|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PROJECT ID** | |  | |  | **NUMBER OF MEMBERS** | 02 |
|  | | | | | | |
| **TITLE** | Smart Quiz Manager | | | | | |
|  | | | | | | |
| **SUPERVISOR NAME** | | | Ch. Shehzad Faisal | | | |

|  |  |  |
| --- | --- | --- |
| **MEMBER NAME** | **REG. NO.** | **EMAIL ADDRESS** |
| Muhammad Hammad Ali | CIIT/SP17-BCS-021/ATK | hammadali6096@gmail.com |
| Muhammad Hamza Islam | CIIT/SP17-BCS-023/ATK | hamza111.mhi@gmail.com |

**CHECKLIST:**

|  |  |  |  |
| --- | --- | --- | --- |
| Number of pages in this report |  | 52 |  |
| I/We have enclosed the soft-copy of this document along-with the codes and scripts created by myself/ourselves | **YES / NO** | | |
| My/Our supervisor has attested the attached document | **YES / NO** | | |
| **I/We confirm to state that this project is free from any type of plagiarism and misuse of copyrighted material** | **YES / NO** | | |
|  |  | | |

|  |  |  |
| --- | --- | --- |
| **MEMBERS’ SIGNATURES** |  | |
|  |  | Supervisor’s Signature |
|  |  |
|  |  |

**This work, entitled “SMART QUIZ MANAGER” has been approved for the award of**

**Bachelors of Science in Computer Science**

**Date**

**External Examiner:**

**Head of Department:**

Department of Computer Science

COMSATS University Islamabad

ATTOCK – PAKISTAN

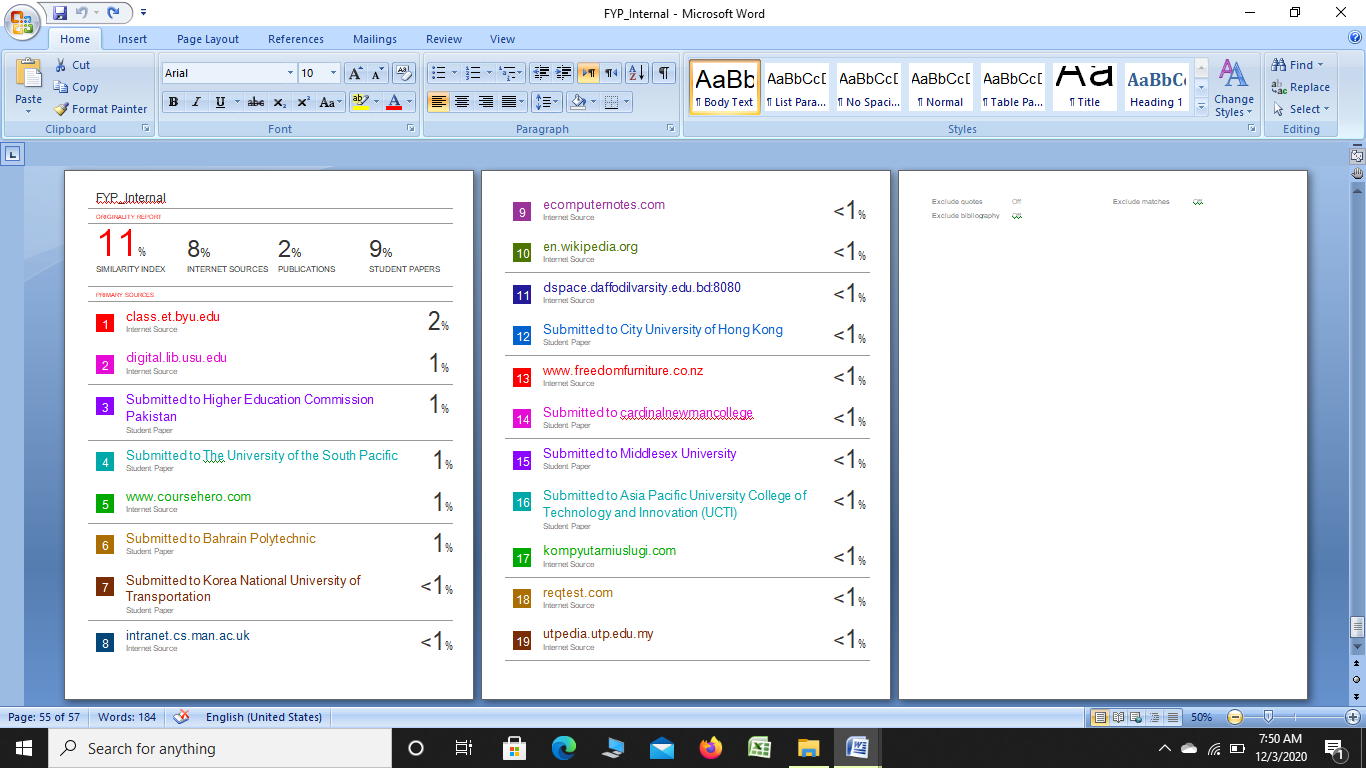


Table of Contents

[**1** **Introduction** 8](#_Toc57468174)

[**1.1** **Brief** 9](#_Toc57468175)

[**1.2** **Relevance to Course Modules** 10](#_Toc57468176)

[**1.2.1** **Student** 10](#_Toc57468177)

[**1.2.2** **Teacher** 10](#_Toc57468178)

[**1.2.3** **Authentication** 10](#_Toc57468179)

[**1.2.4** **Word based Algorithm** 10](#_Toc57468180)

[**1.2.5** **Face Recognition** 10](#_Toc57468181)

[**1.3** **Project Background** 10](#_Toc57468182)

[**1.4** **Benefits** 11](#_Toc57468183)

[**1.5** **Domain** 11](#_Toc57468184)

[**1.6** **Literature Review** 11](#_Toc57468185)

[**1.7** **Analysis from Literature Review** 12](#_Toc57468186)

[**1.8** **Methodology and Software Lifecycle for This Project** 13](#_Toc57468187)

[**1.8.1** **Rationale behind Selected Methodology** 13](#_Toc57468188)

[**2** **Problem Definition** 14](#_Toc57468189)

[**2.1** **Problem Statement** 15](#_Toc57468190)

[**2.2** **DELIVERABLES** 15](#_Toc57468191)

[**2.3** **Development Requirements** 15](#_Toc57468192)

[**2.3.1** **OS Requirements** 16](#_Toc57468193)

[**2.3.2** **DBMS Requirements** 16](#_Toc57468194)

[**2.4** **Related System Analysis/Literature Review** 16](#_Toc57468195)

[**2.5** **How our project different from existing applications?** 16](#_Toc57468196)

[**3** **Requirements Analysis** 17](#_Toc57468197)

[**3.1** **Use Cases** 18](#_Toc57468198)

[**3.1.1** **Student:** 18](#_Toc57468199)

[**3.1.2** **Teacher** 19](#_Toc57468200)

[**3.2** **Detailed Use cases** 19](#_Toc57468201)

[**3.3** **Requirements** 23](#_Toc57468202)

[**3.3.1** **Functional requirements** 23](#_Toc57468203)

[**3.3.2** **Non-functional requirements** 24](#_Toc57468204)

[**4** **Design and Architecture** 26](#_Toc57468205)

[**4.1** **System Architecture** 27](#_Toc57468206)

[**4.1.1** **How does our application works** 27](#_Toc57468207)

[**4.1.2** **Block Diagram** 27](#_Toc57468208)

[**4.2** **Data Representation** 28](#_Toc57468209)

[**Data Flow Diagram (level 0)** 28](#_Toc57468210)

[4.3 Process Flow/Description 30](#_Toc57468211)

[**4.4** **Sequence Diagrams** 32](#_Toc57468212)

[**4.4.2** **Take a Quiz:** 33](#_Toc57468213)

[**4.4.3** **Question addition and modification:** 34](#_Toc57468214)

[**4.4.5** **Activity Diagram:** 36](#_Toc57468215)

[5.1 **Tools and Technologies** 38](#_Toc57468216)

[5.1.1 **Tools** 38](#_Toc57468217)

[5.1.2 **Languages** 38](#_Toc57468218)

[5.1.3 **Data Base** 38](#_Toc57468219)

[5.2 **Software Requirements** 38](#_Toc57468220)

[5.3 **Hardware Requirements** 38](#_Toc57468221)

[5.4 **User Interface** 38](#_Toc57468222)

[**5.4.5** **Login Model** 42](#_Toc57468223)

[**5.4.6** **Teacher Dashboard** 42](#_Toc57468224)

[**5.4.7** **Create Quiz** 43](#_Toc57468225)

[**5.4.8** **Quizzes Saved in Draft** 43](#_Toc57468226)

[**5.4.9** **Add Question in Quiz** 44](#_Toc57468227)

[**5.4.10** **Preview Draft Quiz** 44](#_Toc57468228)

[**5.4.11** **Create Subjective Question** 45](#_Toc57468229)

[**5.4.12** **All Attempted Quizzes** 45](#_Toc57468230)

[**5.4.13**  **List of Participant** 46](#_Toc57468231)

[**5.4.14**  **List of Registered Students** 46](#_Toc57468232)

[**5.4.15**  **Student Profile** 47](#_Toc57468233)

[**5.4.16**  **Teacher Profile** 47](#_Toc57468234)

[**5.4.17**  **Edit Teacher Profile** 48](#_Toc57468235)

[**5.4.18**  **Request to join** 48](#_Toc57468236)

[**5.4.19**  **Student Dashboard** 49](#_Toc57468237)

[**5.4.20**  **Instruction Page for Student** 49](#_Toc57468238)

[**5.4.21**  **Verify Student Through Face Recognition** 50](#_Toc57468239)

[**5.4.22**  **Attempt Subjective Quiz** 50](#_Toc57468240)

[**5.4.23**  **Attempt MCQs Quiz** 51](#_Toc57468241)

[**5.4.24**  **Student Profile** 51](#_Toc57468242)

[**5.4.25**  **Edit Profile** 52](#_Toc57468243)

[6.1 **Testing** 54](#_Toc57468244)

[**6.1.1** **Black Box Testing** 54](#_Toc57468245)

[**6.1.2** **White Box Testing** 54](#_Toc57468246)

[**6.1.3** **Unit Testing** 55](#_Toc57468247)

[**6.1.4** **Functional Testing** 55](#_Toc57468248)

[o Objective 55](#_Toc57468249)

[o Test Cases 55](#_Toc57468250)

[**6.1.4.1**  **Test Case (Teacher and Student Login)** 56](#_Toc57468251)

[**6.1.4.2**  **Test Case (Users Authentication and Registration)** 56](#_Toc57468252)

[**6.1.4.3**  **Test Case (Add Quiz)** 57](#_Toc57468253)

[**6.1.4.4**  **Test Case (Add Questions)** 57](#_Toc57468254)

[**6.1.4.5**  **Test Case (Face Recognition)** 58](#_Toc57468255)

[**6.1.4.6**  **Test Case (Static Screen)** 58](#_Toc57468256)

[**Conclusion** 60](#_Toc57468257)

# **List of Figures**

[Figure 5-5 SQM Login Model 42](#_Toc57468304)

[Figure 5-6 SQM Teacher Dashboard 42](#_Toc57468305)

[Figure 5--7 SQM Create Quiz 43](#_Toc57468306)

[Figure 5-8 SQM Quizzes Saved in Draft 43](#_Toc57468307)

[Figure 5-9 SQM Add Question in Quiz 44](#_Toc57468308)

[Figure 5-10 SQM Preview Draft Quiz 44](#_Toc57468309)

[Figure 5-11SQM Create Subjective Question 45](#_Toc57468310)

[Figure 5-12 SQM All Attempted Quizzes 45](#_Toc57468311)

[Figure 5-13 SQM List of Participant 46](#_Toc57468312)

[Figure 5-14 SQM List of Registered Students 46](#_Toc57468313)

[Figure 5-15 SQM Student Profile 47](#_Toc57468314)

[Figure 5-16 SQM Teacher Profile 47](#_Toc57468315)

[Figure 5-17 SQM Edit Teacher Profile 48](#_Toc57468316)

[Figure 5-18 SQM Request to join 48](#_Toc57468317)

[Figure 5-19 SQM Student Dashboard 49](#_Toc57468318)

[Figure 5-20 SQM Instruction Page 49](#_Toc57468319)

[Figure 5-21 SQM Verify Student Through Face Recognition 50](#_Toc57468320)

[Figure 5-22 SQM Attempt Subjective Quiz 50](#_Toc57468321)

[Figure 5-23 SQM Attempt MCQs Quiz 51](#_Toc57468322)

[Figure 5-24 SQM Student Profile 51](#_Toc57468323)

[Figure 5-25 SQM Edit Profile 52](#_Toc57468324)

[Figure 6-1 Black Box Testing 54](#_Toc57468325)

[Figure 6-2 White Box Testing 55](#_Toc57468326)

# **List of Tables**

[Table 6-1 Test Case (Teacher and Student Login) 56](#_Toc57468333)

[Table 6-2 Test Case Registration 56](#_Toc57468334)

[Table 6-3 Test Case (Add Quiz) 57](#_Toc57468335)

[Table 6‑4 Test Case (Add Questions) 57](#_Toc57468336)

[Table 6‑5 Test Case (Face Recognition) 58](#_Toc57468337)

[Table 6‑6 Test Case (Place Order) 58](#_Toc57468338)

**Chapter # 1**

# **Introduction**

## **1.1 Brief**

Nowadays many countries depends a lot on modern technology, the modern technology is now having a huge impact on the status of a country. People are showing their interest to explore different modern ways to conduct their examination system. But, the people can’t find easily trustable invigilation system. So here, we gave them a platform in the form of a web Application. We are going to provide a solution for teacher and students to take and give exam on computer based examination system in the system has different functionality which is helpful for both teachers and student’s teachers create paper easily and share it with all students and when teacher allow the student than they will able to solve the paper and submit the paper than the system will check MCQs and subjective automatically after that obtained marks is shown to student in his account. If the students try to cheat or will take help from the internet the web cam will take clicks after fix time slice and if the moments are suspicious than system will automatically cancel the quiz.

Computer-based examinations contents sources paying attention on creating productive evaluation questions and have attention on exam's feedback remittance to students. The modern establishments are computerized and computers are working according to the given instructions, it is necessary for the effort of people in a modern firm. “Smart Quiz Manager” is very helpful for Institutions to prepare an quiz, save the time that will take to generate and check the paper and make results. The valid use of “Smart Quiz Manager" is that any Educational Institution can easily used it to enhance their policy for conducting the tests and for having best results in short period of time.

One platform which can deal to all types of online exam needs. In-built mechanism which stops participants from typical exam malpractices like copying from others or bringing cheats-sheets .In case of accidental termination of the tests, system will generate a message that the test is cancel. All major types of questions can be created. It will assist the Establishment with testing of students and build up their abilities. However, the inconvenience for this framework is that we need same number of PCs with similar number of students.

As much as most local institutions of learning have hold E-learning to assemble the bar and so as to manage with the growth of large number of students(learners) some of institutions have choose an online examination system, in fact little has been done in development of online systems. A great deal of time and cash are wasted since students need to move starting with one spot then onto the next to give tests. Time and man power is also consumed as setting and stamping of the tests is done physically. The purpose of Smart Quiz Manager is to take online test in an well organized way and no wastage of time for marking the paper. The main purpose of Smart Quiz Manager is to correctly grade the candidates through a completely mechanized framework that spares parcel of time as well as gives quickly and fulfill capable outcomes.

## **Relevance to Course Modules**

### **Student**

Student will login to the system. He/She can check notifications of the teacher .He can join a class, attempt their quiz and view their grades.

### **Teacher**

This module includes the functionality which will be performed by admin/teacher. i.e. quiz cancelation, quiz announcements, grading, etc.

### **Authentication**

This module includes sign up and login etc. And also for that the user of one class can’t access other class without proper login.

### **Word based Algorithm**

This algorithm will check similarity by checking number of similar words using cosine similarity algorithm.

### **Face Recognition**

This module will recognize the student before taking quiz. We are using face-recognition library to do this.

## **Project Background**

There is number of applications which checks quiz but they only have facility of MCQs checking. Estimating the similarity between words, sentences, text and records is a vital part in different tasks, for example, data recovery, document clustering, short answer grading, machine translation and text summarization. In our project we will use the algorithm known as word based in which we use Cosine Similarity distance algorithm which is used to find out similarity between documents regardless of their size. Formula for cosine doesn’t depend on magnitude of inputs. Algorithm is case delicate so if words are in various case, they consider unique. Arrangement of words doesn't make a difference. Two sentences with same words in various requests consider identical.

We will developed project named as “**Smart Quiz Manager**”. In this system teacher can create a class and student can get registered by himself and join that class by a particular link which is given by the teacher. This application will be provided to the student and as well as to the teacher to connect with each other .The student can see the deadline of the quiz which will be uploaded by the teacher. This web based application is user friendly.

## **1.4 Benefits**

One platform which can cater to all types of online exam needs and. ln-built mechanism which restricts participants from typical exam malpractices like copying from others or carrying cheats-sheets .In case of accidental termination of the tests, system allows continuing from the point of discontinuation. All major types of questions can be created.

## **1.5 Domain**

The domain of the project is Web based application, educational tool and Computer-Aided Education.

## **1.6 Literature Review**

The existing system is manual in which user or student are going to maintain the books which can be stored the information like schedule, instructor details, and general details etc. where student will give their feedback about exams which they given in last sessions like semester or annually.

The following disadvantages of system introduce previously focus the need for moderate system:

* A lot of Xerox papers should be made.
* A lot of reform work while hold up in announcing the results.
* A lot of table form activities to make or gather every subject results.
* In existing systems students easily perform cheating.
* A system should not have subjective part.
* The current system is in efficient.

It is very difficult to judge the exam manually.

## **1.7 Analysis from Literature Review**

To develop an application, that is faster than manual system. To develop simple, handy, effective and advanced application that is time consuming and made an ease for the student and teacher.

To overcome this problem we developed a web based application which helps teacher and student both. This application is used to conduct online examination which is not commonly used in universities, schools and colleges. The students should sit at separate terminals and enter to system to write the quiz in the given time period.

The students’ first register his/her details and then make the account after that student is able to login the system. When the user login the system then he want to connect with the faculty module before login student is not able to connect with faculty module. The user home page is contain two buttons one is for connect and the second is for the start exam so first student is connected to the faculty module when student is connect now he is not allow for the paper without the permission of the instructor when the teacher allow the student than the start exam button is enable on the permission of teacher. On starting a paper first part is MCQs and in this section when they choose the option and then go to the next than the objective become lock, after the objective part there is the subjective part which includes questions.

After clicking on the submit key the quiz is submitted and system will generate the results automatically by using the cosine similarity algorithm. We have disabled the switching between the applications so that student can’t open some other tab while doing his exam if they switch between tabs the quiz will be automatically terminated and mark the student as zero. The students get notification for their quiz which is uploaded by the teacher. The student can monitor the quiz whether it’s checked or not via application. Teacher can also evaluate and can give marks and comments. The results of the student will be shown to them after the deadline.

## **1.8 Methodology and Software Lifecycle for This Project**

**A close up of text on a white background

Description automatically generated**

**Figure 1.1: Methodology**

* We would be using spiral model as our software process model because using this model the system can evolve much according to the needs of users. The system has a powerful approval and report control. Moreover, extra functionality can be put in at a later date.
* The most top best function of the model is its capacity to hold unknown risks after the software has begin; making a prototype makes this achievable.

### **1.8.1 Rationale behind Selected Methodology**

* We select this methodology because we are using web based approach as we are using different tools. Also web based approach provides reusability of functions which would eliminate the risk of redundancy.
* Through this the development is quick and highlights are included an efficient way.
* In spiral model cost was estimated easily because the prototype is done in small fragments. It also helps in risk management.

**Chapter # 2**

# **Problem Definition**

## **Problem Statement**

As technology evolution is getting fast day by day so people are getting more dependent on technology. Technologies are moving toward Wireless World. Everyone wants work to be done by just one click. Due to tiring routines in daily life people want to operate their mobile phone in their homes.

Estimating the similarity between words, sentences, text and records is a vital part in different tasks, for example, data recovery, document clustering, short answer grading, machine translation and text summarization. A ton of time and cash are squandered since students need to move starting with one spot then onto the next to give tests. Time and man power is also consumed as setting and checking of the tests is done physically. The purpose of Smart Quiz Manager is to take online quiz in a well organized manner and no waste of time for checking the paper. The main purpose of Smart Quiz Manager is to properly grade the candidates through a completely mechanized framework that spares parcel of time as well as gives quickly and fulfill capable outcomes.

## **DELIVERABLES**

* **WEB BASED APPLICATION** A web based application will allow teacher to create a class. The students will join that class. Students will submit their quiz and our application will check that quiz using the cosine similarity algorithm.
* **CENTRAL XAMPP DATABASE SERVER** A central database server based on XAMPP which is the pivot point for all the communications in this system.
* **Project Report** A complete Project document that have Software Requirements Specification, Software Design Specification, GUI Mockups, Test Cases, and other major work that will performed by us.

## **Development Requirements**

Following are the requirements which the user of the system must fulfill in order to run the system on their laptops or PCs.

### **OS Requirements**

Windows 7, Windows XP, Windows 8, Windows 8.1, Windows 10

### **DBMS Requirements**

Sql Server 2012, Php, Connectivity of the application with the server.

**2.3.3 Other Requirements**

For Mockup and presentation we use MS Word, MS Power Point, and Just in Mind

## **Related System Analysis/Literature Review**

One platform which can cater to all types of online exam needs and. ln-built mechanism which restricts participants from typical exam malpractices like copying from others or carrying cheats-sheets .In case of accidental termination of the tests, system allows continuing from the point of discontinuation. All major types of questions can be created.

## **How our project different from existing applications?**

Because this application is used to conduct online examination which is not commonly used in universities, schools and colleges. The students should sit at individual terminals and login to system to write the exam in the given duration time. . The questions must be given to the students.

The students first register his/her details and then make the account after that student is able to login the system.

When the user login the system then he want to connect with the faculty module before login student is not able to connect with faculty module.

The user home page is contain two buttons one is for connect and the second is for the start exam so first student is connected to the faculty module when student is connect now he is not allow for the paper without the permission of the instructor when the teacher allow the student than the start exam button is enable on the permission of teacher.

On starting a paper first part is MCQs and in this section when they choose the option and then go to the next than the previous objective become lock after the objective part there is the subjective part in which include questions.

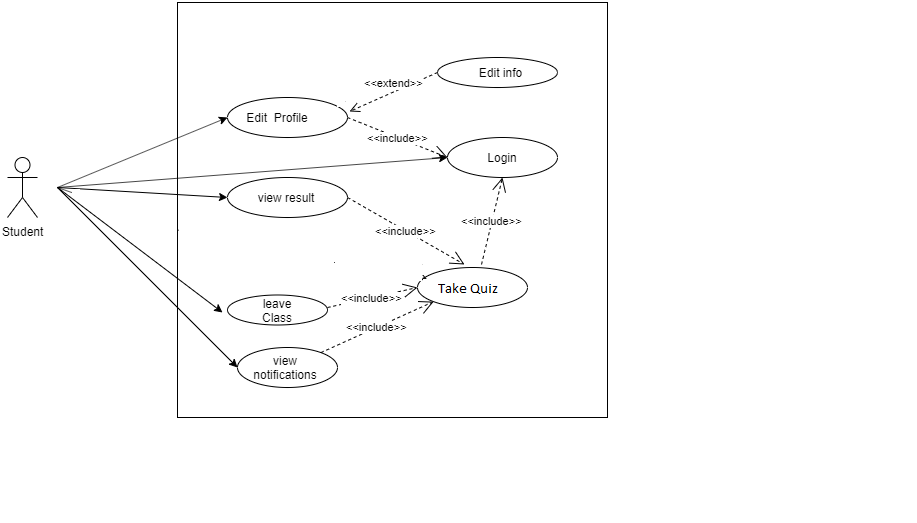
After clicking on the submit button the quiz is submitted and the system will generate the results automatically by using the cosine similarity algorithm. We have disabled the switching between the applications so that student can’t open some other app while doing his exam.

**Chapter # 3**

# **Requirements Analysis**

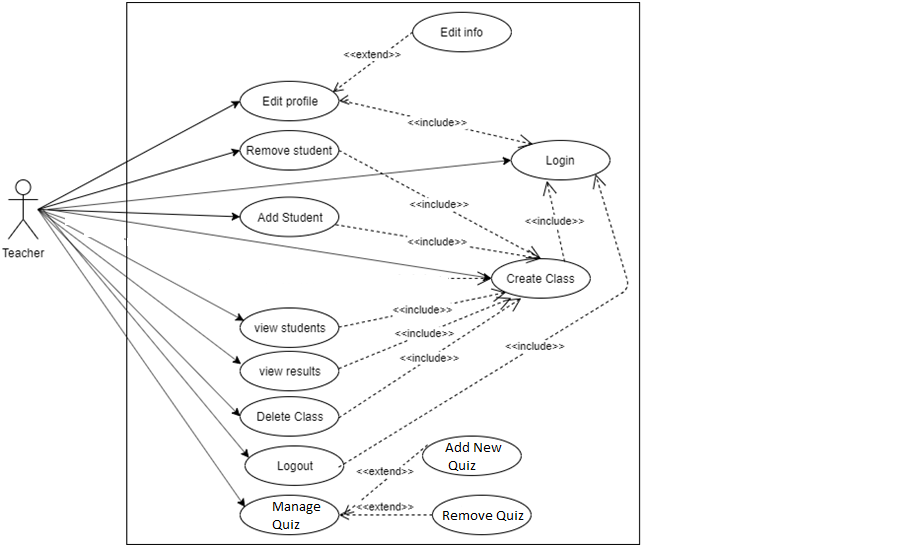
## **Use Cases**

### **Student:**



**Figure 3.1: Student Usecase Diagram**

### **Teacher**



**Figure 3.2: Teacher UseCase Diagram**

## **Detailed Use cases**

**Use** **case** **name**: Login

**Priority**: It has to be done first.

**Actors**: Student, Teacher

**Precondition**: Actor may use application for first time.

**Post-Condition**: If the use case was successful, the actor is now logged into the system. If not the system state is unchanged.

**Extend**: none.

**Uses**: Create Class, join class.

**Normal course of event**: This use case starts when an actor wishes to log into the Course Registration System.

1. The system requests that the actor enter his/her name and password.
2. The actor enters his/her name and password.
3. The system validates the entered name and password and logs the actor into the system.

**Alternative** **path**: **Invalid Email / Password**  
If in the *Basic Flow*the actor enters an invalid email and/or password, the system displays an error message. The actor can choose to either return to the beginning of the *Basic Flow*or cancel the login, at which point the use case ends[6s].

**Use** **case** **name**: Create Class

**Priority**: 2

**Actors**: Teacher

**Precondition**: User must be logged in.

**Post-Condition**: after that teacher can add quiz to their class.

**Extend**: none

**Uses**: remove student, add student, view students, view students, view results, delete class, manage quizez.

**Normal course of event**: This use case starts when an actor wishes to create a class.

1. The system requests that the actor enter id, name and password.
2. The actor enters class’s id, name and password.
3. The system validates the entered id,name and password format and create the class.

**Use** **case** **name**: Edit Profile

**Priority**: none

**Actors**: student, teacher

**Precondition**: users must be logged in.

**Post-Condition**: can change their names and password, etc.

**Extend**:

**Uses**:

**Normal course of event**: This use case starts when an actor wishes to edit his profile.

1. The system requests that the actor must be logged in.
2. The actor choose the info which he want to change e.g. name, password.
3. The system validates the entered info and password format and edit the profile.

**Use** **case** **name**: Manage quiz

**Priority**: 3

**Actors**: Teacher

**Precondition**: class must be created first.

**Post-Condition**: student will be able to submit it.

**Extend**: Delete quiz, Add new quiz.

**Use** **case** **name**: submit quiz

**Priority**: 4

**Actors**: Student

**Precondition**: there must be a quiz by teacher to submit.

**Post-Condition**: system will calculate the result.

**Use** **case** **name**: add student

**Priority**:4

**Actors**: Teacher

**Precondition**: class must be created and student must be register.

**Post-Condition**: student will be part of the class.

**Use** **case** **name**: remove student

**Priority**: 5

**Actors**: teacher

**Precondition**: student must be part of that class.

**Post-Condition**: student will be no longer part of that class.

**Use** **case** **name**: leave class

**Priority**: 6

**Actors**: student

**Precondition**: actor must be part of class.

**Post-Condition**: actor will be no longer part of that class.

**Use** **case** **name**: Delete class

**Priority**:6

**Actors**: teacher

**Precondition**: actor must be owner of class.

**Post-Condition**: the class will no longer exist.

**Use** **case** **name**: view result

**Priority**: 6

**Actors**: student, teacher

**Precondition**: Quiz must be submitted and the result will be shown after due date.

**Post-Condition**: none

**Extend**: nones

**Uses**: create class.

**Use** **case** **name**: view students

**Priority**: 6

**Actors**: teacher, student

**Precondition**: teacher not know how many students are in class.

**Post-Condition**: teacher will come to know that how many students are in class.

**Use** **case** **name**: logout

**Priority**: 7

**Actors**: student, teacher,

**Precondition**: actor want to logout

**Post-Condition**: actor will no longer be logged in to application.

## **Requirements**

A complete description of what the software system will do without describing how it will do it is represented by the **software requirements[3]**. A condition or capability that must be met or possessed by a system to satisfy a contract, standard, specification or other formally imposed documents[4].

### **Functional requirements**

Should be complete and consistesnt. Customer and developers usually focus all their attention on functional requirements. Statements of services the system should provide:

* Reaction to particular inputs
* Behavior in particular situations [1]

Following are functional requirements:

#### **Login**

* Student shall login via Email and password.
* Teacher shall login via email and password.

#### **View Profile**

* Logged in student can view their own Profile.
* Logged in Teacher can view his own Profile.

#### **Edit profile**

* Logged in Student and teacher can edit their own Profile like, name, password, etc.

#### **Add Quiz**

* Logged in Teacher will add a new quiz in class. The quiz shall have its start date, due date and result date(at which the similarity index will be shown to all students) that will be shown by all students. And also can set rules for quiz.

#### **Upload Quiz**

* Logged in Student can upload quiz after joining a class at after the allow date for quiz in the allowed format and file type.

#### **View Students**

* Teacher can view their registered class students.

#### **Add Students**

* Logged in Teacher will add students who has an account.

#### **View Result**

* Logged in Students will view their own result.
* Logged in teacher can view result of all students in his class.

#### **Manage Quiz**

* Logged in teacher can add or remove any quiz from his class and can see recent and upcoming quiz.

#### **Remove student**

* Logged in Teacher can remove any student from class.

#### **Leave Class**

* Logged in Student can leave class.

#### **Create Class**

* Logged in teacher will create class with a class id , name and password.

#### **Join Class**

* Logged in student shall join class by class id and password.

#### **Delete Class**

* Logged in teacher can delete his class.

#### **Logout**

* All Users can logout.

### **Non-functional requirements**

These requirements specify the criteria that can be used to judge the operation of a system.

Rather than specific behavior [2]. It ensures the quality of software. It essentially specifies how

The system should behave and the constraints upon the system [5].

**Security requirement:**

Safety is essential for a software system. Unregistered user cannot access the system. Safety

requirement is maintained through providing individual account to each user.

**Reliability:**

User can undoubtedly give their data it’s secure application.

**Maintainability:**

Errors can be easily handled. In future if anyone want to modify it, the changes can easily be accommodated.

**Reusability:**

It is reusable and used at anywhere. Later on code and design can also be reused to extend this project. The web panel is made using Node js etc. Which are stage autonomous and can be moved to Different workers with least exertion.

**Performance:**

The checking quiz text similarity index is very time taking job and can fail the system so that’s why this work will done on server so the high performance can be achieved.

**Usability:**

Our Application will be user friendly and should require least effort to operate This system must

run on multiple versions.

**Portability:**

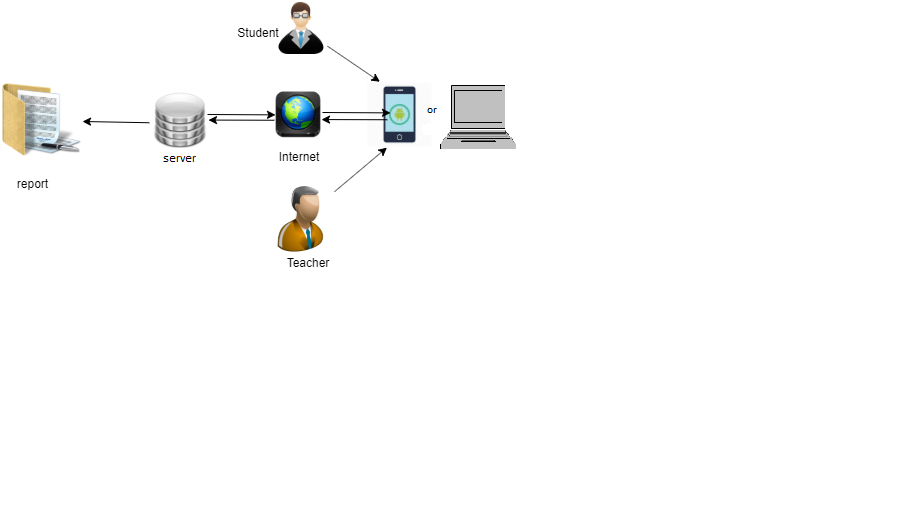
This system must run on multiple devices with almost all versions.

**Chapter # 4**

# **Design and Architecture**

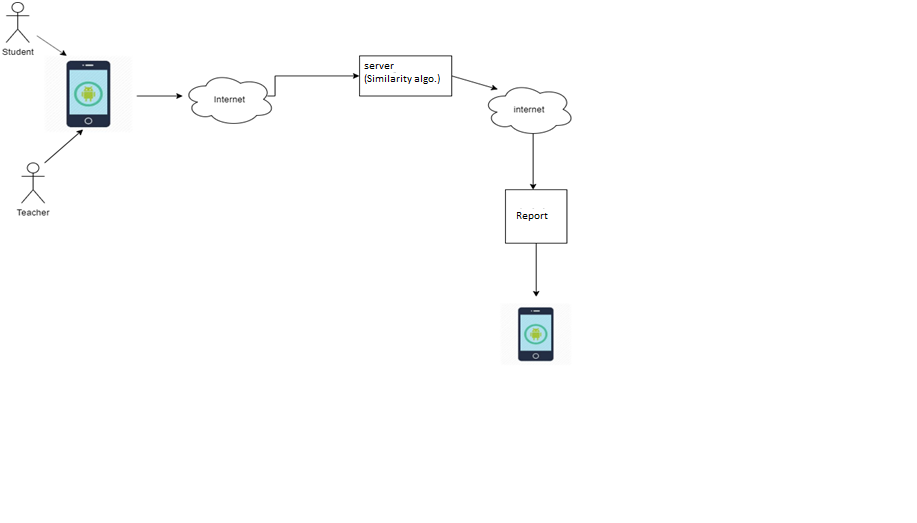
## **System Architecture**

### **How does our application works**



**Figure 4.1: Application working**

### **Block Diagram**



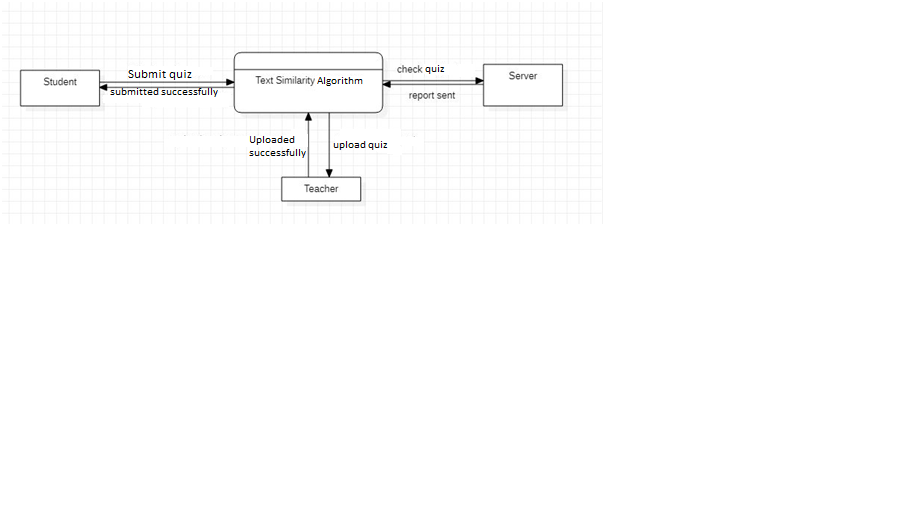
**Figure 4.2: Block Diagram**

**Description**

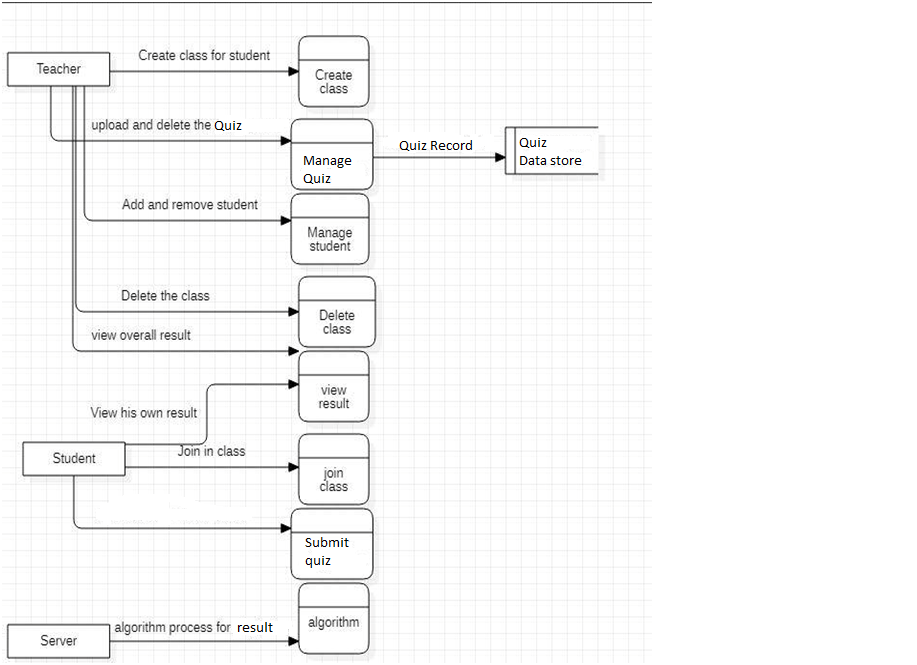
This diagram shows that how we check quiz by using our web Based application in which mobile connect with server through internet and quiz are upload in a server by a student who join a class then the server process the algorithm and report generate and send to application.

## **Data Representation**

**Data Flow Diagram (level 0)**

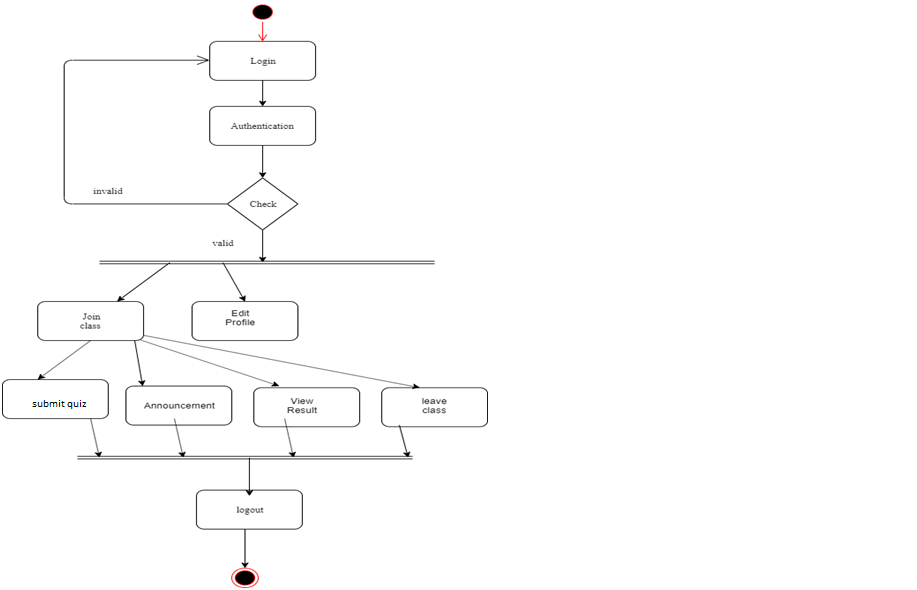


**Figure 4.3: Data Flow Diagram L1**

**Data Flow Diagram (level 1)**

**Figure 4.4: Data Flow Diagram L2**

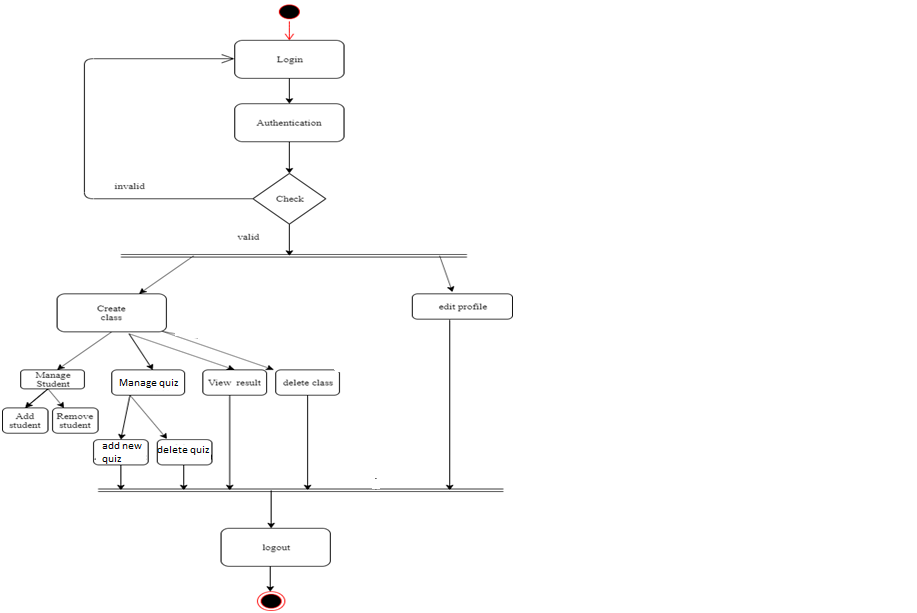
## Process Flow/Description

****

**Figure 4.5: Process Flow 1**

**Description**

This diagram is actually shows the flowchart to show the flow from one to other activity, Figure Demonstrate activity diagram performance of the Student It shows that the student login and join the class and after join the class he was able to edit his profile and to submit his quiz view result announcement and was able to leave the class as well.



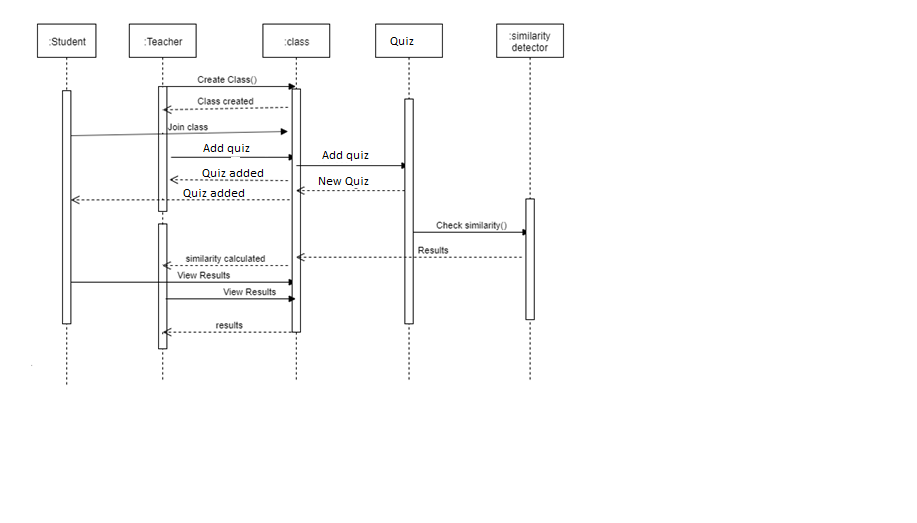
**Figure 4.6: Process Flow 2**

**Description**

This diagram shows the flowchart to show the flow from one phase to another , Figure Demonstrate the activity performance phase of the Teacher It shows that the teacher login and after authentication process create the class and after create the class he was able to edit his profile. The teacher after create his class add a quiz for student. The teacher was able to add or remove student or add or delete quiz and view result of all student and also delete the class that he will create and then the activity is end.

## **Sequence Diagrams**

**4.4.1 Overall sequence diagram:**



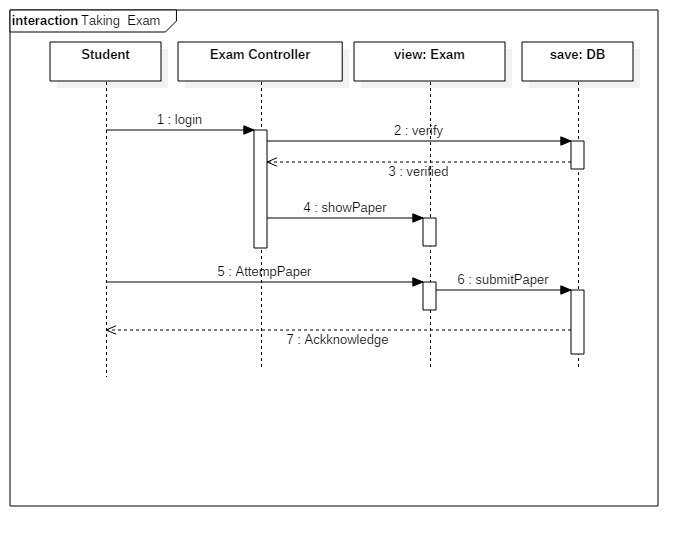
**Figure 4.7: Sequence Diagram Main**

**Description**

This diagram is interaction diagram that detail how operation are carried out. This diagram show that the order of interaction to represent time what message are sent and when. The object include in sequence diagram is Student, Teacher, class, quiz and Similarity detector. The teacher create the class and message is display that class is create the teacher add the quiz and quiz added successfully the teacher also upload new quiz the student join class and submit his quiz. The quiz is check by similarity detector and result is appear on class the student see his result.

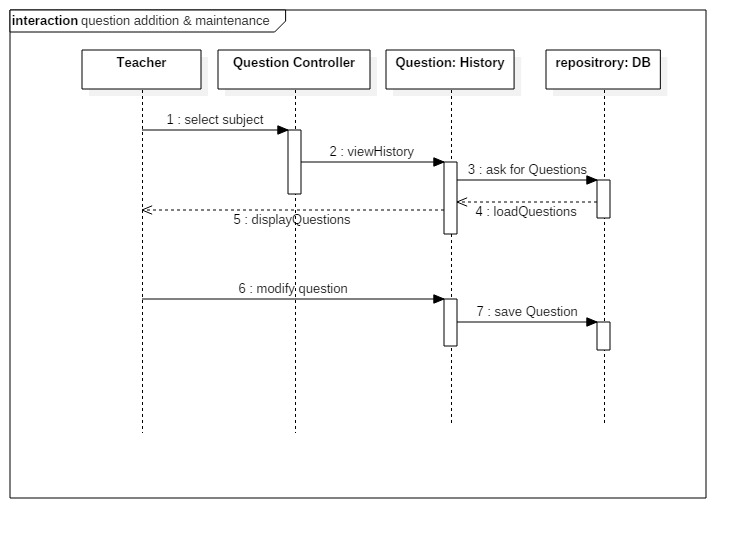
### **4.4.2** **Take a Quiz:**

This diagram shows how a student will attempt his paper with authentications. Paper will be only show if student has login with server and teacher has allowed him.



**Figure 4.8: Sequence diagram quiz taking**

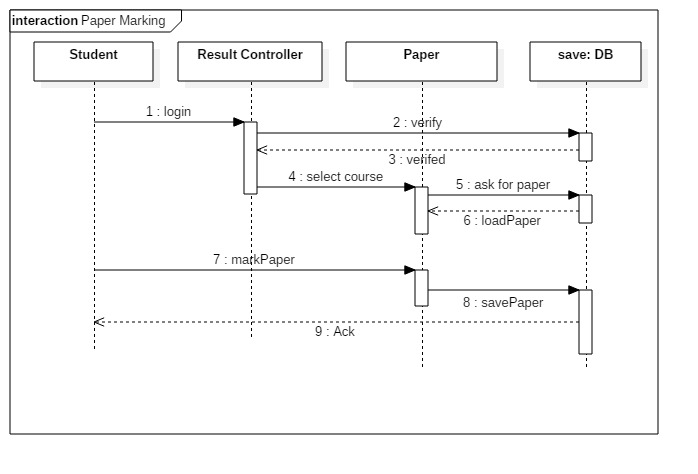
### **Question addition and modification:**

This diagram shows how will be question will add and modify into question bank. If a particular exam already has questions in question bank then teacher can modify them according to needs. 

**Figure 4.9: Sequence diagram modification**

**4.4.4 Mark Paper**

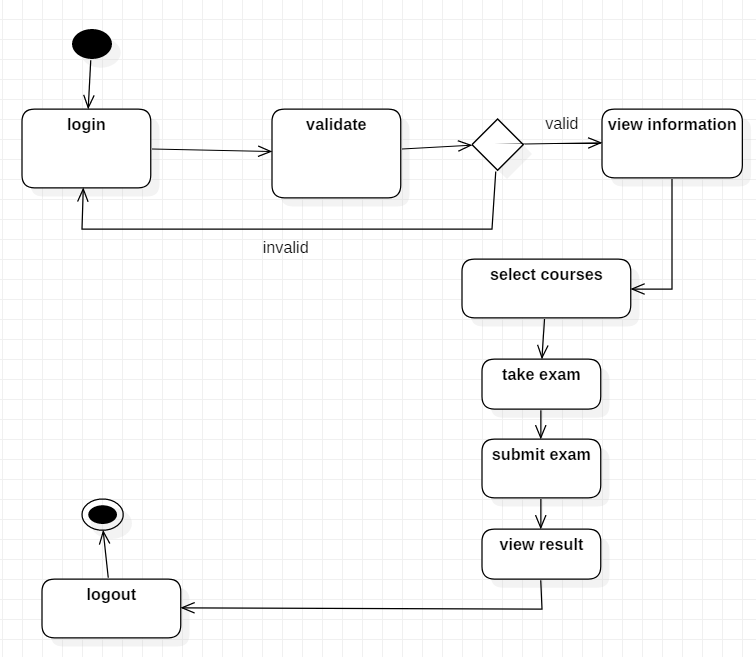
In this diagram it is explain that how a teacher will check subjective paper and save it. After marking the paper students will able to check their papers and results.



**Figure 4.10: Marking**

## **4.4.5 Activity Diagram:**

This diagram shows how will be a student attempt his quiz by login. If users name and password are correct then he/she will be able to enter otherwise he will not be able to attempt the exam.



**Figure 4.11: Activity Diagram**

**Chapter # 5**

**Implementation**

In this chapter we will discuss the implementation of our system. Here we discuss the tools and technologies that we are used to develop our project system. In this phase we convert the theoretic requirements and idea into physical system.

## 5.1 **Tools and Technologies**

Tools are essential requirement for software development, a software developer cannot write programming code without tools. Technologies are also essential for development of software. Technologies provide basic building blocks for software development.

### 5.1.1 **Tools**

* Visual Studio Code(V 1.51.1)
* Pycharm(V 192.6262.63)
* Xampp(V 7.2.0)

### 5.1.2 **Languages**

* Node.Js(V 14.5.0) (For back end coding)
* HTML CSS Bootstrap (For front end designing)

### 5.1.3 **Data Base**

* Mysql

## 5.2 **Software Requirements**

* MS Word
* MS PowerPoint

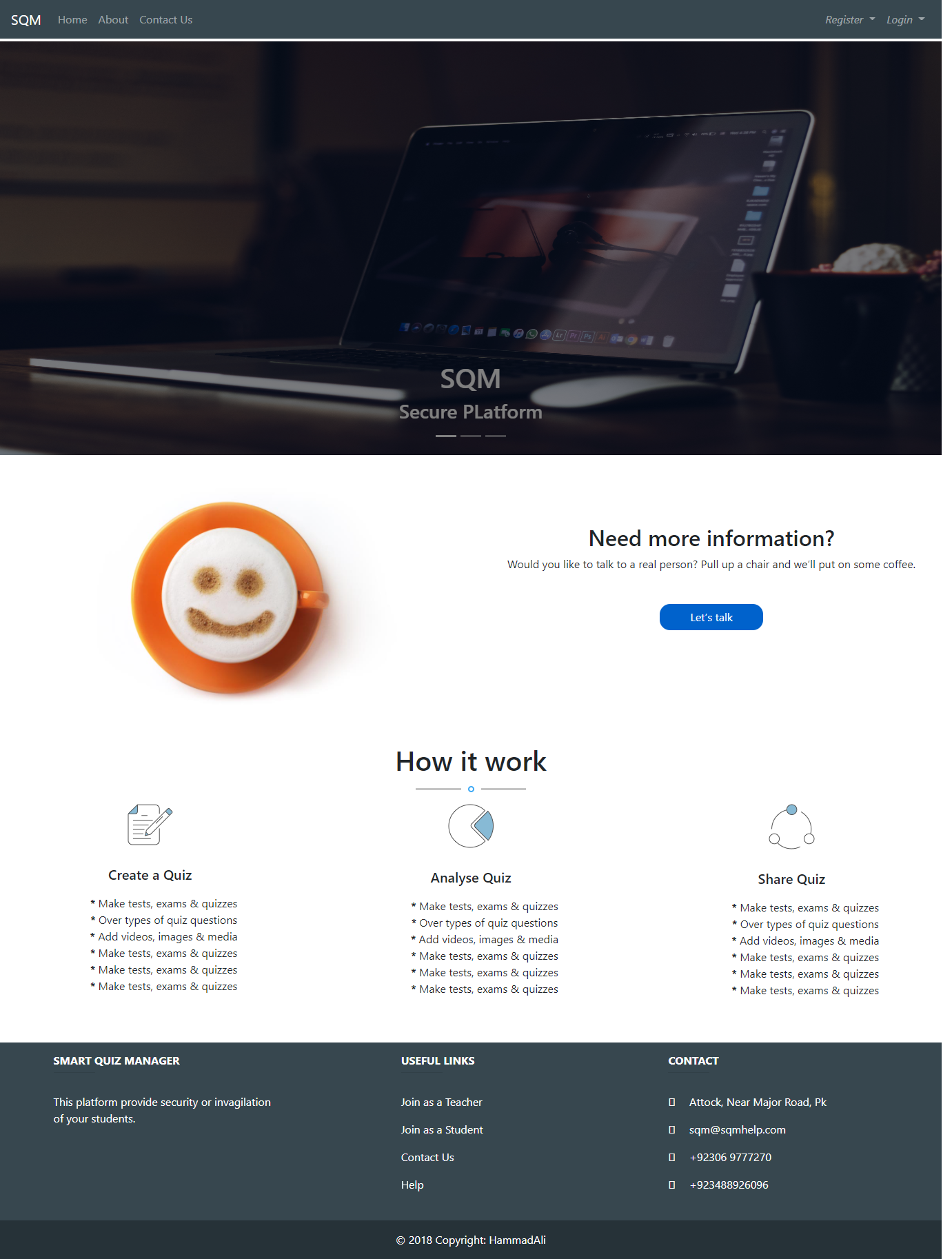
## 5.3 **Hardware Requirements**

* Modern **Operating System**:
* x86 64-bit **CPU**
* 4 GB RAM

## 5.4 **User Interface**

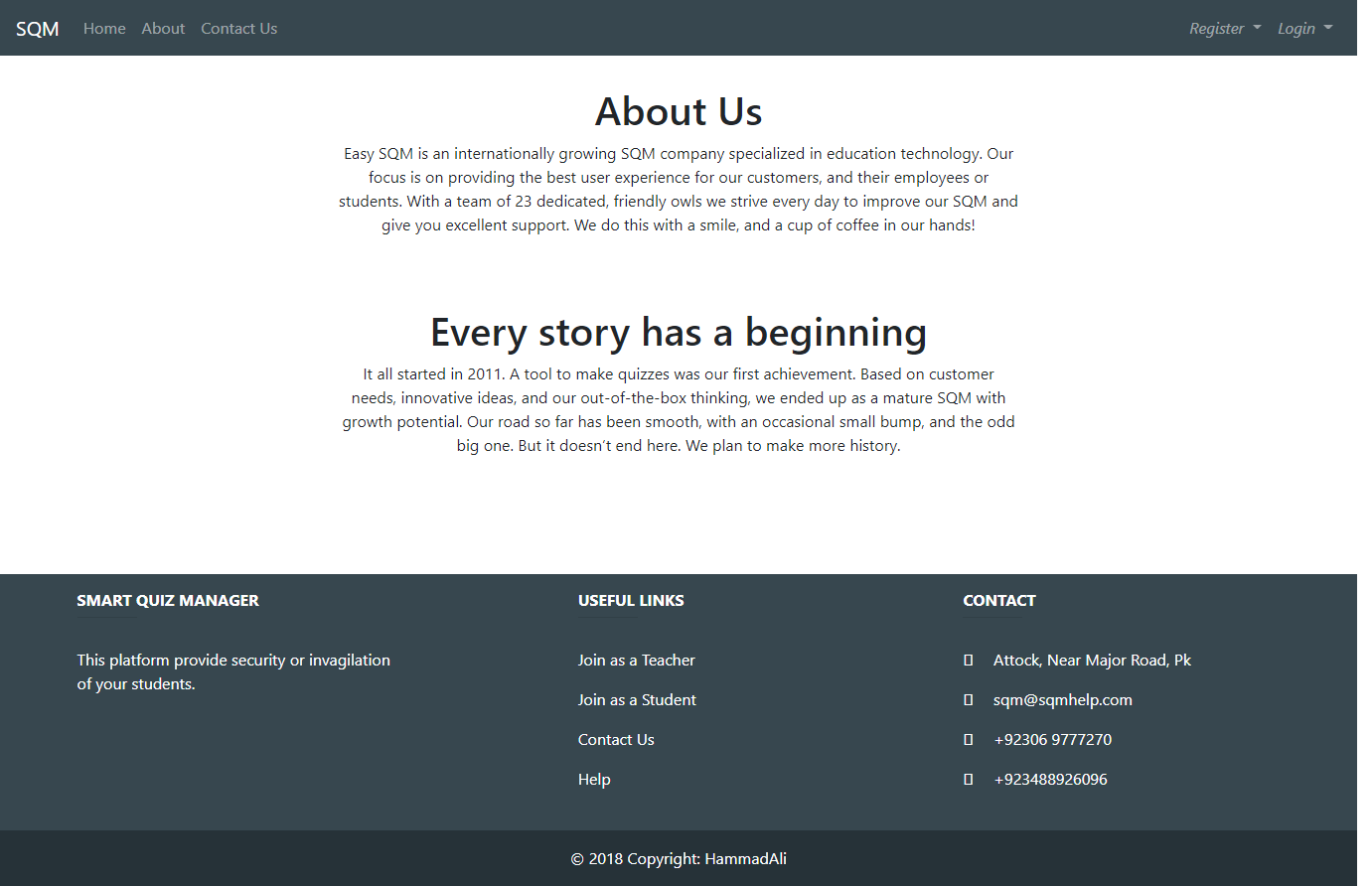
As our project is an web application so it is necessary to create interactive and understandable user interfaces, so everyone can easily use the application. It is important to develop creative and responsive screens that can target every browser. Below are the screenshots of screens of our project application.

5.4.1 **Home Page**



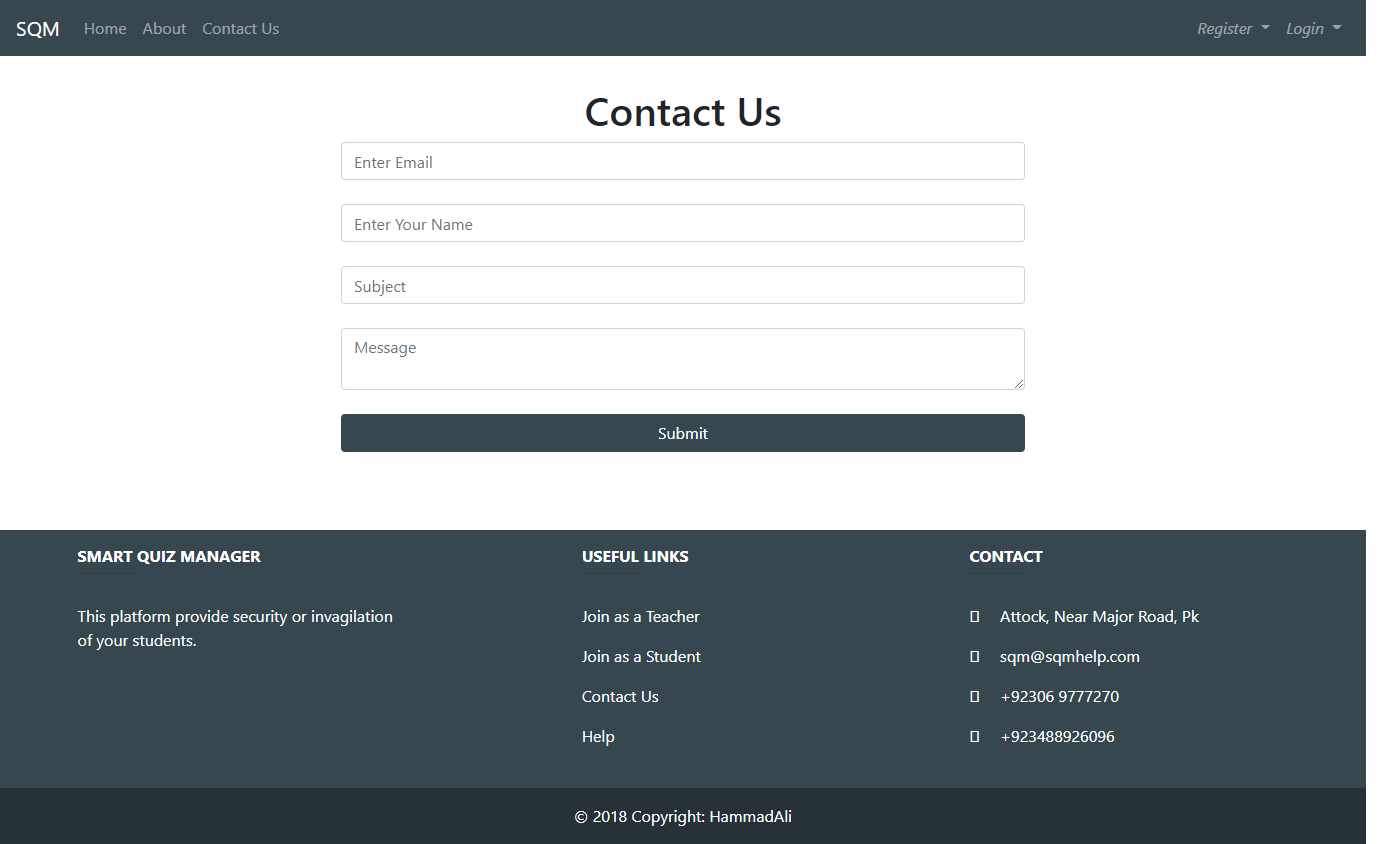
**Figure 5.1: Home Screen**

5.4.2 **About Us page**



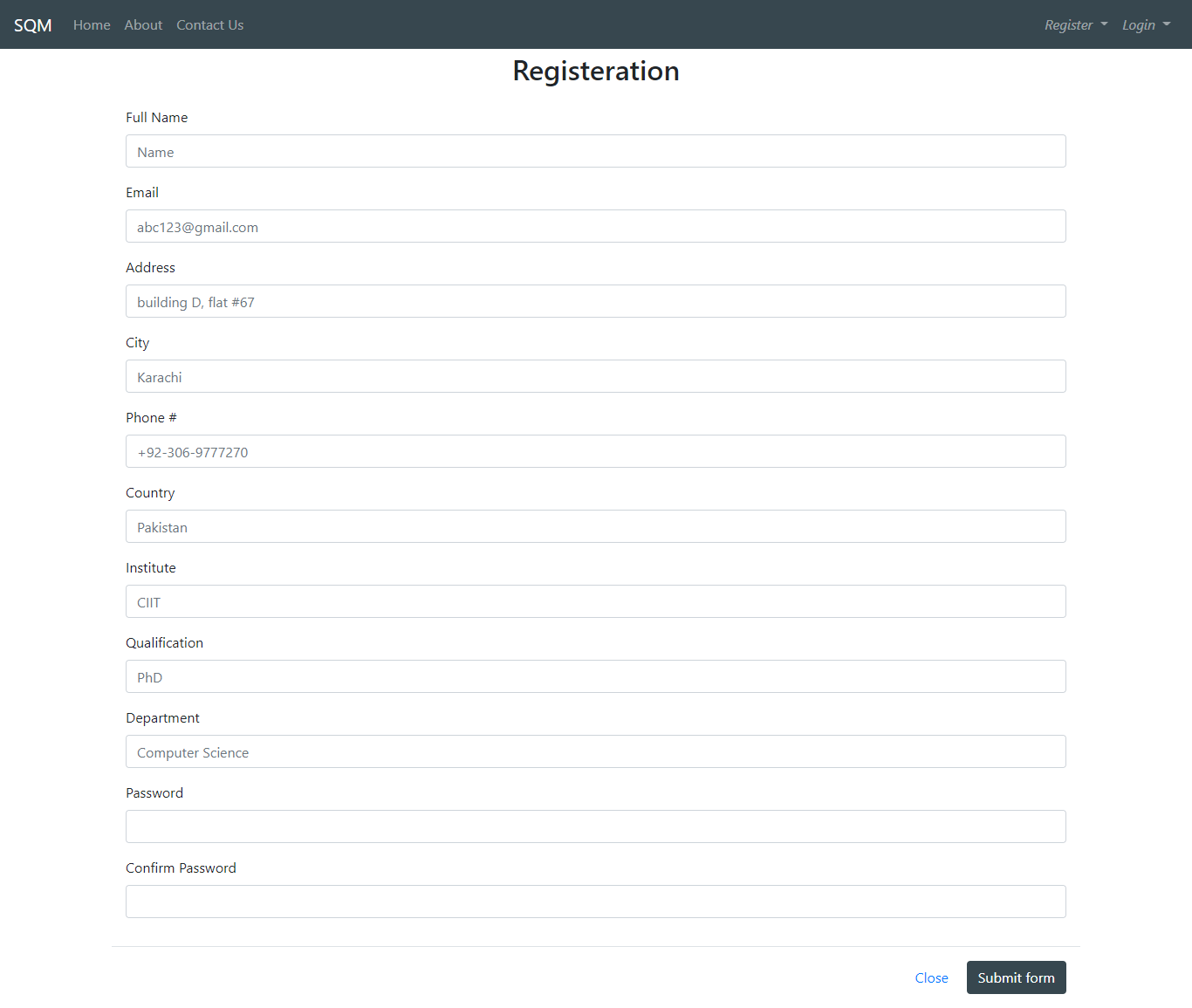
**Figure 5.2: About us screen**

**5.4.3 Feedback and Contact us**

****

**Figure 5.3: Contact us**

**5.4.4 Teachers Registration Menu**

****

**Figure 5.4: Teacher registration**

### **5.4.5** **Login Model**

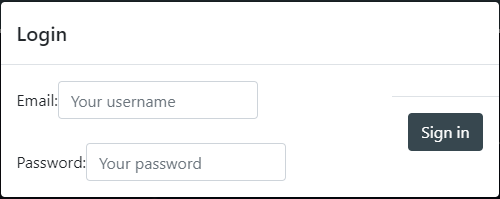


Figure 5-5 SQM Login Model

### **5.4.6 Teacher Dashboard**

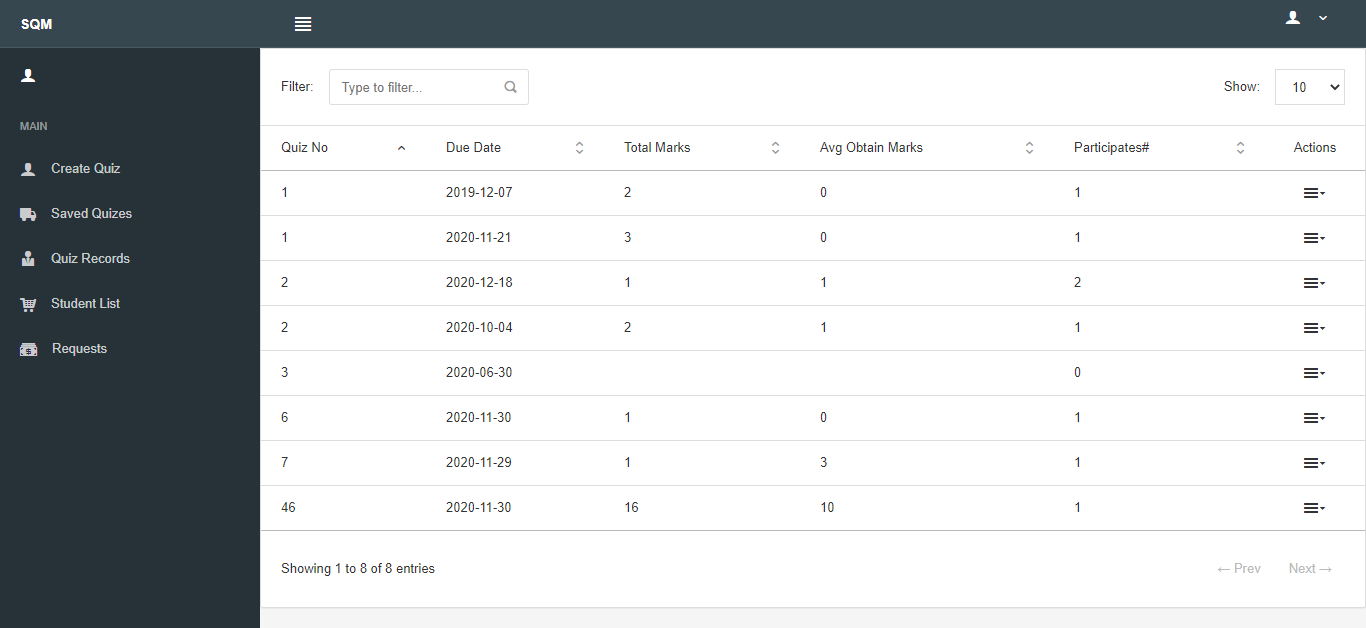


Figure 5-6 SQM Teacher Dashboard

### **5.4.7 Create Quiz**

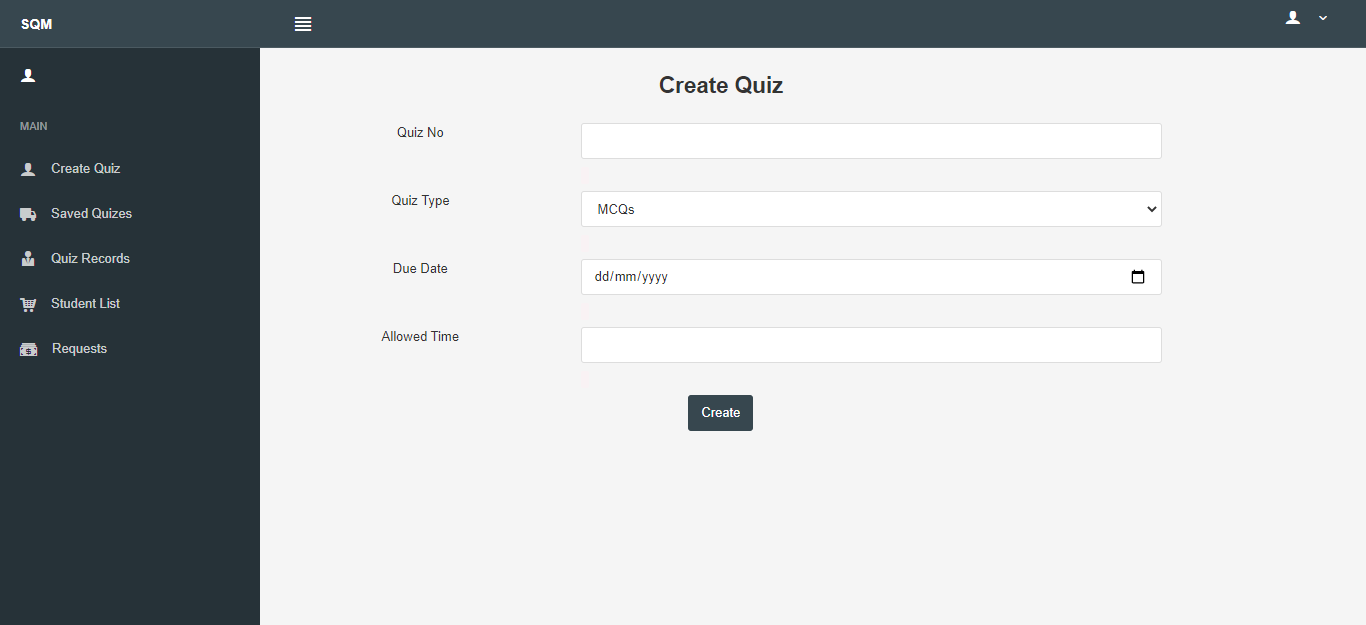


Figure 5--7 SQM Create Quiz

### **5.4.8 Quizzes Saved in Draft**

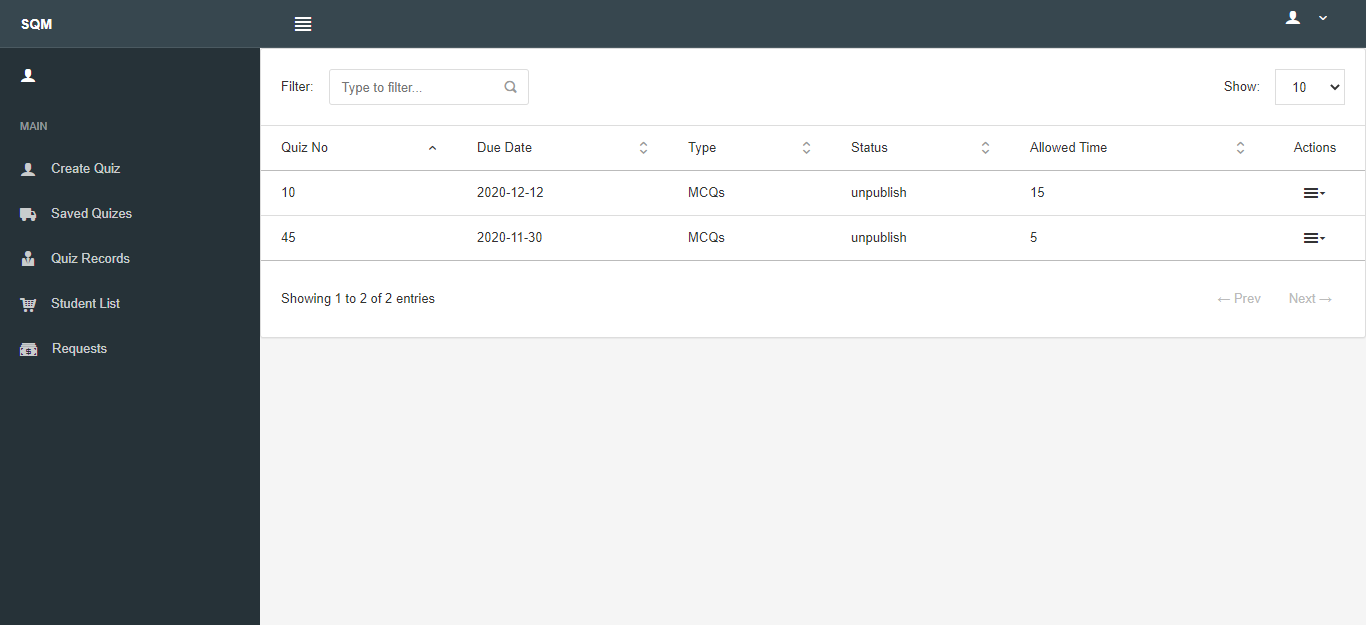


Figure 5-8 SQM Quizzes Saved in Draft

### **5.4.9 Add Question in Quiz**

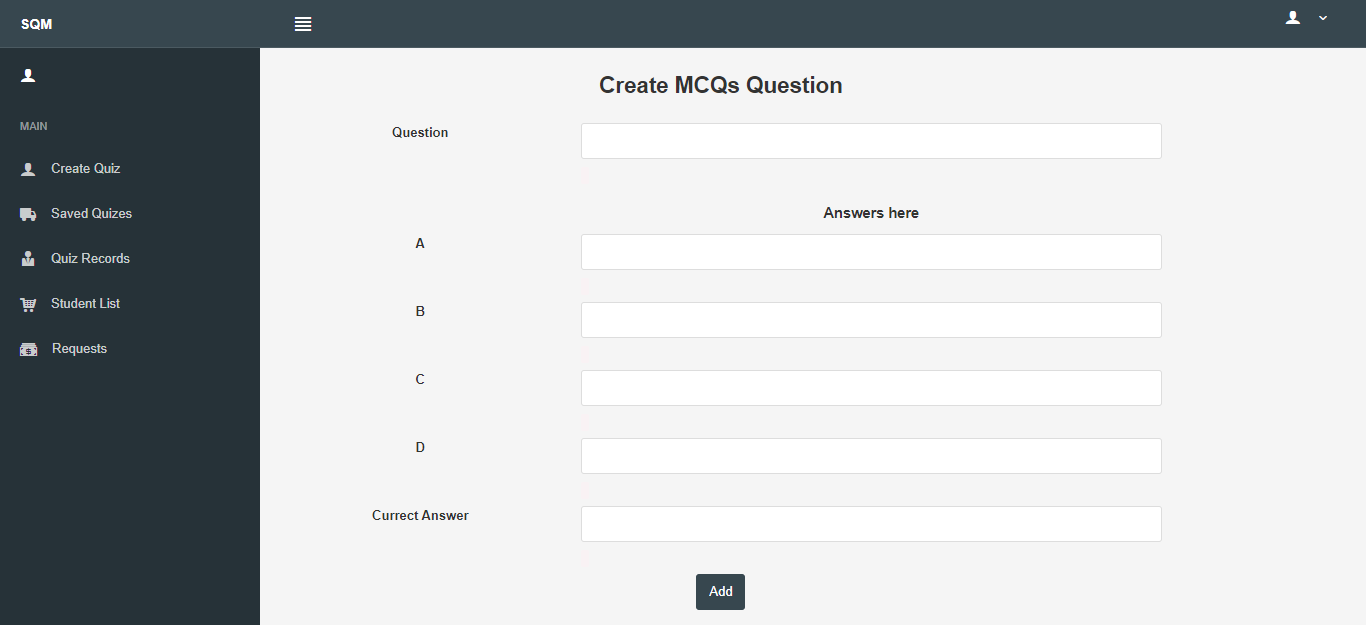
****

Figure 5-9 SQM Add Question in Quiz

### **5.4.10 Preview Draft Quiz**

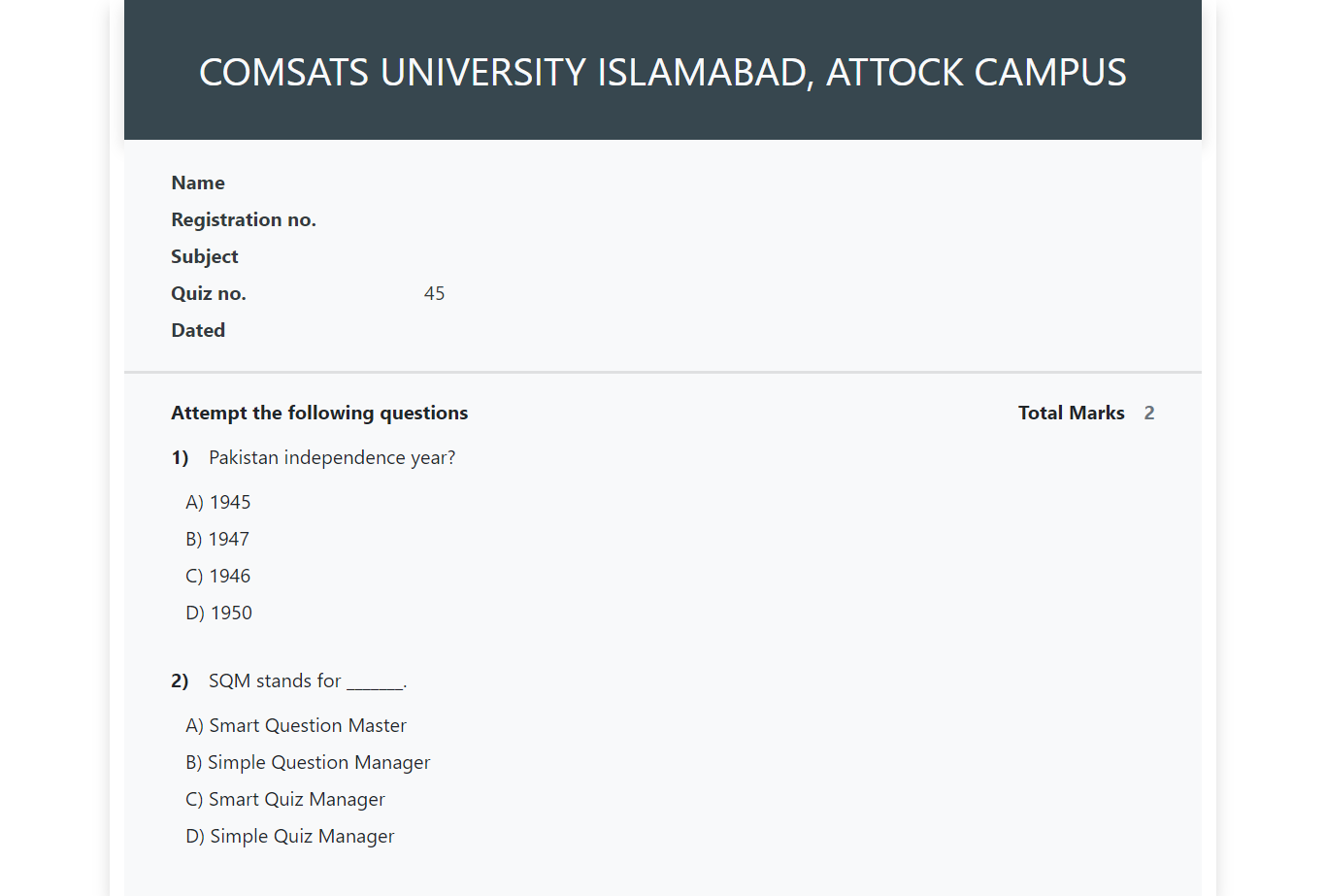


Figure 5-10 SQM Preview Draft Quiz

### **5.4.11 Create Subjective Question**

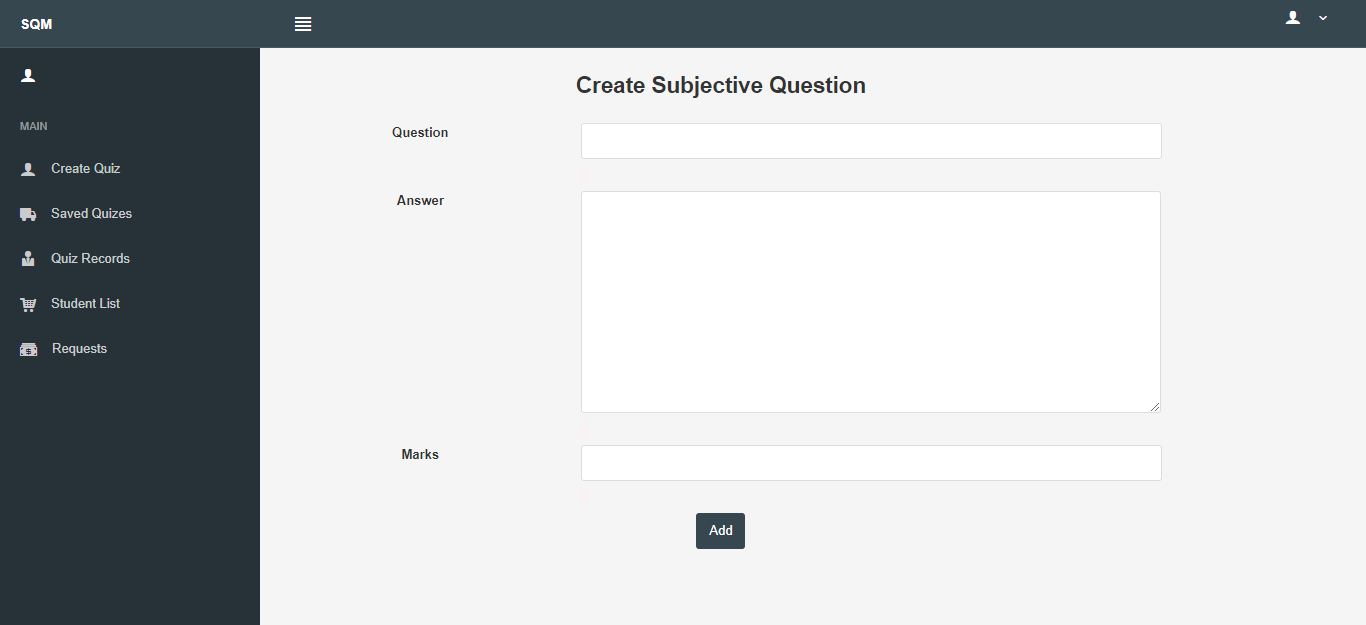


Figure 5-11SQM Create Subjective Question

### **5.4.12 All Attempted Quizzes**

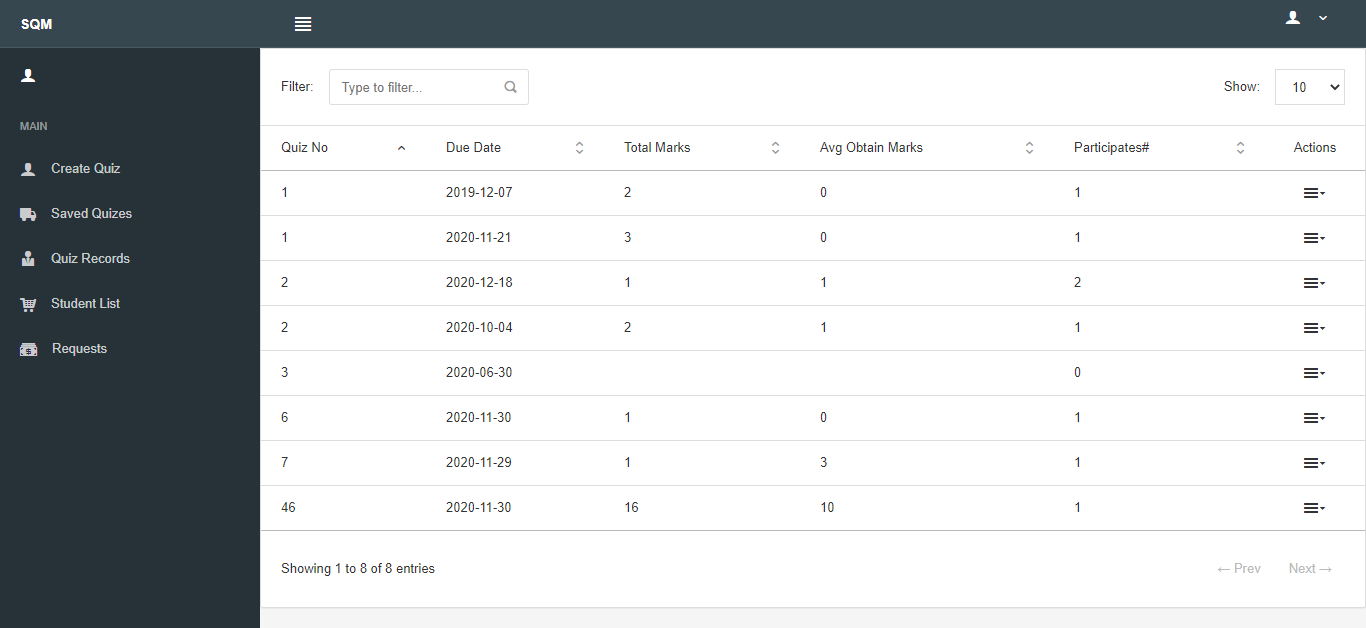


Figure 5-12 SQM All Attempted Quizzes

### **5.4.13 List of Participant**

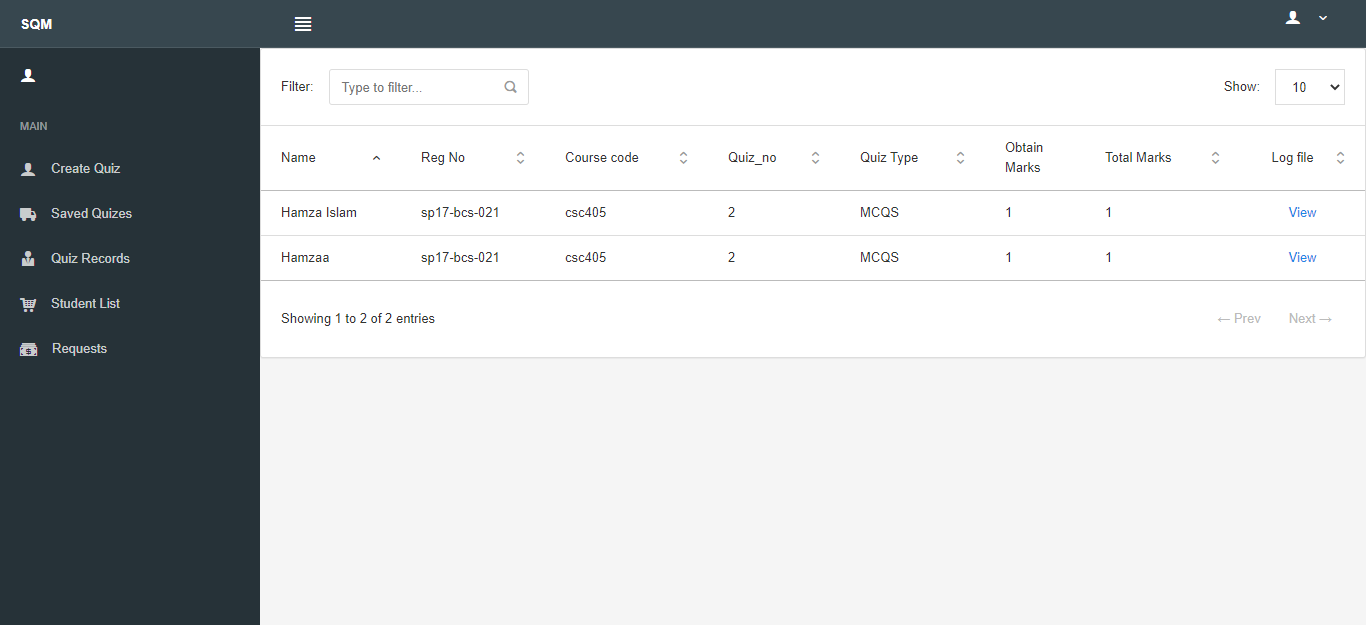


Figure 5-13 SQM List of Participant

### **5.4.14 List of Registered Students**

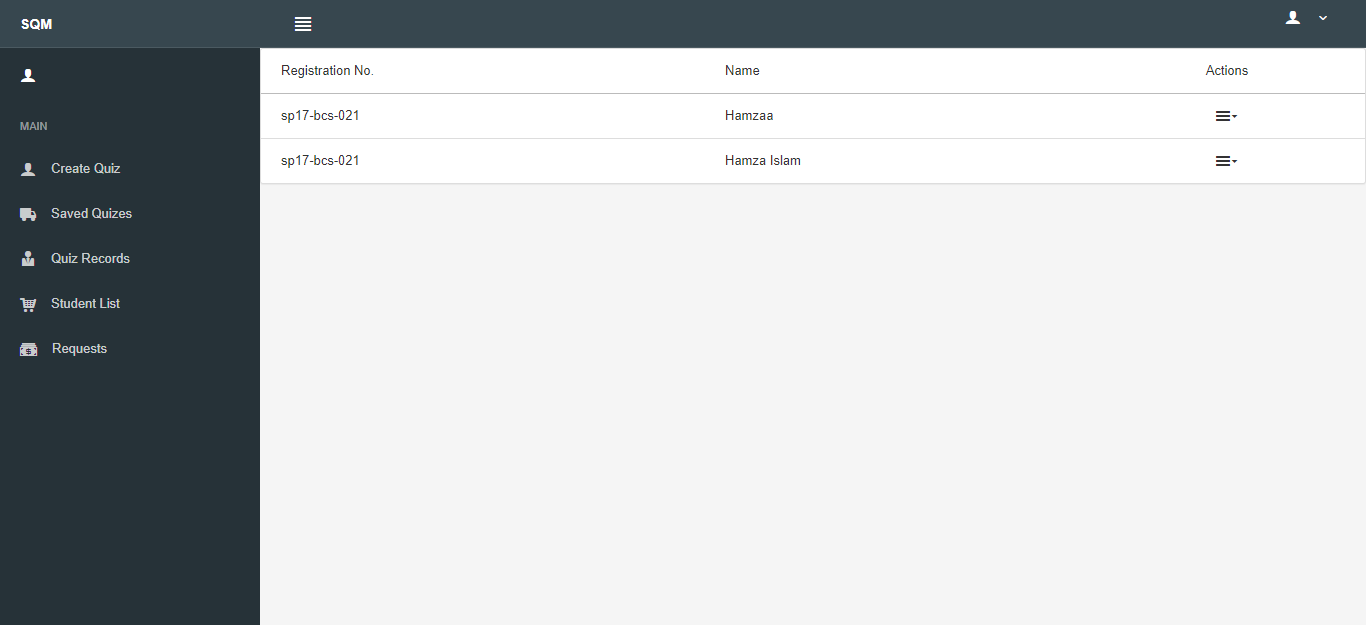


Figure 5-14 SQM List of Registered Students

### **5.4.15 Student Profile**

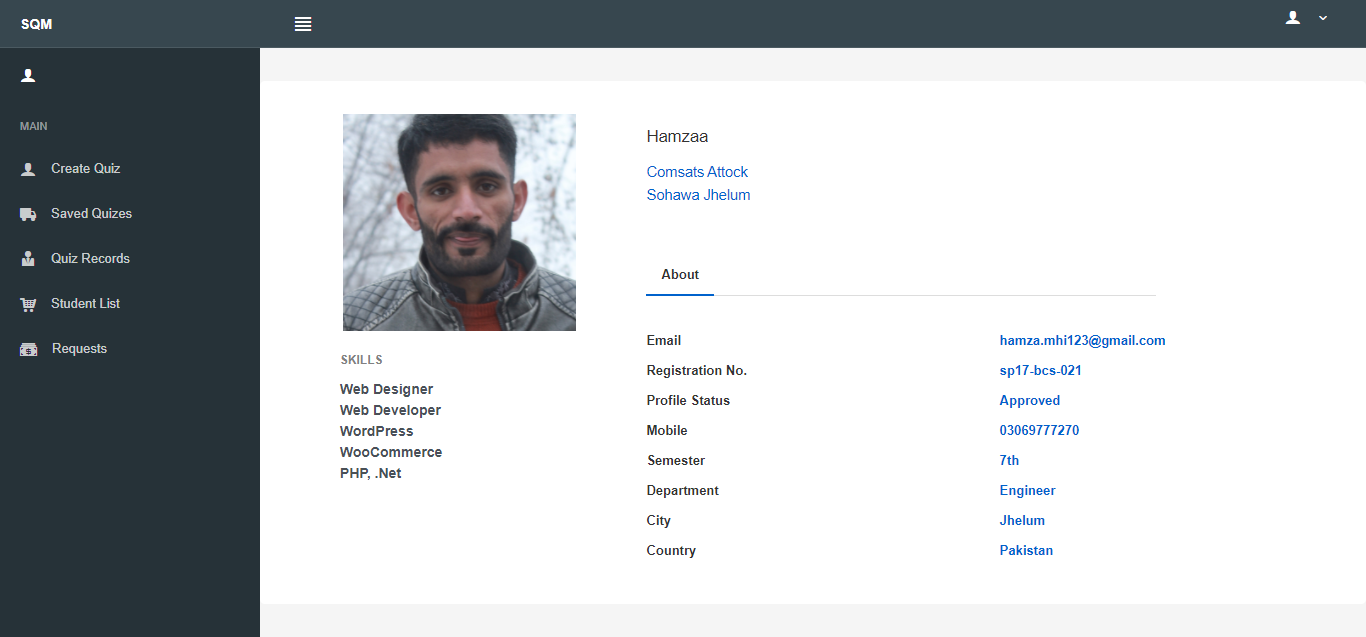


Figure 5-15 SQM Student Profile

### **5.4.16 Teacher Profile**

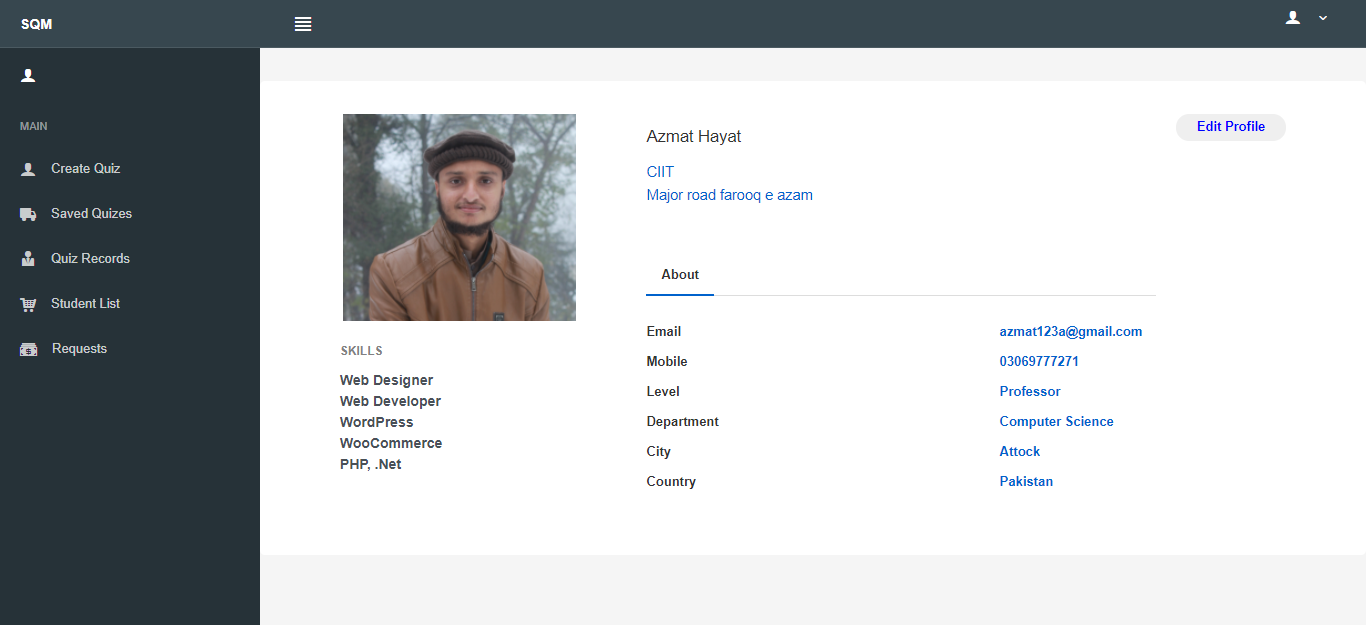


Figure 5-16 SQM Teacher Profile

### **5.4.17 Edit Teacher Profile**

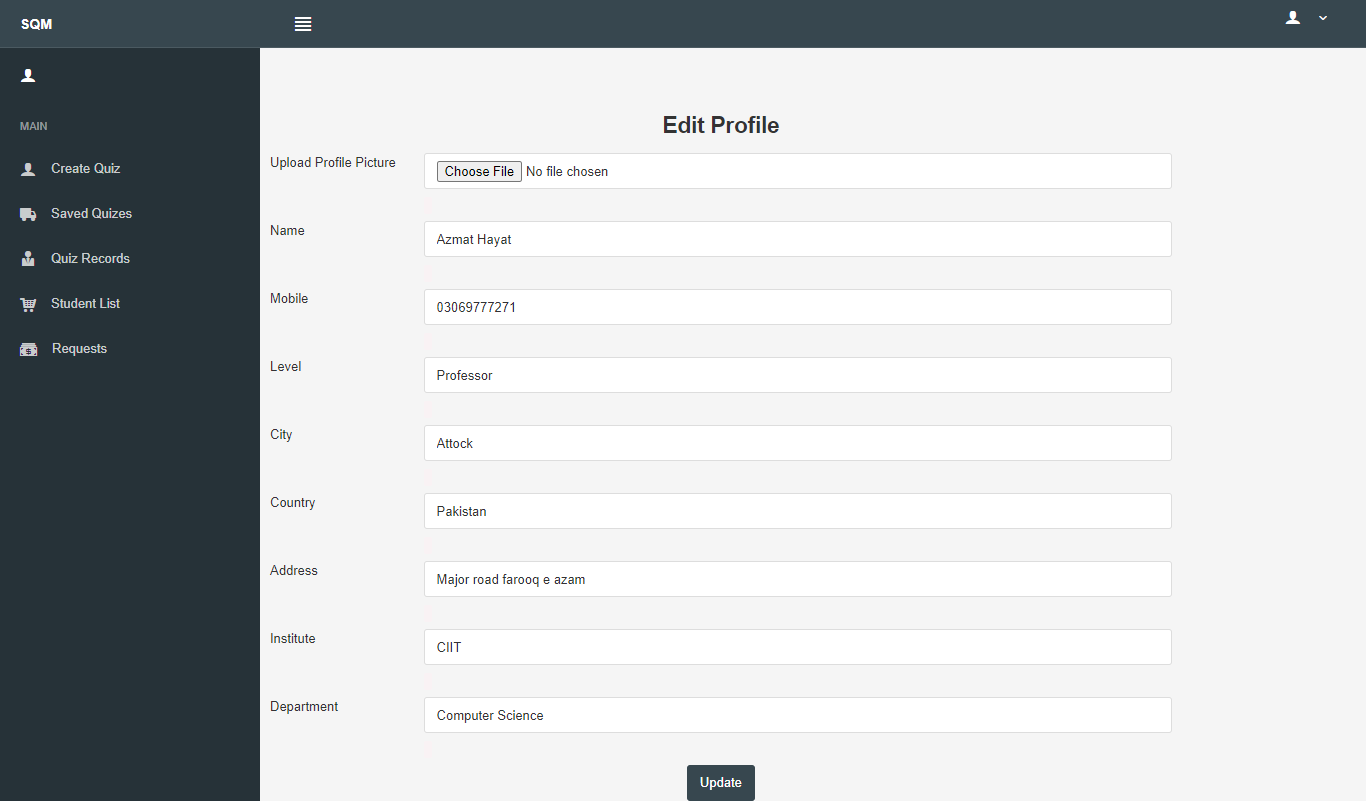


Figure 5-17 SQM Edit Teacher Profile

### **5.4.18 Request to join**

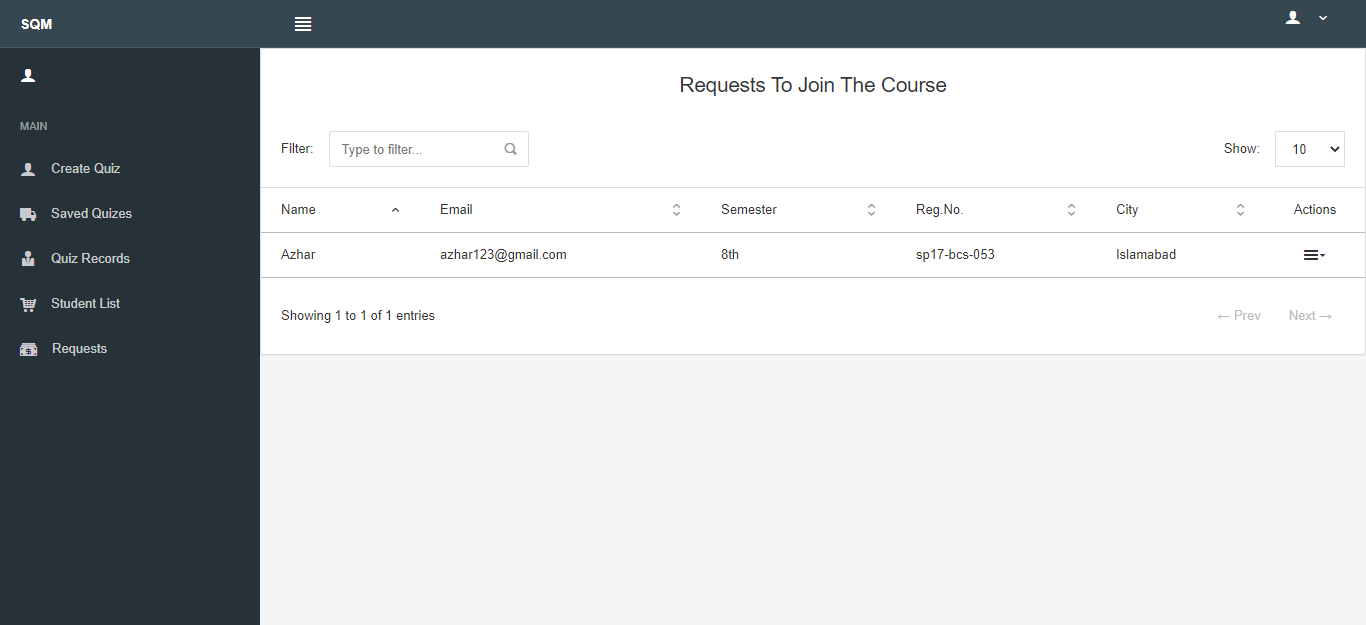


Figure 5-18 SQM Request to join

### **5.4.19 Student Dashboard**

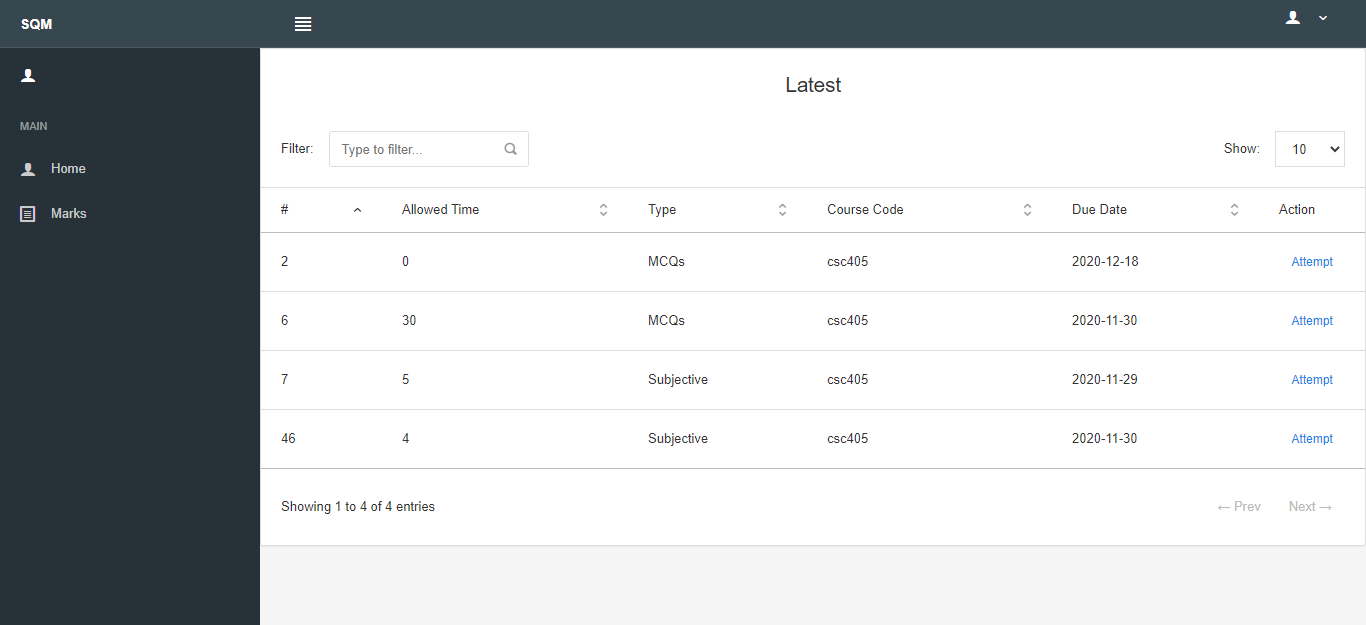


Figure 5-19 SQM Student Dashboard

### **5.4.20 Instruction Page for Student**

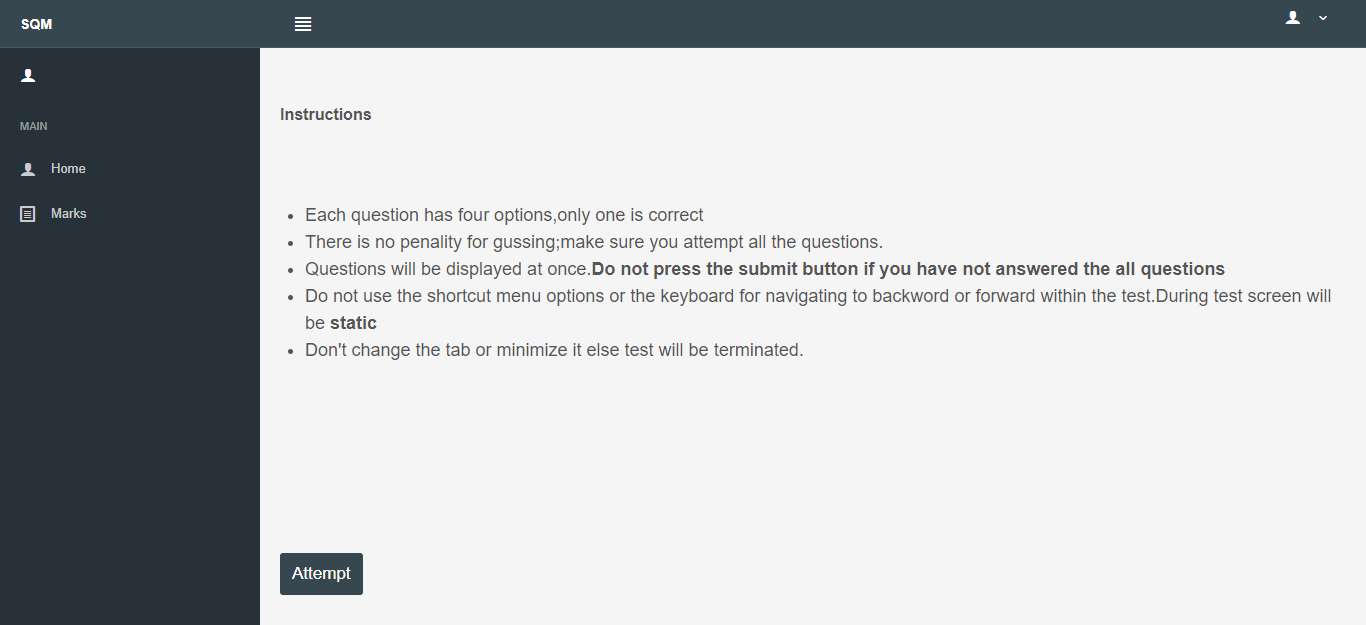


Figure 5-20 SQM Instruction Page

### **5.4.21 Verify Student Through Face Recognition**

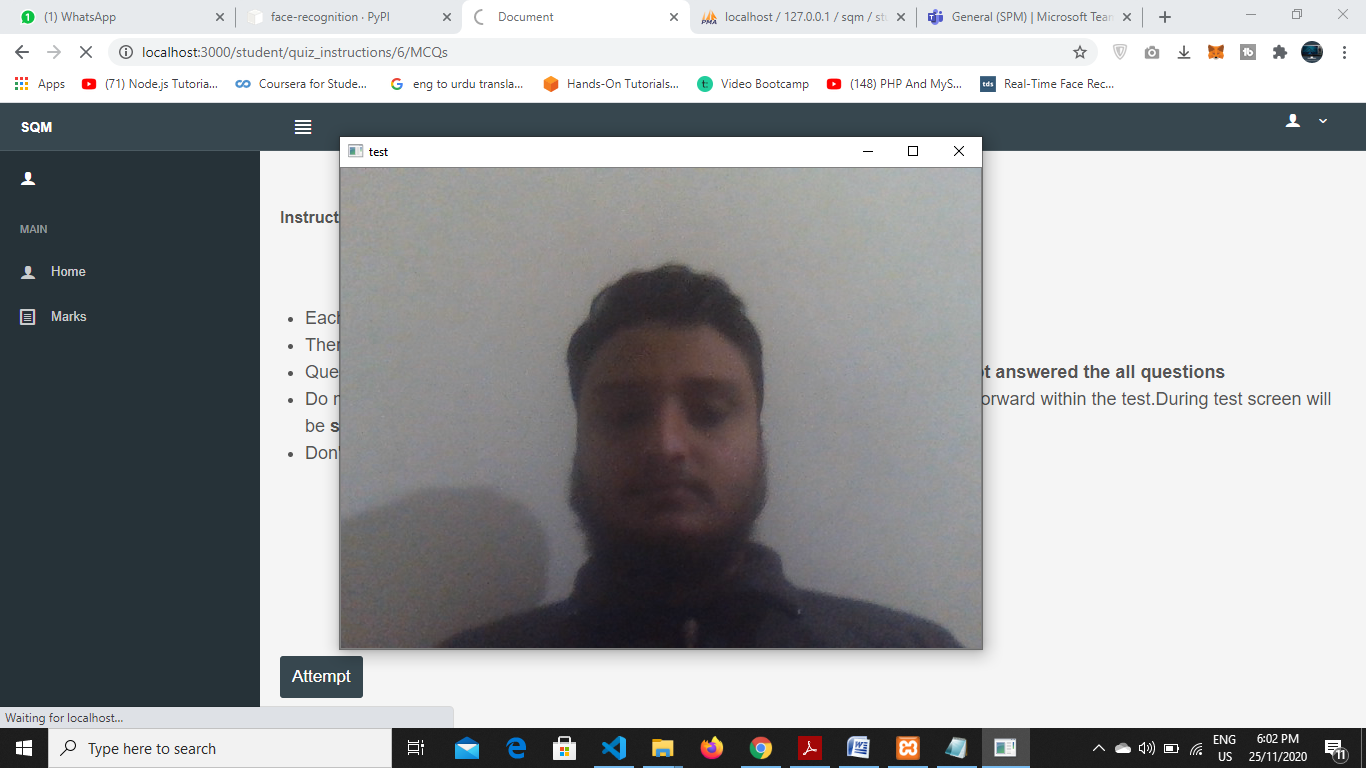


Figure 5-21 SQM Verify Student Through Face Recognition

### **5.4.22 Attempt Subjective Quiz**

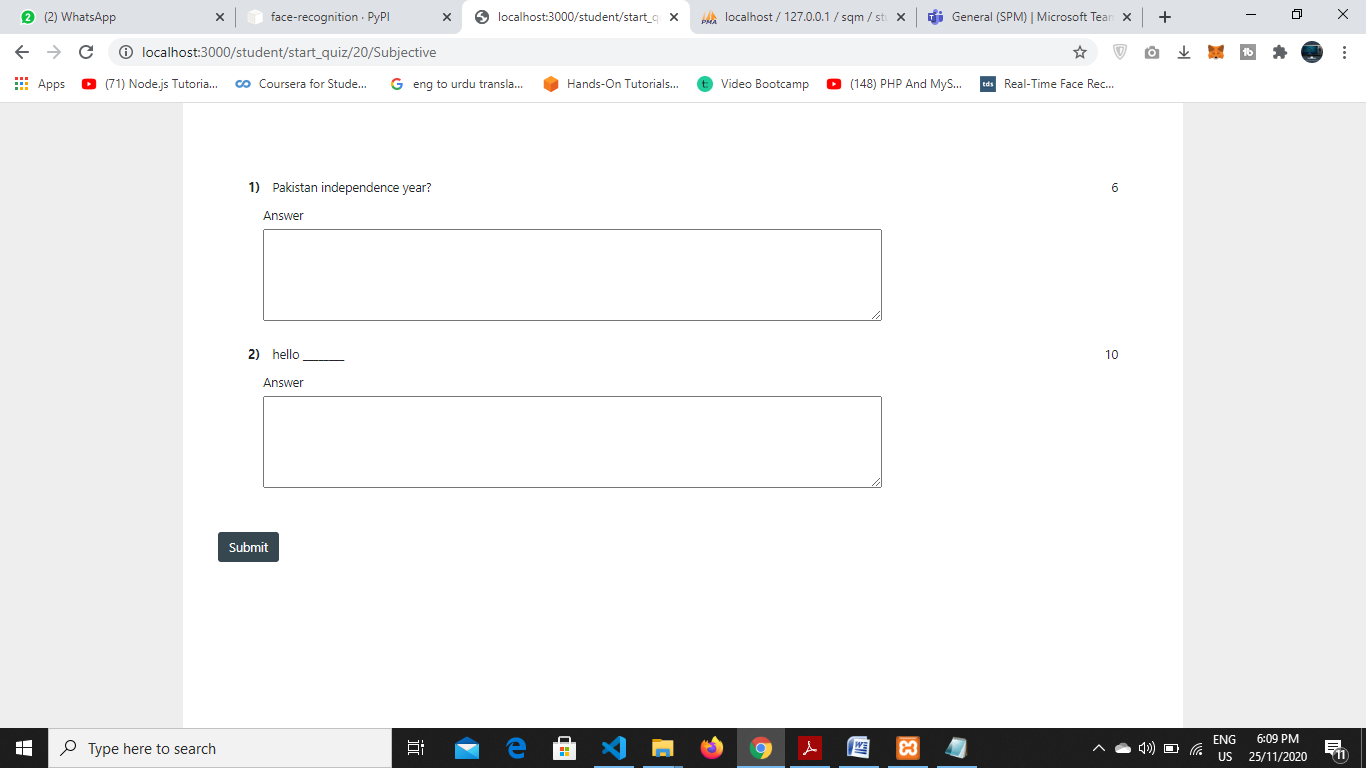


Figure 5-22 SQM Attempt Subjective Quiz

### **5.4.23 Attempt MCQs Quiz**

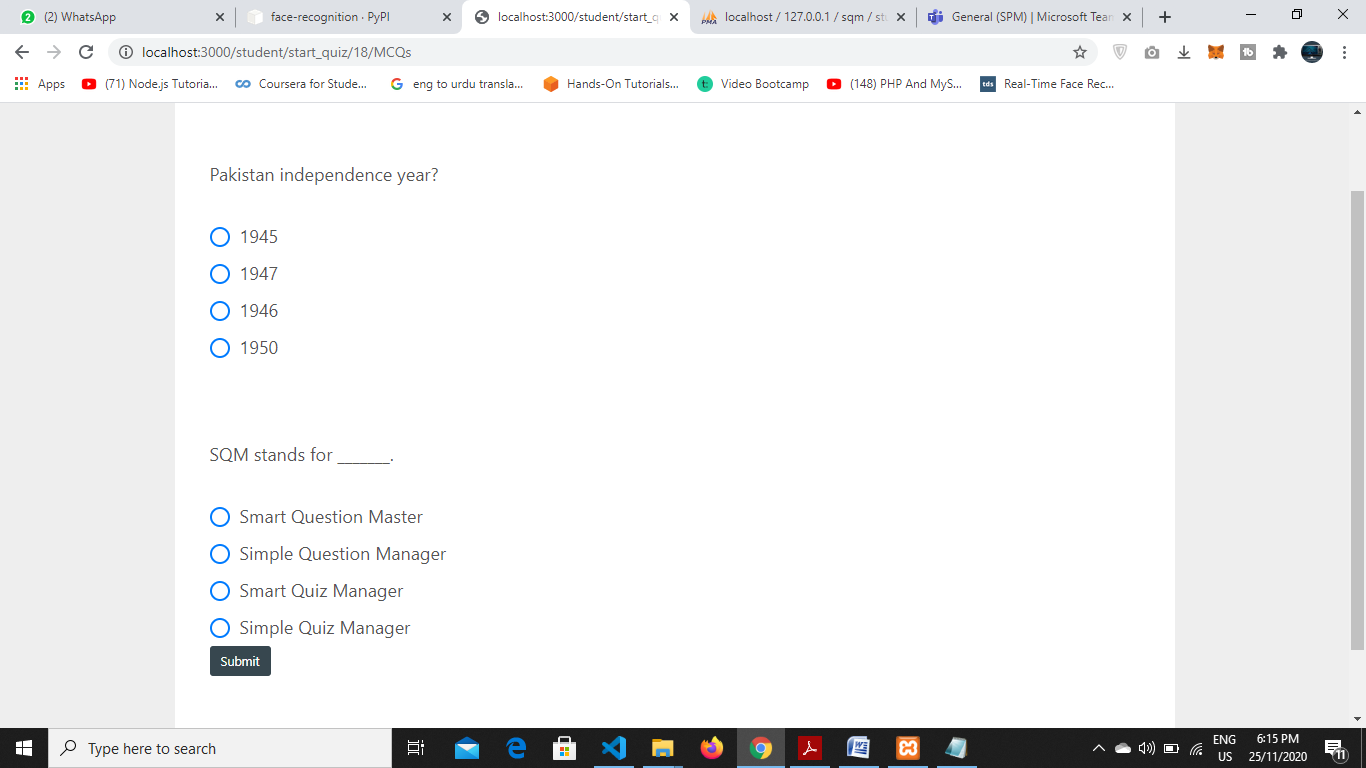


Figure 5-23 SQM Attempt MCQs Quiz

### **5.4.24 Student Profile**

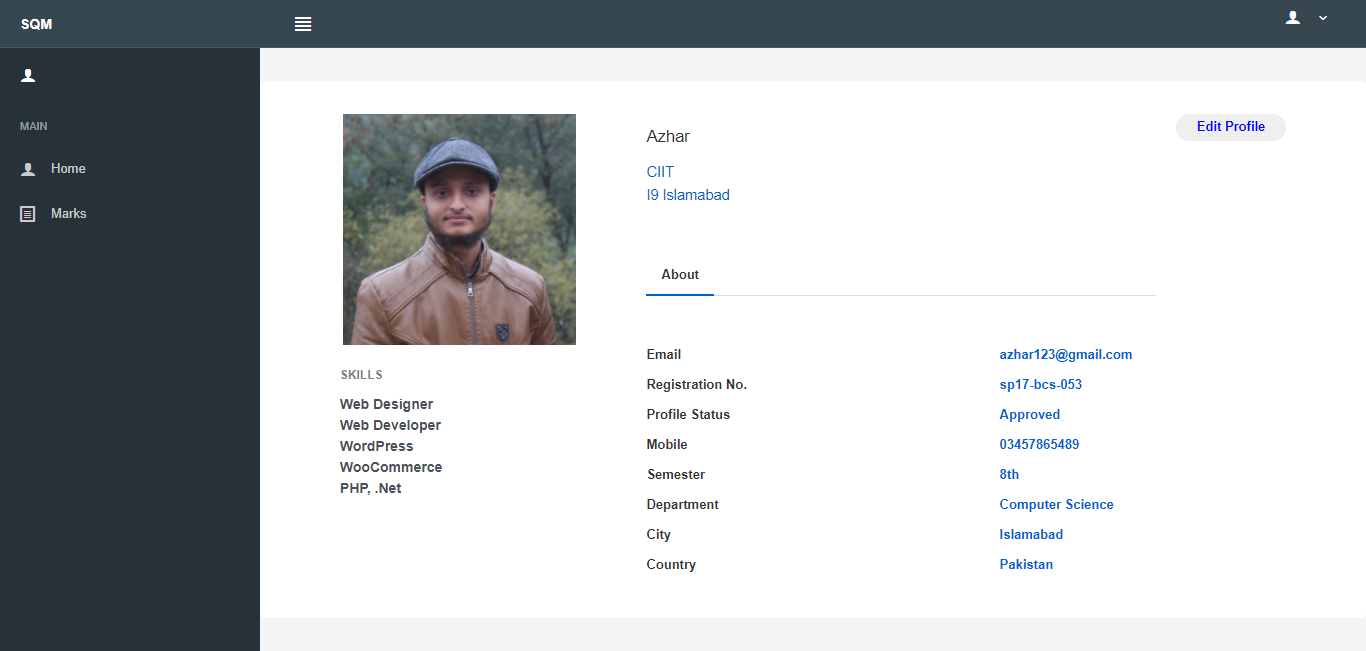


Figure 5-24 SQM Student Profile

### **5.4.25 Edit Profile**

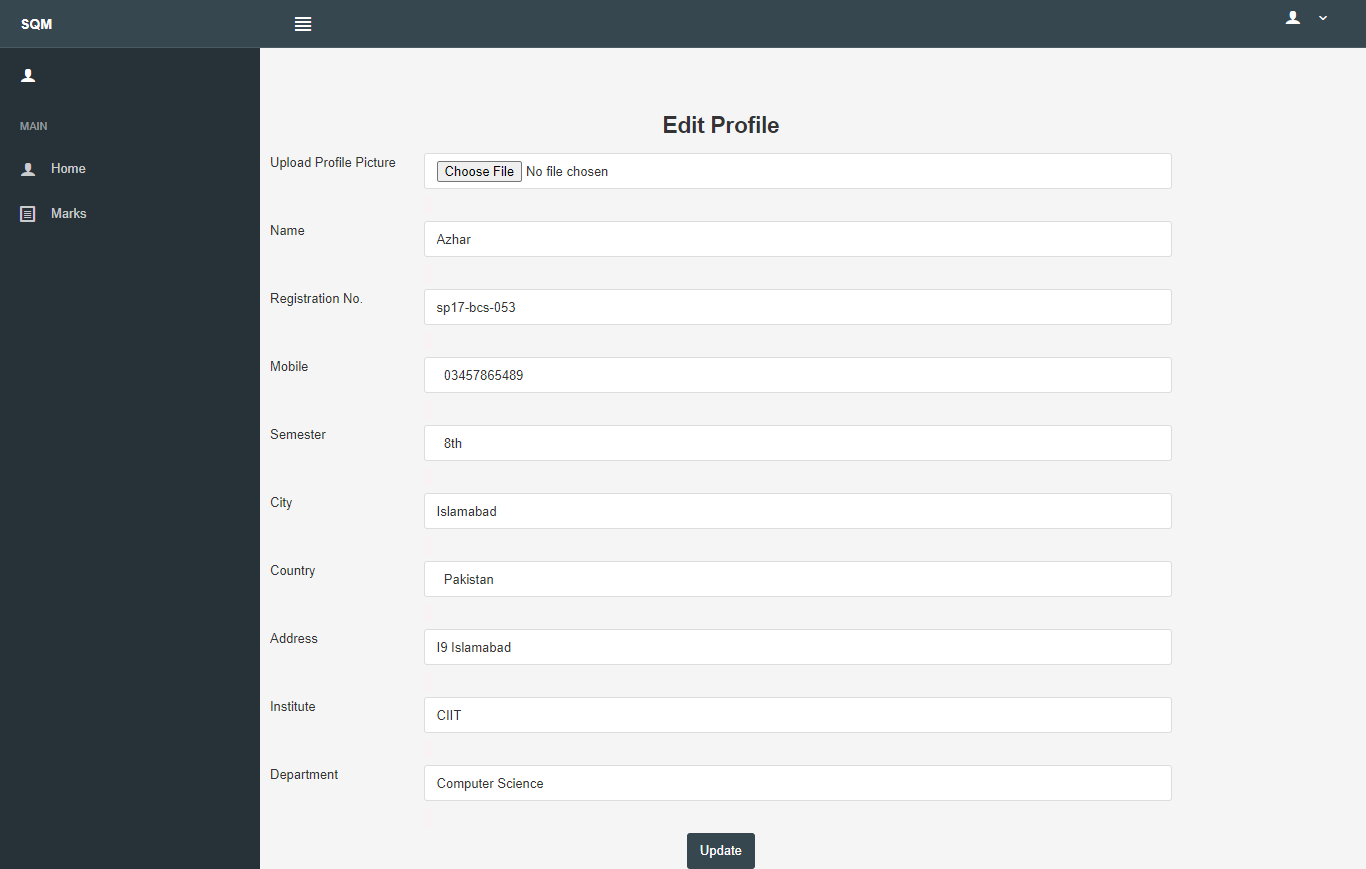


Figure 5-25 SQM Edit Profile

**Chapter # 6**

**Evaluation**

In this chapter, we evaluate our web application by applying different test cases to check the working of application.

## 6.1 **Testing**

Important phase of the development of software system is testing. Testing is process of analyzing a software system to check out that developed system is meeting its functional and non-functional requirements or not [6] and also to detect bugs in software (The difference between expected and actual condition of system). In testing phase we evaluate the features of our system. It can also important for identify or suggest optimizations in software system. Different types of testing are.

### **6.1.1 Black Box Testing**

Black box testing technique, in science black box is refer to a device which can be viewed only in terms of its input, output without knowledge of any internal working. In this type of testing tester test the behavior of system by giving input and looks output on the screen. Tester did not know the internal code structure, logic and implementations. By using this testing type we can only test constraints, validations and interfaces design and responsiveness. It is also known as functional testing.

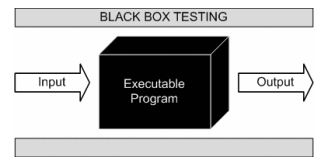


Figure 6-1 Black Box Testing

### **6.1.2 White Box Testing**

This type of testing is use to test the internal structure or working of application. In this type of testing, tester has visibility of internal structure and implementation. For operating white box testing, internal perspective and good programming skills are required. We use this testing type to find out logical mistakes and to optimize the code as much as possible. In this type tester give input to system and track the path from where input is converted to output.

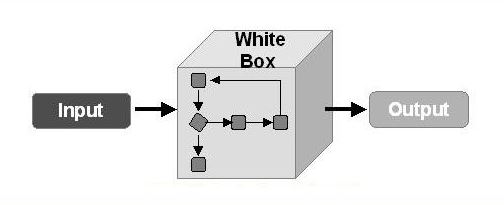


Figure 6-2 White Box Testing

### **6.1.3 Unit Testing**

Type of testing in which units or parts of system tested separately. As we mention in above chapters that we are using iterative software development approach, so our system is develop in iterations or modules. When a increment is completed we test it completely because output some increments is operate as input for next increment. So it is very important to test increment otherwise integration is very difficult. Developer itself works as tester in this type and checks each unit by executing the code that either it is giving the expected result or not.

### **6.1.4 Functional Testing**

Function testing is applied or used when software system is completely developed. In this phase we can check all functionalities of application that they provide required output or not. We can perform all functionalities of application. For example in our application we can authenticate mobile number, check login signup constraints and all the features of application that they are working correctly or not.

## Objective

Primary objective is to ensure that our application is working correctly without any bugs, errors or exceptions and test our application that it can meet its requirements or not [5]. We can perform testing process to find logical mistakes by applying several test cases. Testing can discovered optimizations in code logic and structure.

## Test Cases

Test cases are created to test the functional and non-functional of system. Test cases for our project are as follows.

### **6.1.4.1 Test Case (Teacher and Student Login)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected Result** | **Actual Result** | **P/F** |
| **1** | Click on login button with empty fields | Please Fill Out This Field(Required) | Please Fill Out This Field(Required) | **P** |
| **2** | Entered incorrect Email and password | Incorrect Email or Password | Incorrect Email or Password | **P** |
| **3** | Enter correct phone number and password | Sign In successful | Redirected To Dashboard | **p** |

Table 6-1 Test Case (Teacher and Student Login)

### **6.1.4.2 Test Case (Users Authentication and Registration)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected Result** | **Actual Result** | **P/F** |
| **1** | Click Submit button with empty email | Please Fill Out This Field | Please Fill Out This Field | **P** |
| **3** | Click Submit button with all empty fields | Please enter all fields | Please enter all fields | **P** |
| **4** | Click Submit button with different passwords | Password not Matched  Top of Form | Password not Matched | **P** |
| **5** | Enter all fields and click register | Account created successfully | Render to Dashboard | **P** |

Table 6-2 Test Case Registration

### **6.1.4.3 Test Case (Add Quiz)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected Result** | **Actual Result** | **P/F** |
| **1** | Click Create button without entering Quiz no, Type, Date and Time | Please enter all fields | Please enter all fields | **P** |
| **2** | Click Create button without selecting Quiz Type | Please Fill Out This Field | Please Fill Out This Field | **P** |
| **3** | Click Create button with entering all correct details | Quiz Created successfully | Render to saved quiz page to add questions | **P** |

Table 6-3 Test Case (Add Quiz)

### **6.1.4.4 Test Case (Add Questions)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected Result** | **Actual Result** | **P/F** |
| **1** | Click Add button without entering question, answer and marks | Please enter all fields | Please enter all fields | **P** |
| **2** | Click Add button without entering answer | Please Fill Out This Field | Please Fill Out This Field | **P** |
| **3** | Click Add button with entering all correct details | Question added to the quiz successfully | Question add(Render again this page) | **P** |

Table 6‑4 Test Case (Add Questions)

### **6.1.4.5 Test Case (Face Recognition)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected Result** | **Actual Result** | **P/F** |
| **1** | Click Space button with incorrect person | Face Not Matched | Face not matched | **P** |
| **2** | Click Space button when no profile pic is set | Please set your profile picture first | Please set your profile picture first | **P** |
| **3** | Click Space button with correct person | Face Matched | Face Matched(render to attempt quiz) | **P** |

Table 6‑5 Test Case (Face Recognition)

### **6.1.4.6 Test Case (Static Screen)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected Result** | **Actual Result** | **P/F** |
| **1** | Right Click outside the tab | Quiz Terminated | Quiz Terminated | **P** |
| **2** | Change tab using Alt+Tab | Quiz Terminated | Quiz Terminated | **P** |

Table 6‑6 Test Case (Place Order)

**Chapter # 7**

**Conclusion and Future Work**

**Future Work:**

In future this project can be further extended in this we also add the functionality of auto quiz generation. It will generate questions dynamically by the selected topic of the teacher. This project can be further extended in this we also add the functionality of speaking and listening due to which student will also give answer of the voice question and also make its live recording for the teacher when students will open any other functionality for cheating. With the passage of time and advancement of technology many more functionalities can be added to the smart quiz manager.

# **Conclusion**

The basic idea of this project was to control the cheating attempts during an online exam, save time, money and make easy examination system that cannot irritate teacher as well as student. We did this it in our project we make an easy approach to take quizzes and also work as a spy to control over the malpractices.

**References**

1. ecomputernotes.com
2. en.wikipedia.org
3. dspace.daffodilvarsity.edu.bd
4. [www.freedomfurniture.co.nz](http://www.freedomfurniture.co.nz)
5. Reqtest.com
6. Utpedia.utp.edu.my