by the administrator.

**3.6.3 Security**

1. Passwords will be saved encrypted in the database in order to ensure the user's privacy.

2. The user's IP will be logged.

3. The system will be protected against vulnerabilities such as SQL injection attacks.

**3.6.4 Maintainability**

MySQL is used for maintaining the database and the Apache server takes care of the site. In case of a failure, a re-initialization of the program is recommended.

**3.6.5 Portability**

The application is Linux-based and should be compatible with other systems. Apache, PHP and MySQL programs are practically independent of the OS-system which they communicate with. The end-user part is fully portable and any system using any web browser should be able to use the features of the application.