## VISVESVARAYA TECHNOLOGICAL UNIVERSITY



**“JNANA SANGAMA” ,BELAGAVI-590018**

**A Mini Project Report On**

S ***“ONLINE EXAMINATION SYSTEM”***

**Submitted in the partial fulfillment of the requirement for the award of degree of**

### BACHELOR OF ENGINEERING IN

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

|  |  |  |
| --- | --- | --- |
| **RAKESH M R** | **B G VINAYAK** | **HARSHITA T** |
| **1VA17CS040** | **1VA17CS010** | **1VA17CS015** |

**Under the Guidance of**

|  |  |  |
| --- | --- | --- |
| **Prof. Sunil G L** | **Prof. Deeksha Hegde B** | **Prof. Nagamahesh B S** |
| **Assistant Professor,** | **Assistant Professor,** | **Assistant Professor,** |
| **Dept., of IS&E** | **Dept., of CS&E** | **Dept., of CS&E** |



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## SAI VIDYA INSTITUTE OF TECHNOLOGY

**(Affiliated to Visvesvaraya Technological University, Belagavi | Recognized by Govt. of Karnataka |Approved**

**by AICTE, New Delhi)**

**RAJANUKUNTE, BENGALURU – 560 064**

**2018-19**

S

SAI VIDYA INSTITUTE OF TECHNOLOGY

**(Affiliated to Visvesvaraya Technological University, Belagavi | Recognized by Govt. of Karnataka | Approved**

**by AICTE, New Delhi)**

**Rajanukunte, Bengaluru- 560 064**

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

Certified

hat

the

Mini

Project

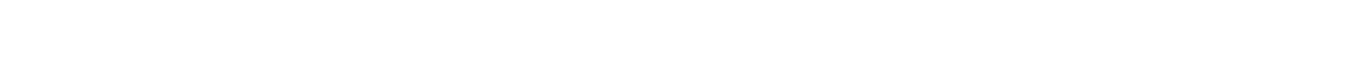
work entitled *“* ***ONLINE EEXAMINATION***

***SYSTEM****”* carried out by **Mr. RAKESH M R (1VA17CS040) Mr. B VINAYAK (1VA17CS040) HARSHITHA T (1VA17CS016) ,** bonafide students of **SAI VIDYA INSTITUTE OF TECHNOLOGY**, Bengaluru, in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering of **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**,

t

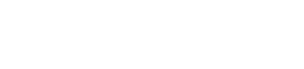
**G**

**n**



Belagavi during the year **2019-20.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said Degree.





**Prof. Sunil G L** Assistant Professor, Dept., of IS&E



**Prof*.* Deeksha Hegde B** Assistant Professor, Dept., of CS&E



**Prof. K Ananthapadma abha**

Professor & Head, Dept., of CS&E



**Dr. H S Ramesh Babu**

Principal





**Name of External Examiners**



**Signature with Date**

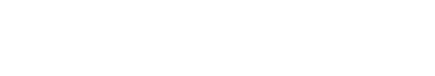




1.



2.



# ACKNOWLEDGEMENT

The completion of Project brings with and sense of satisfaction, but it is never completed without thanking the persons who are all responsible for its successful completion. First and foremost, we wish to express our deep sincere feelings of gratitude to our Institution, **Sai Vidya Institute of Technology**, for providing us an opportunity to do our education.

We would like to thank the **Management** and **Prof. M R Holla,** Director, Sai Vidya Institute of Technology for providing the facilities.

We extend our deepest sense of sincere gratitude to **Dr. H S Ramesh Babu**, Principal, Sai Vidya Institute of Technology, Bengaluru, for having permitted us to carry out the project work on ***ONLINE EXAMINATION SYSTEM*.**

We are thankful to **Prof. A M Padmareddy,** Professor and Dean (Student affairs), Department of Computer Science and Engineering, Sai Vidya Institute of Technology, for his constant support and motivation.

We express our heartfelt sincere gratitude to **Prof. K Ananthapadmanabha** Professor and Head, Department of Computer Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for his valuable suggestions and support.

We express my special in-depth, heartfelt, sincere gratitude to **Prof. Sunil G L,** Asst. Prof, Dept., of CS&E, **Prof. Deeksha Hegde B,** Asst. Prof, Dept., of CS&E, **Prof. Nagamahesh B S,** Asst. Prof, Dept., of CS&E, Sai Vidya Institute of Technology, Bengaluru for their constant support in completing the project.

Finally, we would like to thank all the Teaching, Technical faculty and supporting staff members of Department of Computer Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for their support.

**RAKESH M R 1VA17CS040 B G VINAYAK 1VA17CS010 HARSHITHA T 1VA17CS015**



# ABSTRACT

ONLINE EXAMINATION SYSTEM is a web-based examination system where examinations are given online. either through the internet or intranet using computer system. The main goal of this online examination system is to effectively evaluate the student thoroughly through a totally automated system that not only reduce the required time but also obtain fast and accurate results.

ONLINE EXAMINATION SYSTEM is an online test simulator is to take online examination, test in an efficient manner and no time wasting for manually checking of the test paper. The main objective of this web based online examination system is to efficiently evaluate the student thoroughly through a fully automated system that not only saves lot of time but also gives fast and accurate results. For students they give papers according to their convenience from any location by using internet and time and there is no need of using extra thing like paper, pen etc.

Functional Specification:

1. Registering new Student
2. Getting the student and staff info and storing it to databases
3. Getting the type and number of question paper 4. Generating result after the exam.



# CONTENTS

### Chapters Page No

**ACKNOWLEDGEMENT i**

**ABSTRACT ii**

1. **INTRODUCTION TO DBMS 1**
2. **SYSTEM SPECIFICATIONS 4**
3. **SYSTEM DESIGN 6**
4. **FUNCTIONS 9**
5. **MODULES OF ONLINE 11**

**EXAMINATION SYSTEM**

1. **CONCLUSION & FUTURE ENHANCEMENT 13**

**BIBILOGRAPHY 14**

**APPENDIX-A**

**SNAPSHOTS**

**CHAPTER 1**

**INTRODUCTION**

#### Overview of Database Management System

Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this one can define DBMS as a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data.

**Storage:** According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage. Let’s take a layman example to understand this. In a banking system, suppose a customer is having two accounts, one is saving account and another is salary account. Let’s say bank stores saving account data at one place (these places are called tables we will learn them later) and salary account data at another place, in that case if the customer information such as customer name, address etc. are stored at both places then this is just a wastage of storage (redundancy/ duplication of data), to organize the data in a better way the information should be stored at one place and both the accounts should be linked to that information somehow. The same thing we achieve in DBMS.

**Fast Retrieval of data**: Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

#### Problem Statement

Redefining conventional examination methodologies, this project endeavors to construct a dynamic online examination system empowered by a sophisticated DBMS infrastructure. The central objective is to elevate user engagement, streamline administrative tasks, and fortify data security within educational domains. The system's architecture embraces a seamless test management approach, orchestrating scheduling, test-taking, and result generation with precision. By harnessing the potential of the DBMS, the system ensures efficient resource allocation, scalability, and optimal data storage. Rigorous security measures encompass encryption, access controls, and data anonymization to thwart unauthorized access and safeguard sensitive information. This endeavor holds the promise to revolutionize the educational assessment landscape, offering institutions an avant-garde solution that not only expedites examination processes but also fosters a more secure and technologically adept learning environment. The project anticipates serving as a cornerstone for future enhancements and innovations in the realm of online examination systems.

#### Objective

The project envisions an innovative Online Examination System, integrating Database Management Systems (DBMS) principles to modernize traditional exam processes. It aims to:

* Optimize resource allocation, data integrity, and user experience.
* Automate administrative tasks, streamlining test creation, scheduling, and result analysis.
* Ensure robust security measures, including encryption and access controls.
* Promote eco-friendly practices through reduced paper usage.
* Facilitate scalability and modular design for future enhancements.

By contributing to efficient, secure, and sustainable online assessments, this project stands at the forefront of educational technology, enriching learning experiences and administration alike.

* 1. **Dataset Description**

The dataset for this project encompasses a comprehensive collection of examination-related data, meticulously structured to support the functionality and objectives of the Online Examination System. It includes:

* **Student Information**: Details of registered students, including names, unique identifiers, contact information, and authentication credentials.
* **Examination Schedule**: Information regarding test dates, times, duration, and associated courses or subjects.
* **Test Responses**: Individual student responses to specific questions during an examination, stored anonymously to ensure data privacy.
* **Results and Scores**: Compilation of student scores, providing a comprehensive overview of performance, including total scores, section-wise scores, and correct/incorrect responses.
* **Administrative Logs**: Detailed records of system activities, user logins, test creations, and modifications to ensure accountability and traceability.
* **Security Data**: Encryption keys, access control lists, and other security-related parameters to uphold the confidentiality and integrity of the system.

The dataset ensures compatibility with the DBMS architecture, facilitating efficient data retrieval, updates, and queries. Its comprehensive nature empowers the Online Examination System to seamlessly execute test management, user authentication, result analysis, and system maintenance, contributing to an enhanced and secure examination experience.

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

* 1. **Software and Hardware**

The Online Examination System demands a robust technical infrastructure to ensure its smooth operation, security, and scalability. The following system requirements are crucial for its successful implementation:

**SOFTWARE REQUIREMENTS**

Front end : PHP, HTML, CSS

Operating system : Windows 11, 64 Bit

Server side language : PHP

Back End : MySQL web sever : Apache

Application Software\ : XAMP

**HARDWARE REQUIREMENTS**

Processor: Intel Core i5

RAM: 8 GB

Hard disk: 1TB

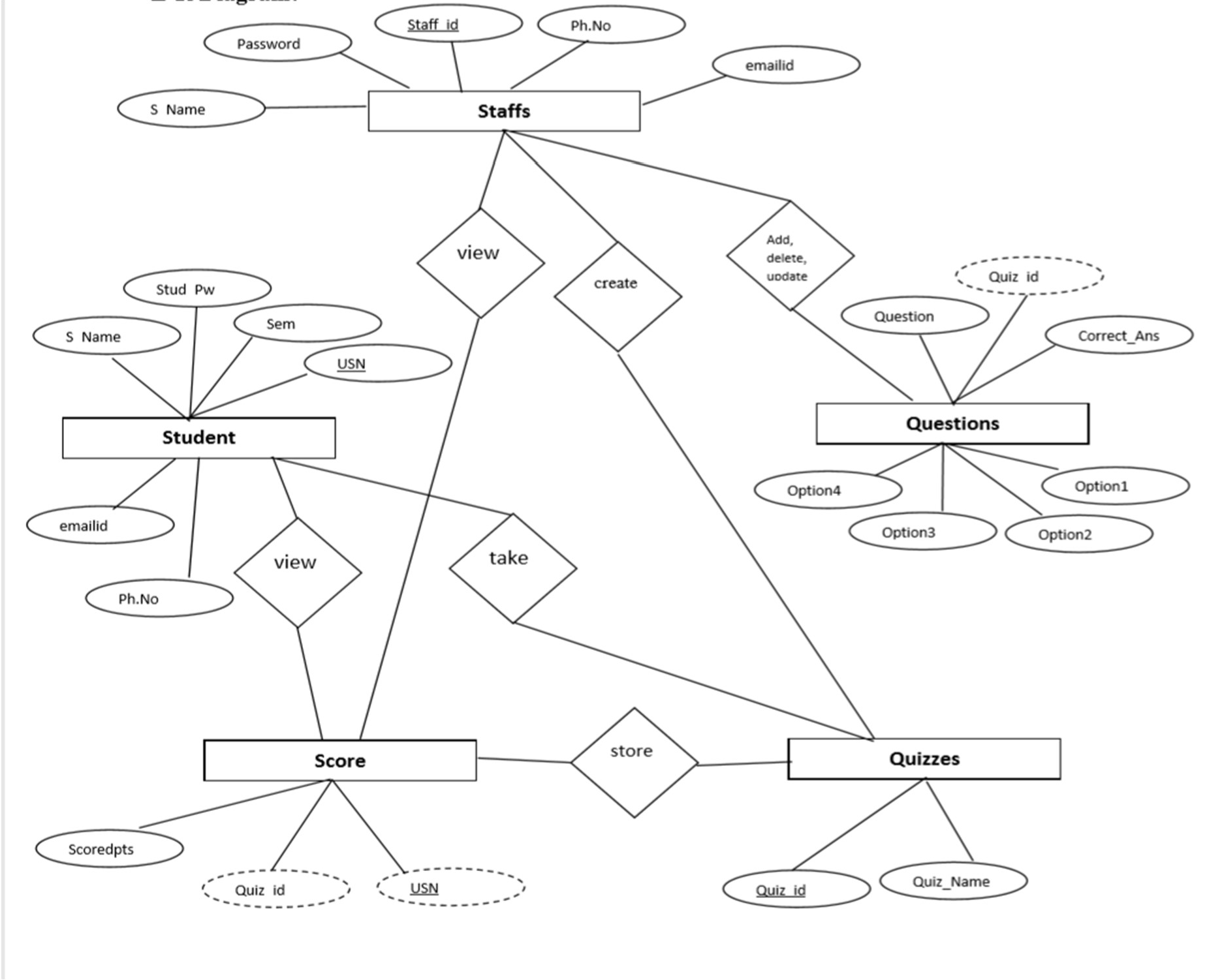
## CHAPTER: 3

**SYSTEM DESIGN**

This chapter of the report describes the structure of the project, followed by Entity Relationship Diagram, Schema Diagram and the table structures.

**3.1 ER Diagram with relationships and cardinality ratio**

An entity relationship model*,* also called an entity-relationship (ER) diagram*,* is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.The cardinalit*y* or fundamental principle of one data aspect with respect to another is a critical feature. The relationship of one to the other must be precise and exact between each other in order to explain how each aspect links together. In simple words Cardinality is a way to define the relationship between two entities.



1

N

M

N

N

N

N

M

M

M

Dept\_Name

Dept\_id

**Dept**

**Fig 3.1: ER Diagram of Online Examination System**

**3.2 ER to Relation Mapping**

The Entity-Relationship (ER) diagram serves as a graphical representation of the relationships among various entities in our project. To translate this visual representation into a functional database design, we perform ER to Relation Mapping. This process involves creating structured tables that represent entities, their attributes, and the relationships between them.

**Mapping the regular entities**

The entities do not include the foreign and relational attributes. The relations that are created from the mapping of entity types are sometimes called entity relations because each tuple represents an entity instance.

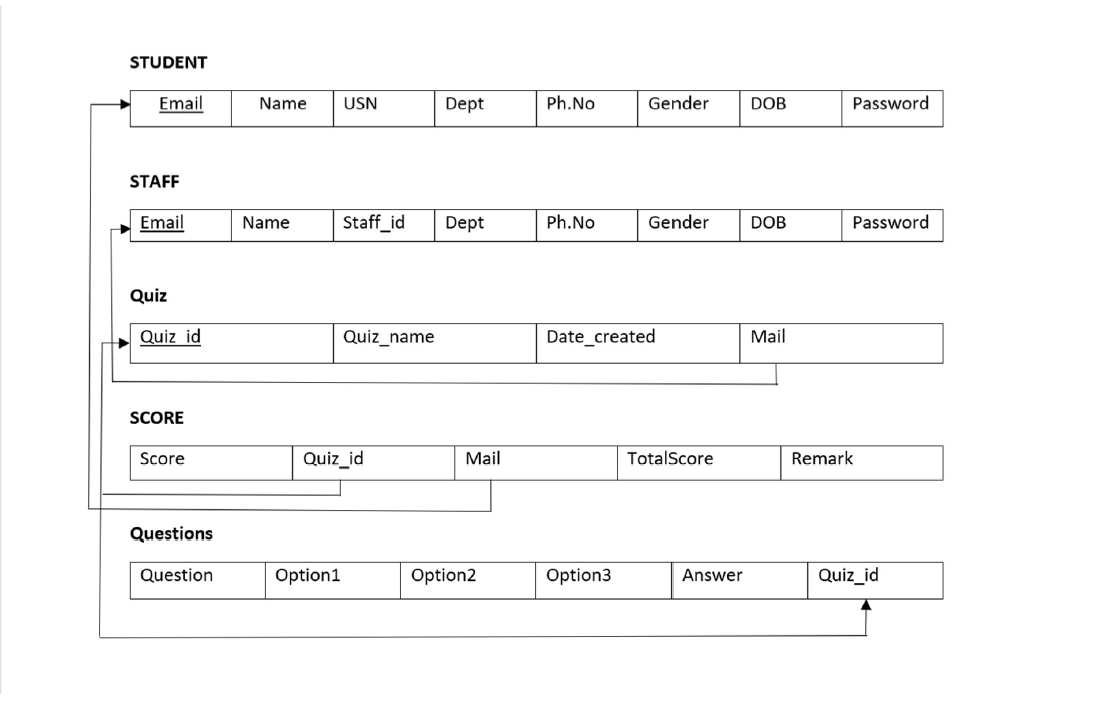




Fig.3.2.1 All the entities and attributes

**Mapping of weak entity types**

We do not have weak entities in our ER Diagram. A entity is called "weak" when its existence depends on the existence of another (i.e., strong) entity.

**Mapping of 1:1 cardinality relationships**

In a 1:1 relationship, one record in one table is associated with exactly one record in another table, and vice versa. In your project, you have a 1: n relationship between the student table and the department table. Each student can have one corresponding department. Same goes with the staff table.

**Mapping of 1:N cardinality relationships**

A 1:N relationship signifies that one record in a table is related to multiple records in another table. In your project, you have several 1:N relationships:

Each student can take multiple quiz but each order is associated with only one student. This represents a 1:N relationship .

**Mapping of M:N cardinality relationships**

There are many M:N cardinality relationships we can analyze the existing tables to identify potential M:n relationships based on the foreign key relationships between the tables.

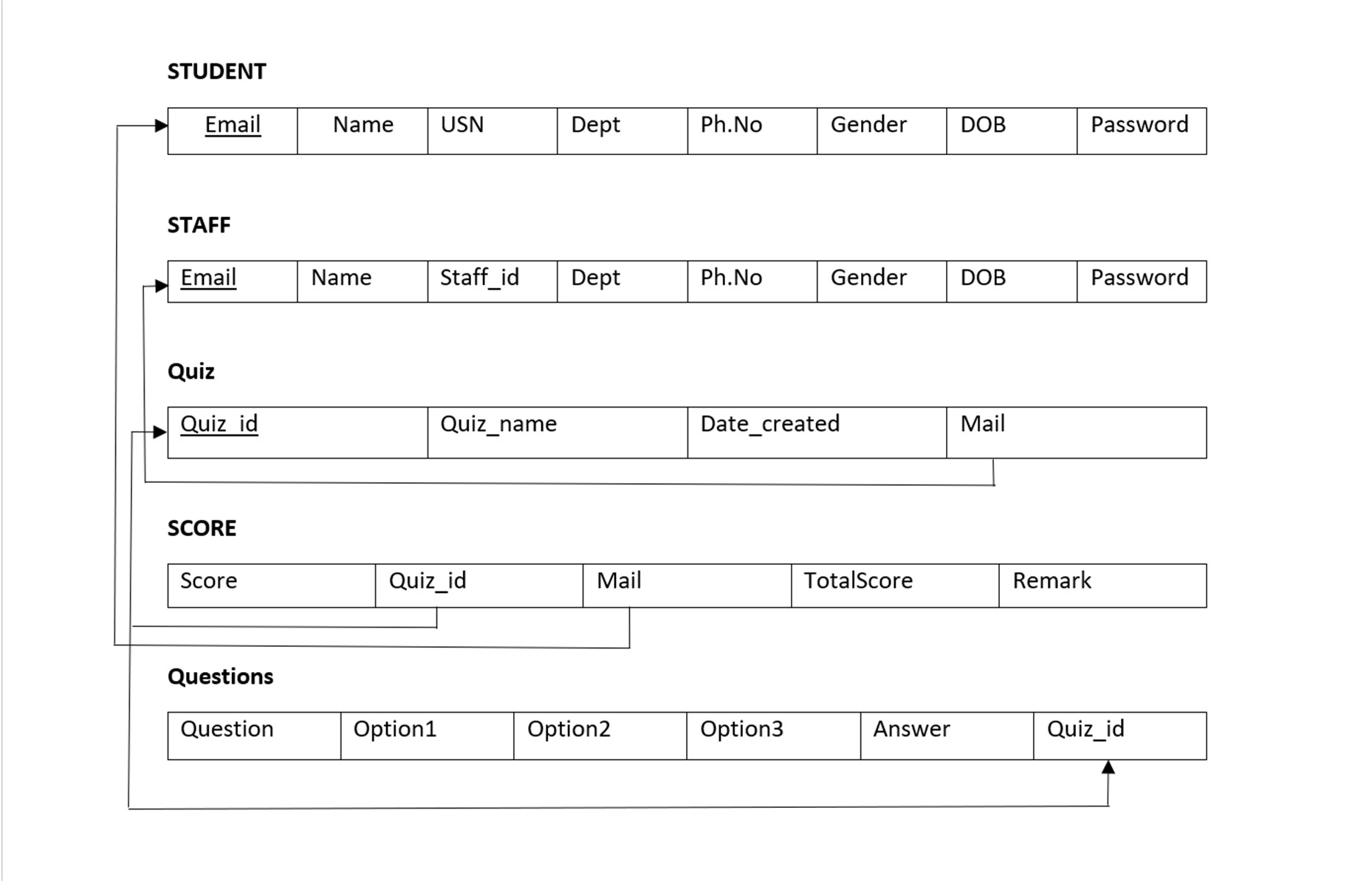
* Table questions and quiz:
  + The questions table has a foreign key quizid that references the quiz table's primary key quizid.
  + This indicates a one-to-many relationship between quiz (one) and questions (many), where each quiz can have multiple questions.
* Table score and student:
  + The score table has a foreign key mail that references the student table's primary key mail.
  + This indicates a one-to-many relationship between student (one) and score (many), where each student can have multiple scores (for different quizzes).
* Table score and quiz:
* The score table has a foreign key quizid that references the quiz table's primary key quizid.
* This indicates a one-to-many relationship between quiz (one) and score (many), where each quiz can have multiple scores (from different students).
* Table student and quiz:
* The quiz table has a foreign key mail that references the staff table's primary key mail.
* This indicates a one-to-many relationship between staff (one) and quiz (many), where each staff member can create multiple quizzes.

**Foreign Keys And Relationships**

* Quiz.mail is a foreign key that references the mail column in the staff table. This means that each quiz must be created by a staff member. The mail column in the quiz table is the primary key of the staff table, so this foreign key constraint ensures that each quiz is associated with a unique staff member.
* Score.quizid is a foreign key that references the quiz\_id column in the quiz table. This means that each score must be for a particular quiz. The quizid column in the score table is the primary key of the quiz` table, so this foreign key constraint ensures that each score is associated with a unique quiz.
* Score.mail is a foreign key that references the mail column in the student table. This means that each score must be taken by a particular student. The mail column in the score table is the primary key of the student` table, so this foreign key constraint ensures that each score is associated with a unique student.

**3.3 Schema Diagram**

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and the relations among them are associated. It formulates all the constraints that are to be applied on data. A database schema defines its entities and relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It shows the various relations, references between entities. The arrows point towards the primary key and the other side will be the foreign keys in the relations respectively. The underlined attributes are the primary keys of the relations respectively. They are used to uniquely identify the rows of a table. Thus, a row that needs to be uniquely identified, the key constraint is set as the Primary key to that field.

Fig. 3.3.1 

**Fig 3.3: Schema Diagram**

**3.4 Over view of GUI**

A Graphical User Interface (GUI) is a software interface that leverages the visual capabilities of a computer to simplify program usage. An effective GUI eliminates the need for users to learn complex command languages, enhancing user experience. Conversely, some users find command-driven interfaces more productive, especially if they are already familiar with the command language.

In the realm of human-computer interaction, designing a GUI's visual layout and behavior is pivotal in software application programming. The goal is to enhance the logical design of an underlying program, a facet of design referred to as usability.

• HTML (Hypertext Markup Language) is the markup language used to create web pages and applications. Paired with JavaScript, it forms a trio of essential web-enabling technologies. Web browsers render multimedia web pages by transforming HTML documents received from servers or local storage. HTML originally signaled content appearance and semantically defined webpage structure.

• Cascading Style Sheets (CSS) is a language for crafting style sheets that dictate the presentation of a markup language document like HTML. Alongside HTML and JavaScript, CSS is a cornerstone of the World Wide Web. CSS segregates layout, color, and font choices from content. By defining relevant CSS in a separate .css file, multiple web pages can share formatting, enhancing content accessibility, flexibility, control over presentation, and reducing complexity.

In the context of web development with PHP, these principles extend to creating dynamic and interactive web applications. PHP facilitates server-side scripting, enabling seamless integration of data processing and user interaction within the GUI. The PHP, HTML, and CSS trio empowers developers to build feature-rich and user-friendly web applications.

**3.5 Normalization**

Normalization is a multi-step process that involves breaking down a relational database into smaller, well-organized tables to eliminate data redundancy and anomalies. In your project, this process ensures that data is stored efficiently, improving data integrity and overall system performance.

1. First Normal Form (1NF):

Ensure that each table has a primary key and that no column contains repeating groups.

All tables in the provided SQL script already have primary keys.

2. Second Normal Form (2NF):

Remove partial dependencies (attributes that depend on only part of the primary key) from the tables.

The tables seem to already adhere to 2NF, as there are no partial dependencies.

3. Third Normal Form (3NF):

Remove transitive dependencies (attributes that depend on non-key attributes) from the tables.

There might be a transitive dependency in the questions table. The op1, op2, op3, and answer columns depend on qs, which is not a candidate key. To resolve this, we can create a separate table for options and link them to the questions table.

After applying normalization, the tables might look like this:

* dept (Dept\_ID [PK], Dept\_Name)
* questions (Qs\_ID [PK], Qs, Quiz\_ID [FK], Answer)
* options (Option\_ID [PK], Option, Question\_ID [FK])
* quiz (Quiz\_ID [PK], Quiz\_Name, Date\_Created, Mail [FK])
* score (Slno [PK], Score, Quiz\_ID [FK], Mail [FK], Total\_Score, Remark)
* staff (Staff\_ID [PK], Name, Mail, Phno, Gender, DOB, PW, Dept [FK])
* student (USN [PK], Name, Mail, Phno, Gender, DOB, PW, Dept [FK])

**CHAPTER: 4**

**IMPLEMENTATIONS**

**4.1 Table Creation**

CREATE TABLE IF NOT EXISTS `dept` (

`dept\_id` int(11) NOT NULL,

`dept\_name` varchar(3) DEFAULT NULL,

PRIMARY KEY (`dept\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `questions` (

`qs` varchar(200) NOT NULL,

`op1` varchar(30) NOT NULL,

`op2` varchar(30) NOT NULL,

`op3` varchar(30) NOT NULL,

`answer` varchar(30) NOT NULL,

`quizid` int(11) NOT NULL,

UNIQUE KEY `qs` (`qs`),

KEY `quizid` (`quizid`),

KEY `quizid\_2` (`quizid`),

KEY `quizid\_3` (`quizid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `quiz` (

`quizid` int(11) NOT NULL AUTO\_INCREMENT,

`quizname` varchar(20) NOT NULL,

`date\_created` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`mail` varchar(30) DEFAULT NULL,

PRIMARY KEY (`quizid`),

KEY `mail` (`mail`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=7 ;

CREATE TABLE IF NOT EXISTS `score` (

`slno` int(11) NOT NULL AUTO\_INCREMENT,

`score` int(11) NOT NULL,

`quizid` int(11) NOT NULL,

`mail` varchar(30) DEFAULT NULL,

`totalscore` int(11) DEFAULT NULL,

`remark` varchar(20) DEFAULT NULL,

PRIMARY KEY (`slno`),

KEY `quizid` (`quizid`),

KEY `mail` (`mail`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=15 ;

CREATE TABLE IF NOT EXISTS `staff` (

`staffid` varchar(10) NOT NULL,

`name` varchar(20) NOT NULL,

`mail` varchar(30) NOT NULL,

`phno` varchar(10) NOT NULL,

`gender` varchar(1) NOT NULL,

`DOB` varchar(10) NOT NULL,

`pw` varchar(200) NOT NULL,

`dept` varchar(3) DEFAULT NULL,

PRIMARY KEY (`mail`),

UNIQUE KEY `mail` (`mail`,`phno`),

UNIQUE KEY `staffid` (`staffid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `student` (

`usn` varchar(10) NOT NULL,

`name` varchar(20) NOT NULL,

`mail` varchar(30) NOT NULL,

`phno` varchar(10) NOT NULL,

`gender` varchar(1) NOT NULL,

`DOB` varchar(10) NOT NULL,

`pw` varchar(200) NOT NULL,

`dept` varchar(3) DEFAULT NULL,

PRIMARY KEY (`mail`),

UNIQUE KEY `mail` (`mail`),

UNIQUE KEY `phno` (`phno`),

UNIQUE KEY `usn` (`usn`),

KEY `dept` (`dept`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Constraints for table `score`

--

ALTER TABLE `score`

ADD CONSTRAINT `score\_ibfk\_1` FOREIGN KEY (`quizid`) REFERENCES `quiz` (`quizid`) ON DELETE CASCADE,

ADD CONSTRAINT `score\_ibfk\_2` FOREIGN KEY (`mail`) REFERENCES `student` (`mail`) ON DELETE CASCADE ON UPDATE CASCADE;

**CHAPTER 5**

**MODULES IN ONLINE EXAMINATION SYSTEM**

Online Examination System is that which enables the Students & Teachers to register for the system. Students are allowed to take the online test and see their progress. Also, it enables the Teachers to add, delete, update the test questions and also to keep track of the student progress.

* **Student**
* **Staff**
* **Login**
* **Add/Remove quizzes and questions**
* **Score**

**Student:**

Student has to log in to the system and can then view all the quizzes. Student can see the quiz list and attend the quizzes. After attending the quizzes students will get instant result and the same will be stored in the database.

**Staff:**

Staff also has to log in to the system first, then they can add/remove quizzes. They

can see the progress and they can also update the existing quiz.

**Login:**

Login is must and should for both staff and student. So that all records will be safely saved to the database. If someone had forgot the password ,one can reset that password.

**Add/Remove quiz and question:**

Staff can add, remove the quiz. And also they can add extra questions to an existing

quiz.

**Score:**

Staff can see the scoreboard of the quiz which is added by him/her. And student

can see the score of the quiz which he/she is attended.

Dept., of CS&E,SVIT 2019-20 11

**CHAPTER 6**

**CONCLUSION**

The online examination system provides better functionality for an examination to be more efficient and reduce manual paperwork in order to automate all possible tasks. For implementing this system, PHP, HTML, CSS, JavaScript and MySql are used.

The system comprises of following features:

* Management of quiz.
* Automated grading.
* Adding/deleting quizzes and questions.

#### SCOPE OF ENHANCEMENT

There are also few features which can be integrated with the system to make it more flexible.

Below list shows the future points to be considered:

* Implementing the timer for the quiz.
* Sending mails on sign up and when student takes the quiz.
* Supporting all type of questions including MCQ’s.
* Including Programming Questiond where user can compile or interpret on site only.

Dept., of CS&E,SVIT 2018-19 12

#### BOOK REFERENCE

**BIBLIOGRAPHY**

1. Database System Models, Languages, Ramez Elmasri and Sham Kant B. Navathe, 7th

Edition, 2017 Pearson.

1. Fundamentals of Web Development, Randy Connolly and Ricardo Hoar ,First Impression, 2016 Pearson

#### WEBSITES

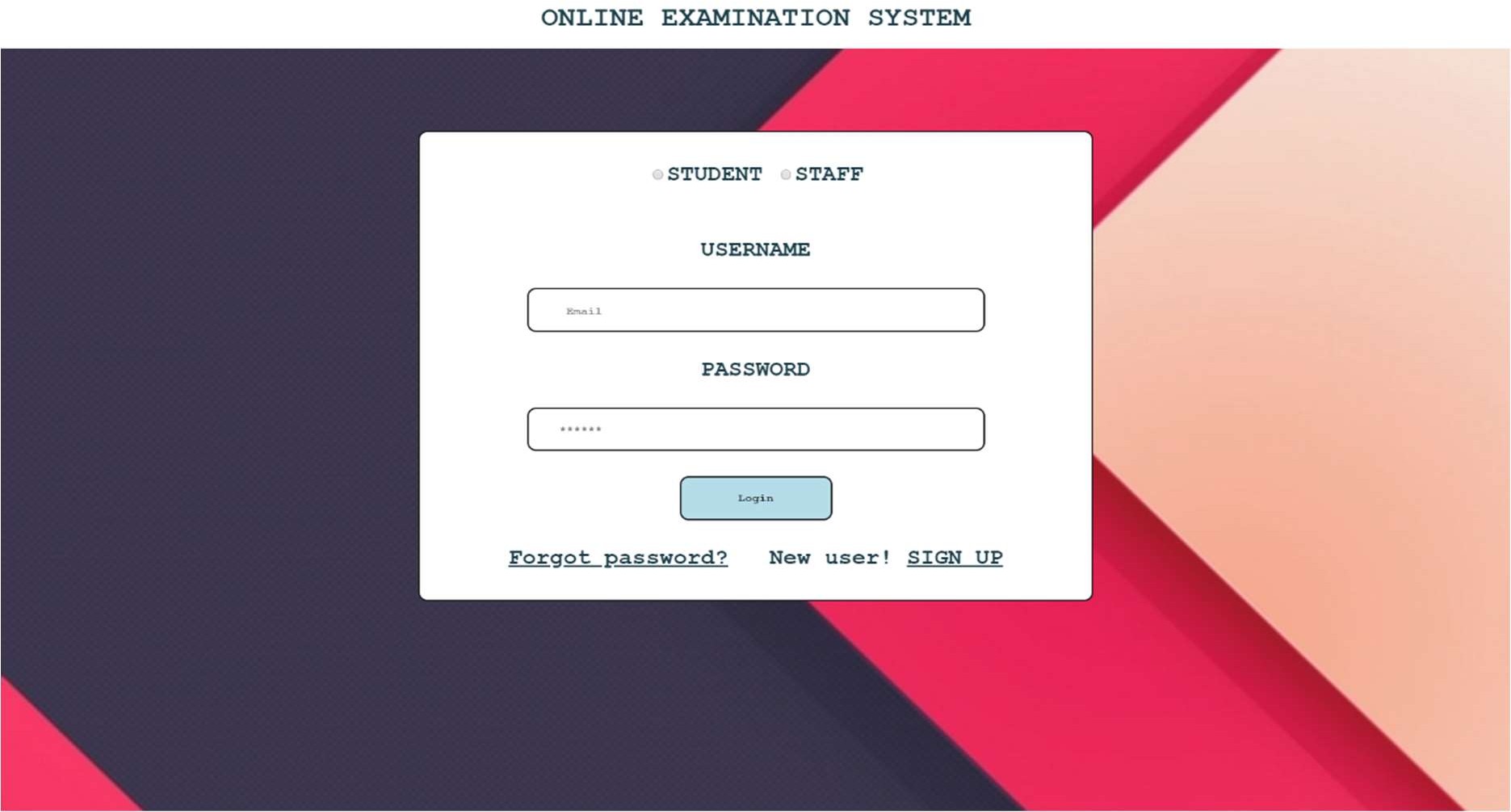
1. w
2. w

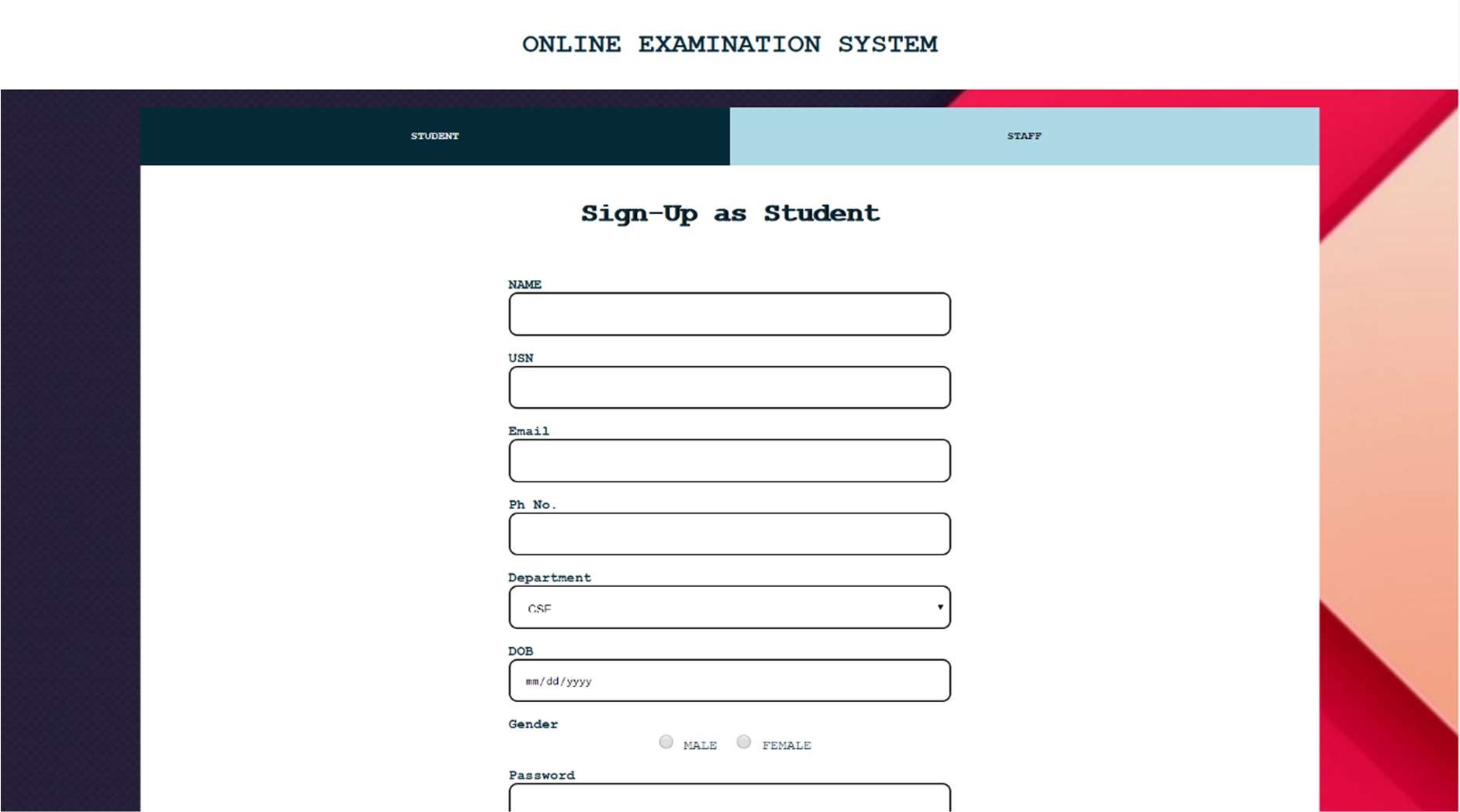
ww.stackoverflow.com ww.youtube.com

1. h ttps://[www.php.net](http://www.php.net/)
2. w ww.google.com
3. h ttps://[www.w3schools.com](http://www.w3schools.com/)

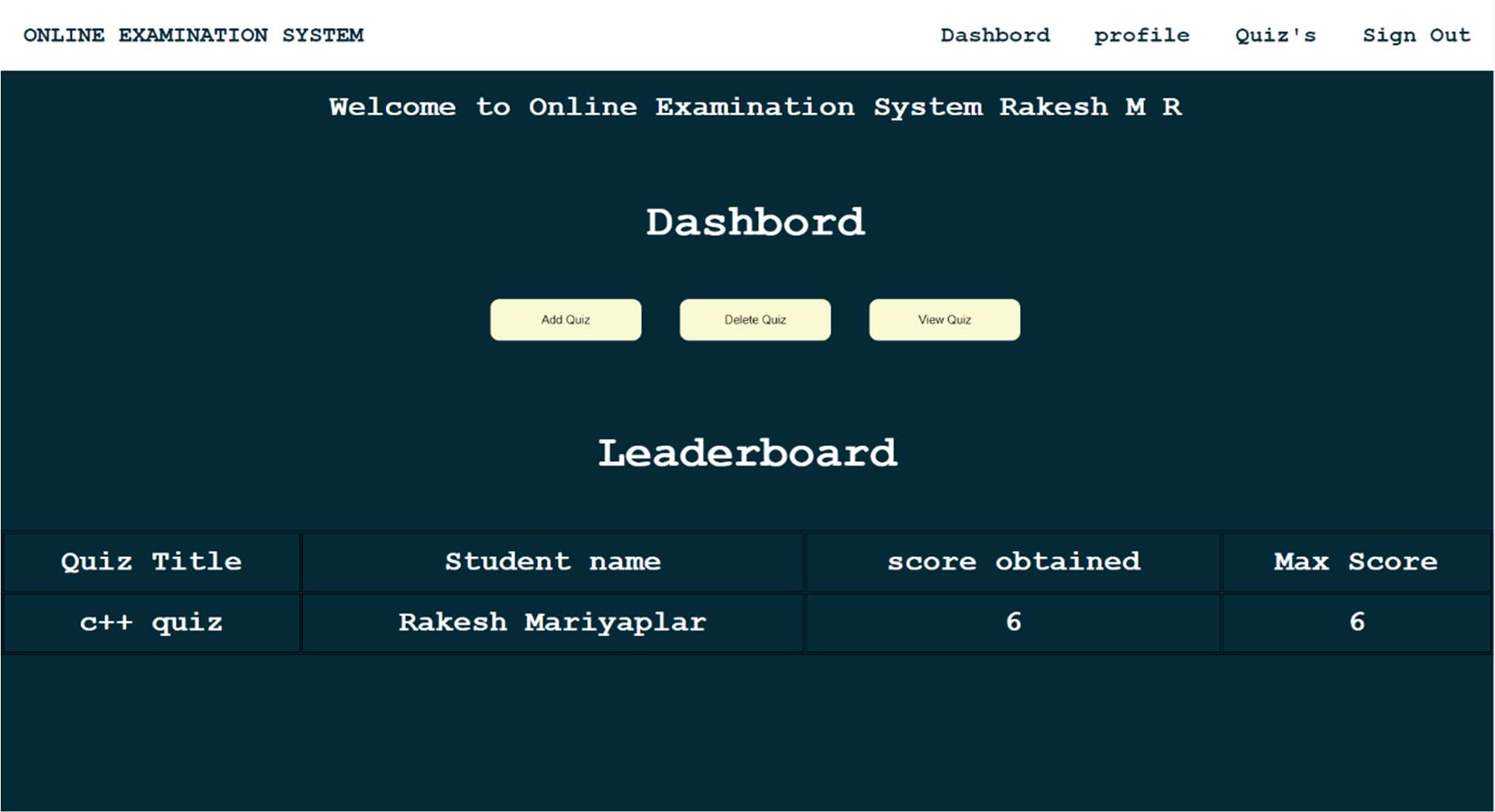
Dept. of CS&E, SVIT 2019-20 13

**APPENDIX A**

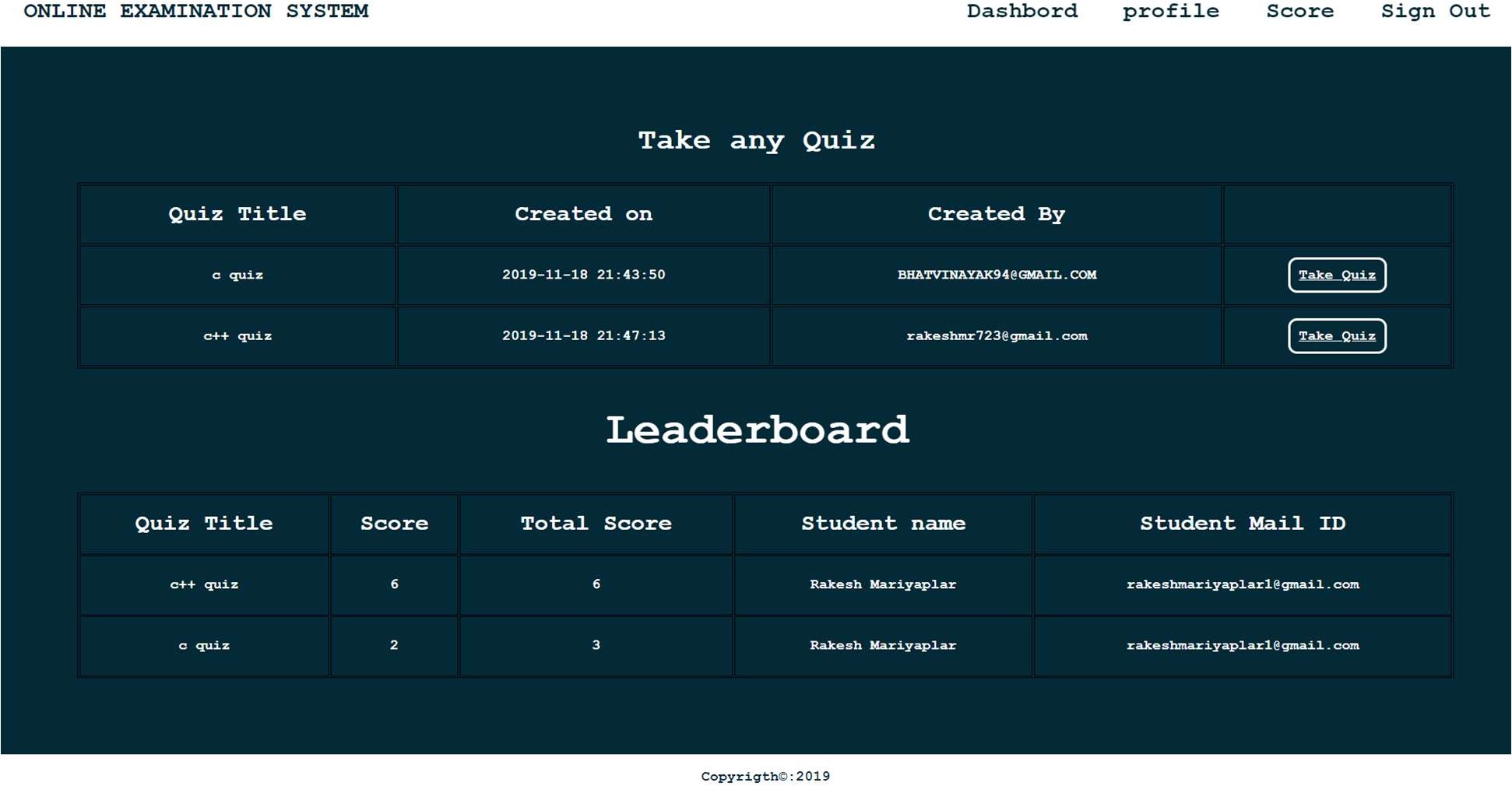
**SNAPSHOTS**

**Snapshot 1: Start page of the project**

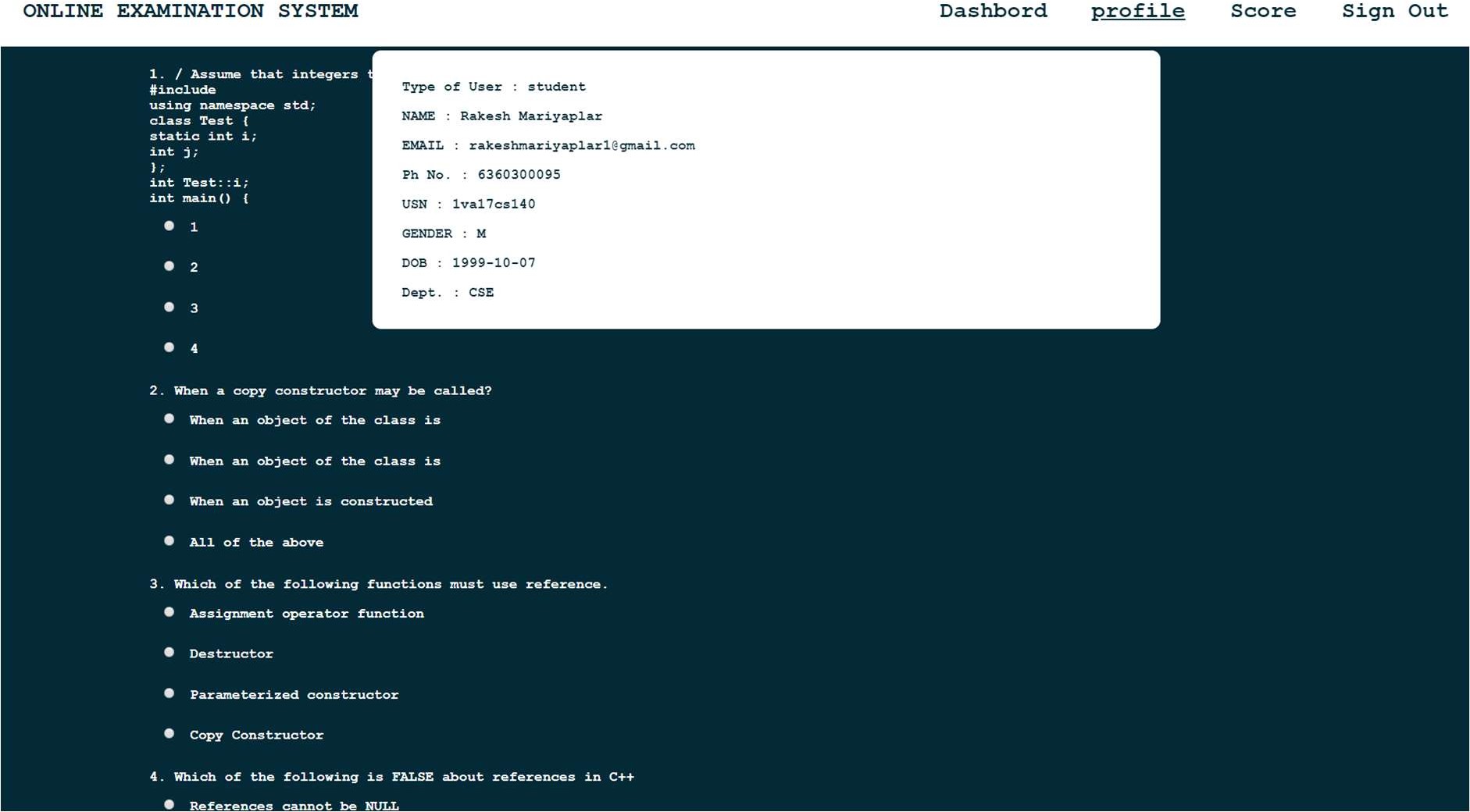
**Snapshot 2: Sign Up Page**



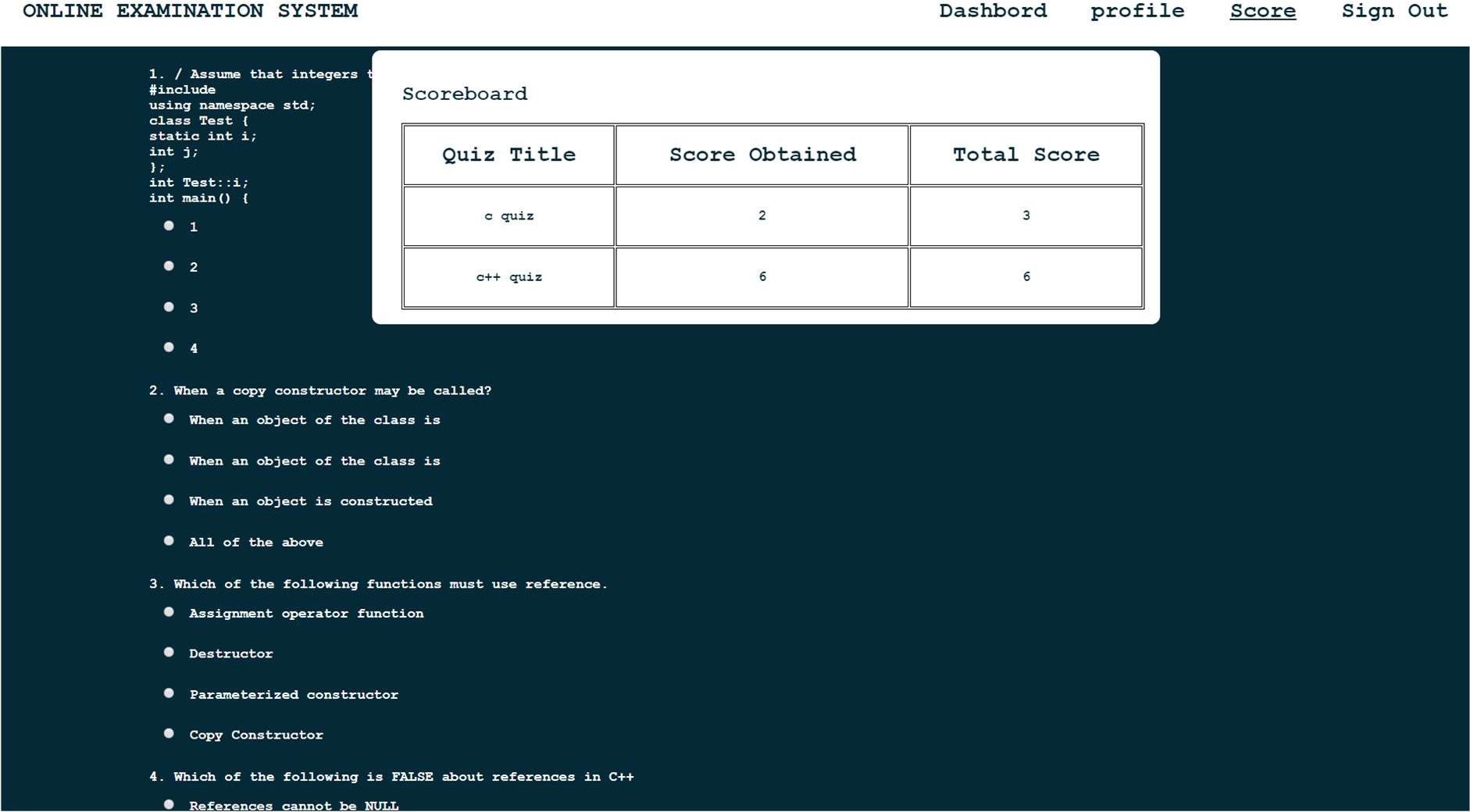
**Snapshot 2: Dashboard for Staff**



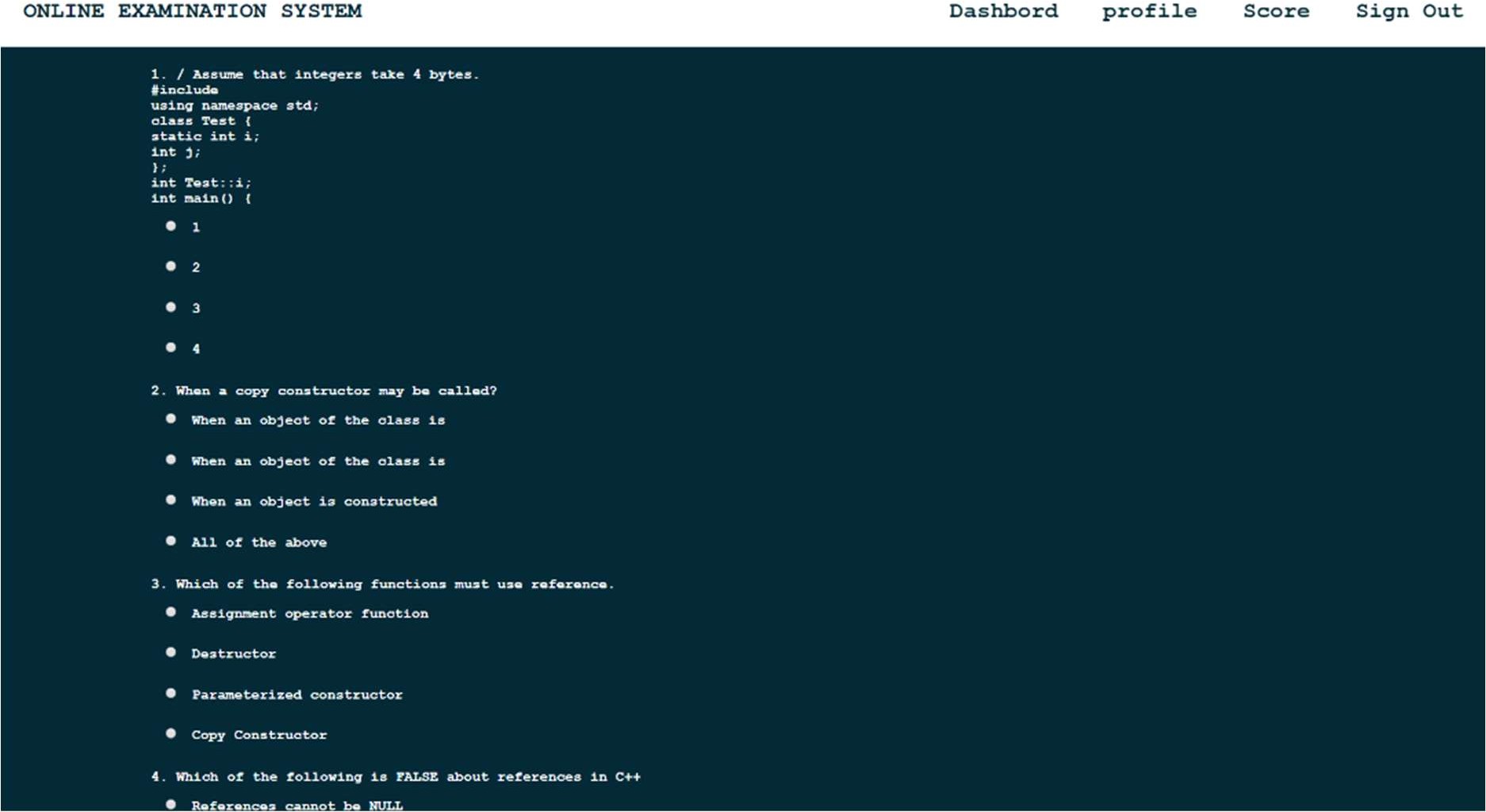
**Snapshot 3: Dashboard for Students**



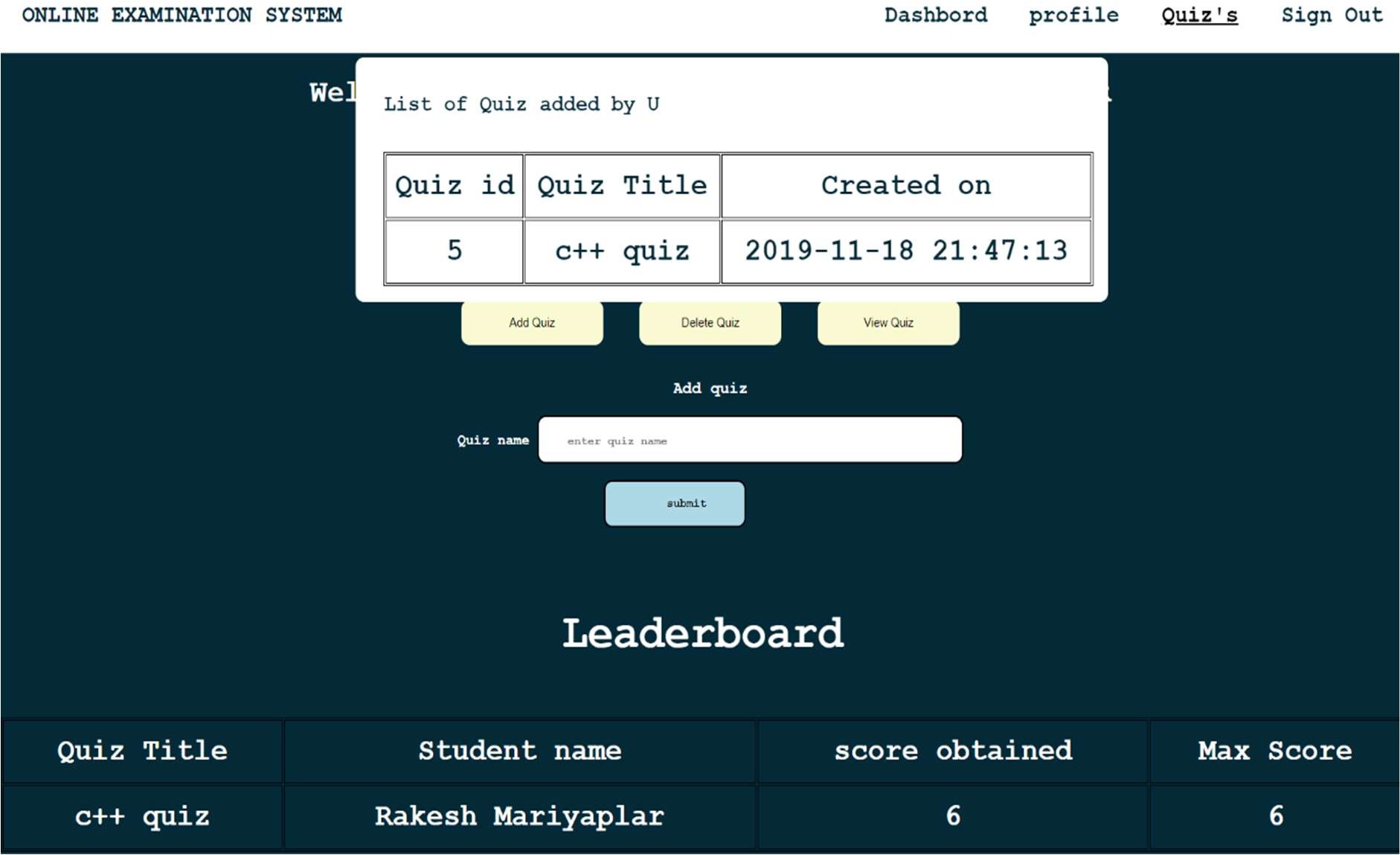
**Snapshot 4: Profile view of Student**



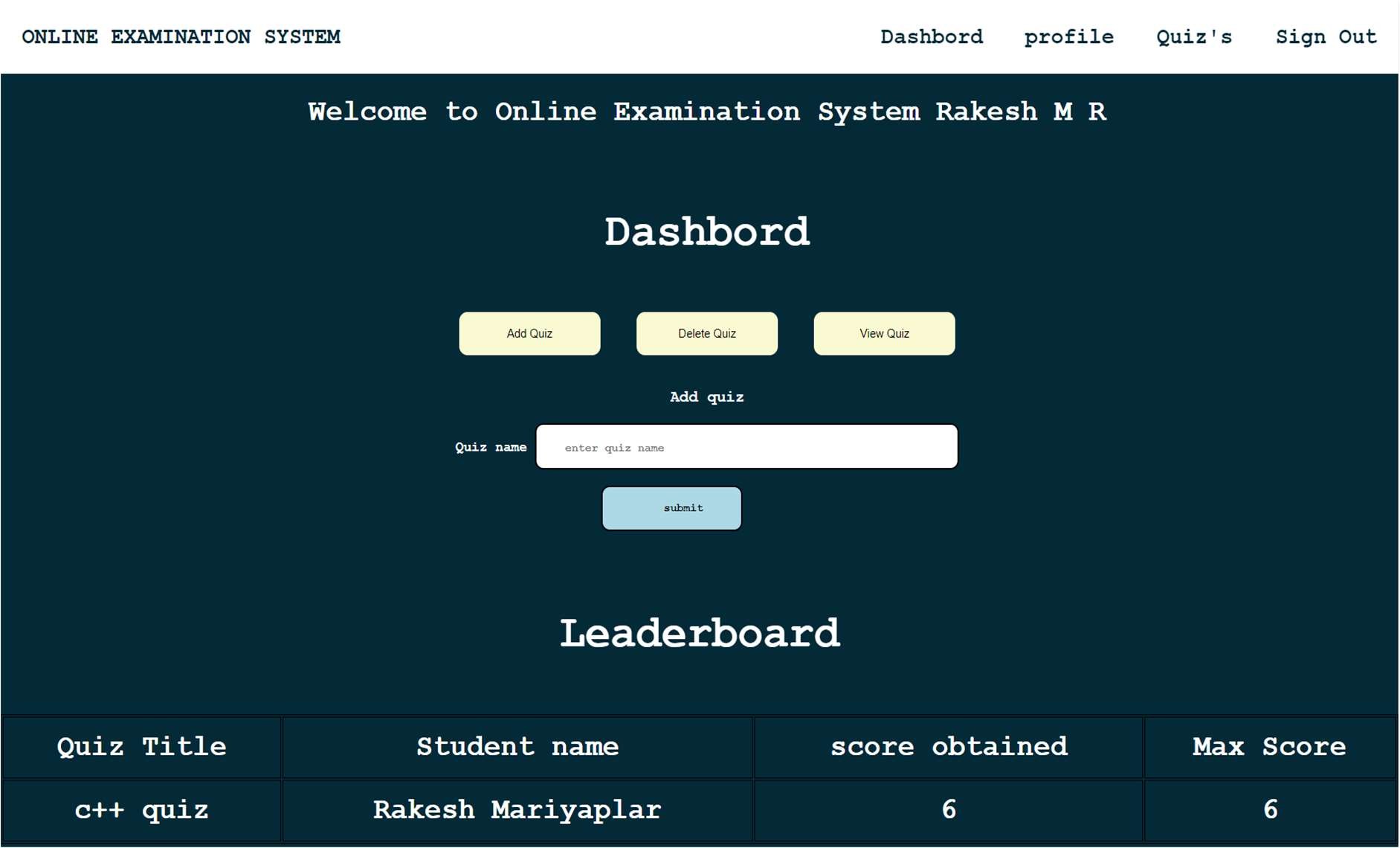
**Snapshot 5: Scoreboard of user**



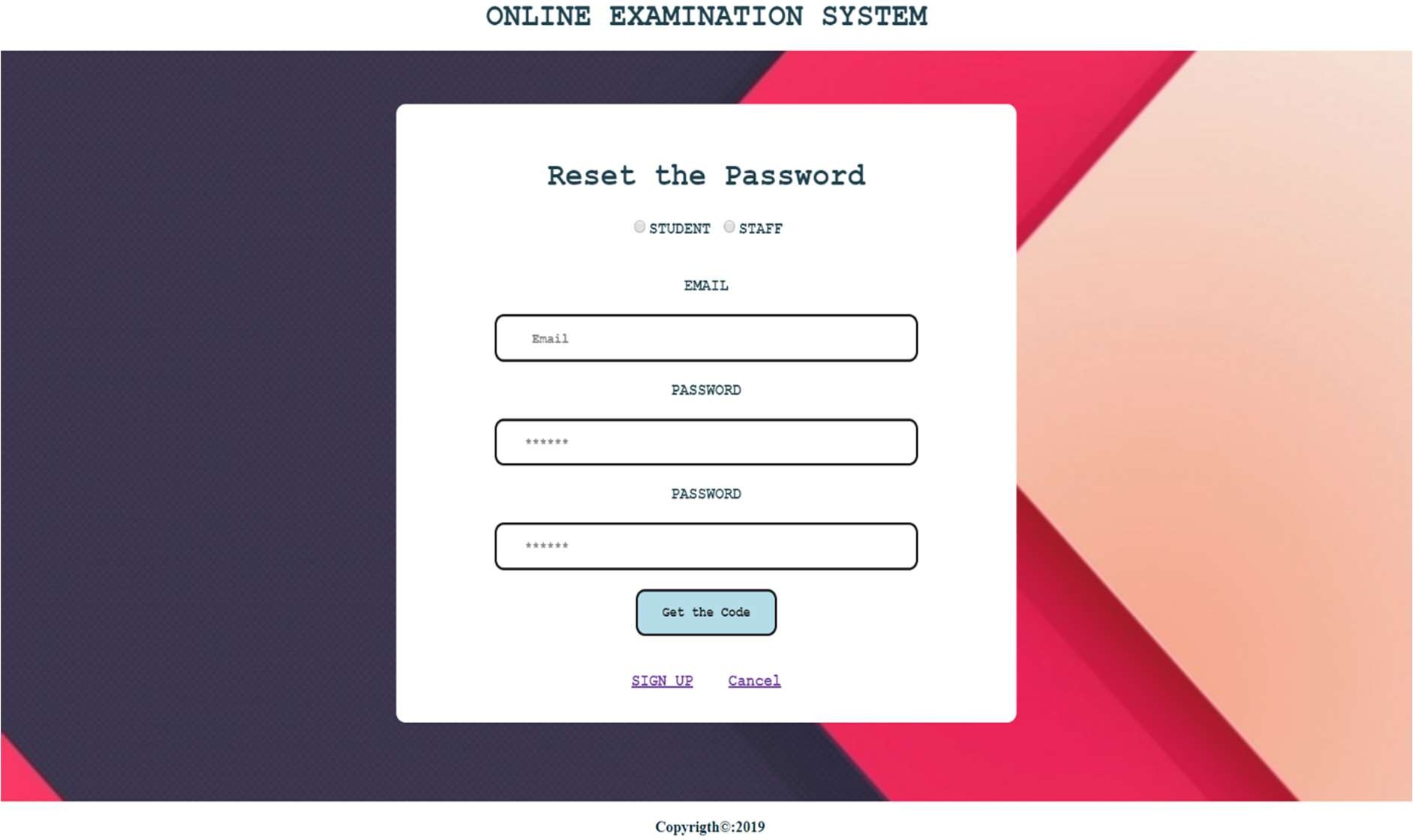
**Snapshot 6: Taking Quiz by Student**



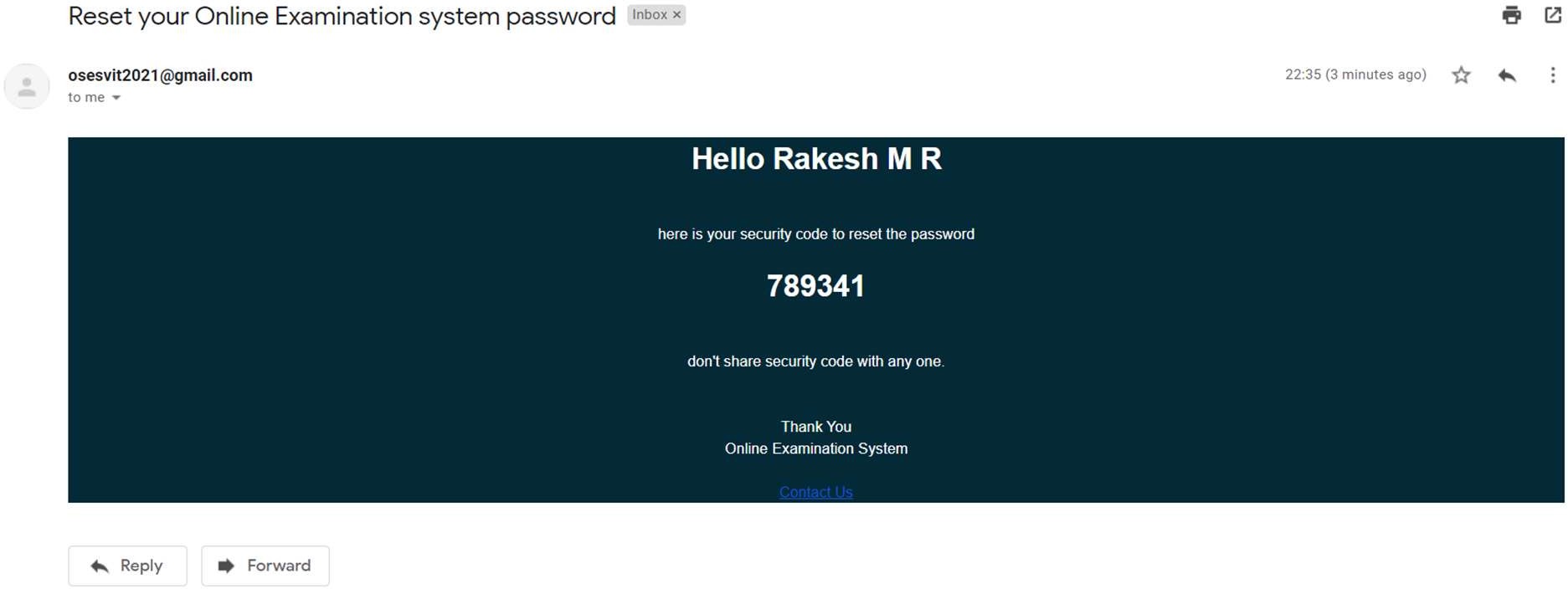
**Snapshot 7: Quiz Added by The Staff**



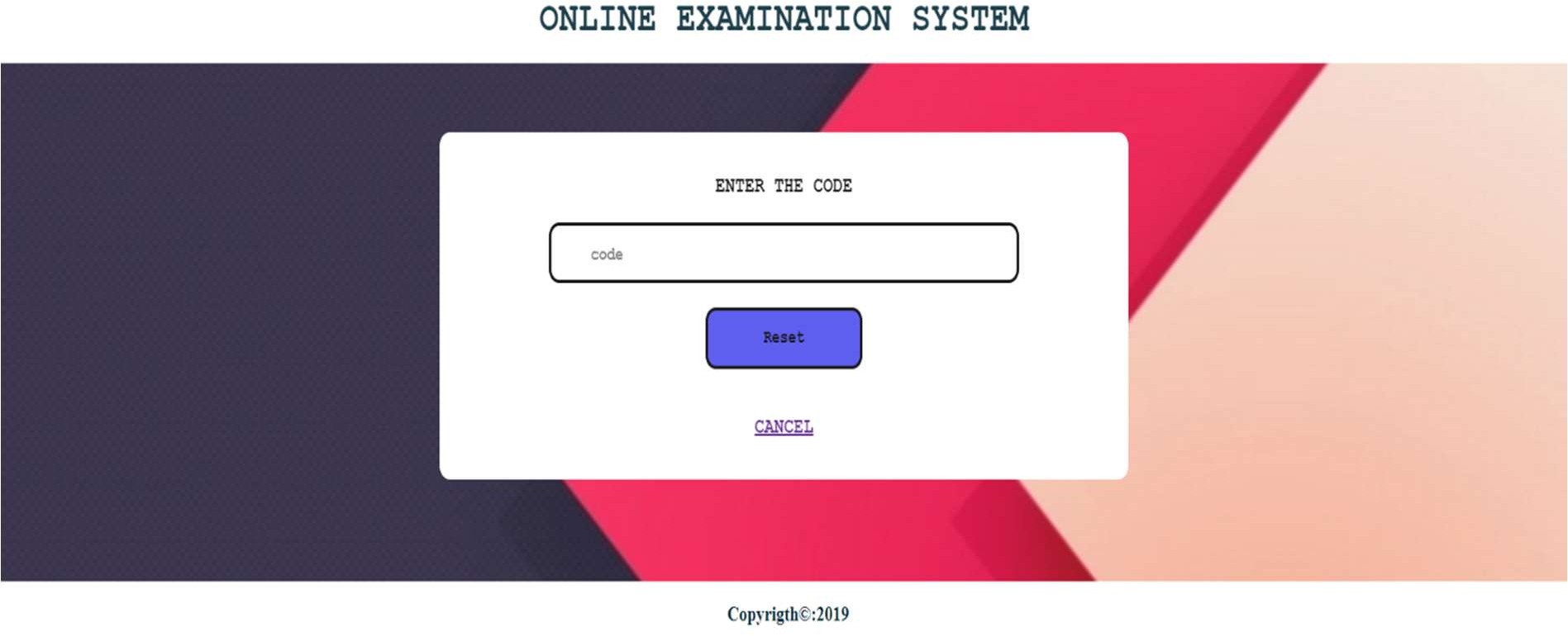
**Snapshot 8: Adding quiz page by staff**



**Snapshot 9: Request for security code**



**Snapshot 10: mail received by user with security code**



**Snapshot 11: verifying the security code and update password**