VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnanasangama", Belagavi-590 018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY

KR Road, VV Puram, Bengaluru- 560 004

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DATABASE MANAGEMENT SYSTEM MINI PROJECT

18CSL58

"ONLINE QUIZ MANAGEMENT SYSTEM"

Submitted by

ZAINAB DRABU

1BI19CS193

for the academic year 2021-22

Department of Computer Science and Engineering

Bangalore Institute of Technology

KR Road, VV Puram, Bengaluru- 560 004

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnanasangama", Belagavi-590 018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY

KR Road, VV Puram, Bengaluru- 560 004

Department of Computer Science and Engineering

Certificate

This is to certify that the implementation of DBMS MINI PROJECT entitled

"ONLINE QUIZ MANAGEMENT SYSTEM" has been successfully completed by **USN:1BI19CS193** **NAME: ZAINAB DRABU** of V semester B.E. for the partial fulfilment of the requirements for the Bachelor's degree in Computer Science & Engineering of the Visvesvaraya Technological University during the academic year 2020-2021.

Lab In-Charge

Prof. N THANUJA	**Dr. GIRIJA J**
Assistant Professor	Professor and Head
Dept of CS&E	Dept of CS&E
Bangalore Institute of Technology	Bangalore Institute of Technology
Bengaluru	Bengaluru

Examiners: 1) 2)

ACKNOWLEDGEMENT

The knowledge and satisfaction that accompany the successful completion of any task would be incomplete without mention of people who made it possible, whose guidance encouragement crowned my effort with success.

I would like to thank all and acknowledge the help I have received to carry out this Mini Project. I would like to convey my thanks to Head of Department Dr. J .GIRIJA for being kind enough to provide the necessary support to carry out the mini project.

I am most humbled to mention the enthusiastic influence provided by the lab in-charge Prof. N THANUJA on the project for her ideas, time to time suggestions for being a constant guide and co-operation showed during the venture and making this project a great success.

I would also take this opportunity to thank my friends and family for their constant support and help. I express my sincere gratitude to the friendly Science Department, BIT.

ZAIANB DRABU

1BI19CS193

LIST OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1. INTRODUCTION

1.1 INTRODUCTION

The **Simple Online Quiz System** is a project that can be used for educational purposes. This project will help faculty to manage the student quizzes records. Faculty can also monitor if the student already has taken the quiz. Faculty can create questions as much as he/she wants and answerable by choosing four options. Quizzes are restricted to the selected students who are listed by the faculty. Each option is randomly queued per question and so is the question per quizzes. The student can immediately know his/her score after submitting the quiz and also can review their answer sheet to know which question they answered wrong or right.

1.2 PROBLEM STATEMENT

To design and implement an Online quiz management system for the purpose of managing the quiz records, creating the quiz and taking the quiz online in an efficient way.

CHAPTER-2

BACK END DESIGN

2. BACKEND DESIGN

2.1 CONCEPTUAL DATABASE DESIGN

ER DIAGRAMs

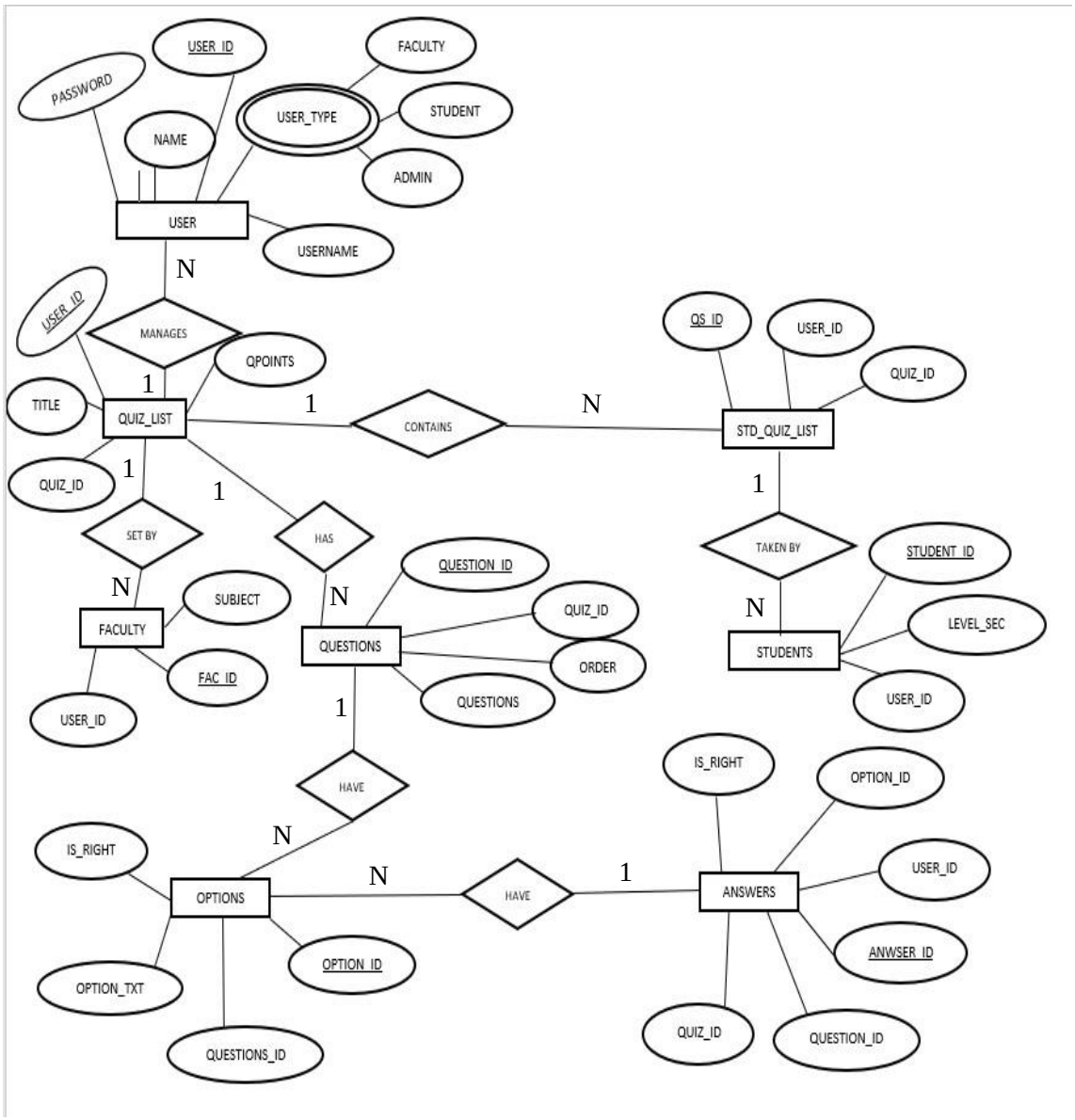


Fig 2.1.1: ER Diagram

2.2 LOGICAL DATABASE DESIGN

ER-MAPPING

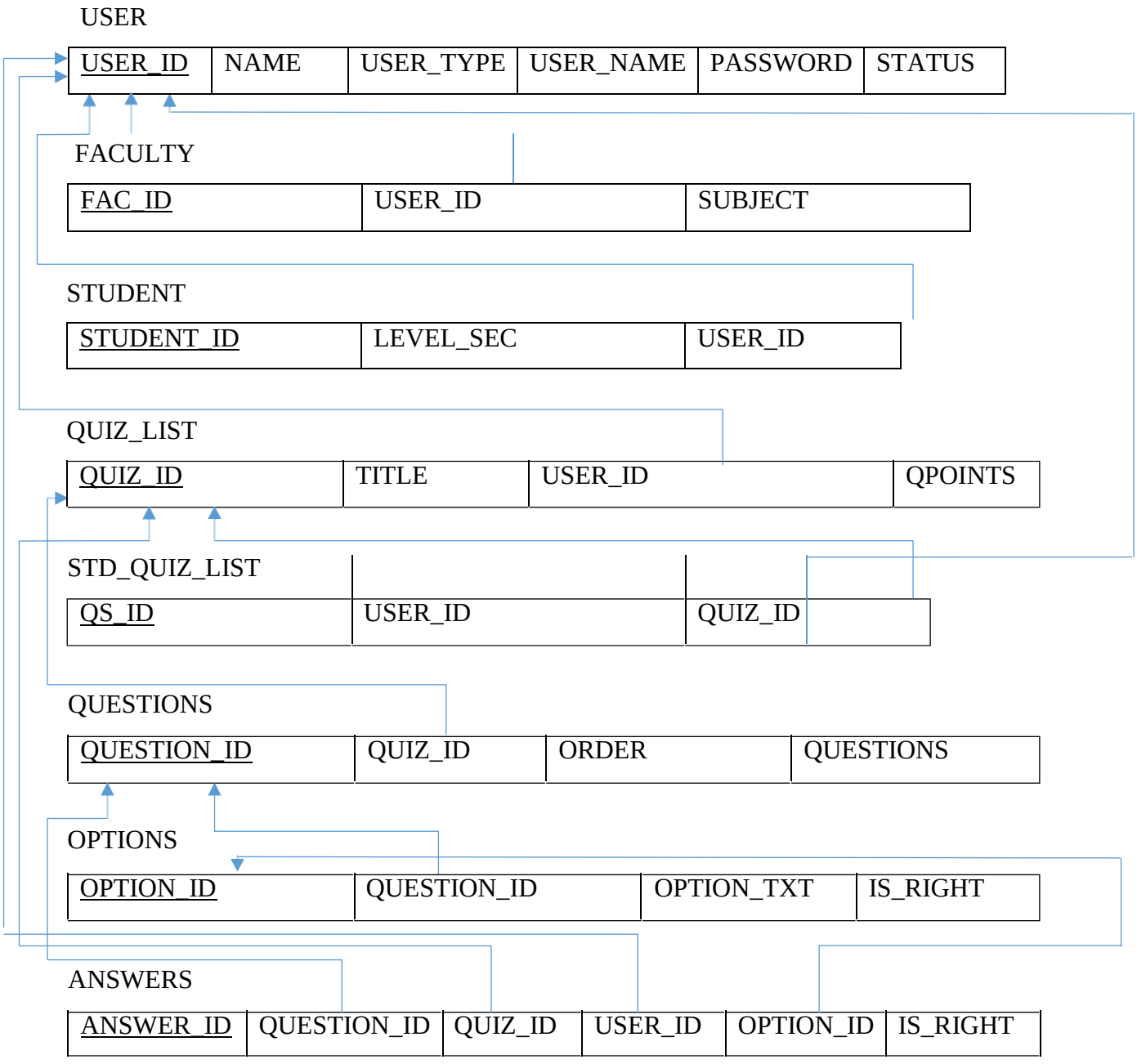


Fig 2.2.1: ER to Relational Mapping

2.3 NORMALIZATION

Database Normalization is a technique of organizing the data in database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like avoid insertion anomaly, update anomaly & deletion anomaly. It is a multistep process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

- Eliminating Redundant(useless) data
- Ensuring data dependencies make sense i.e data id logically stored

2.3.1 : CONDITIONS FOR NORMALIZATION First Normal Form(1NF):

As per First Normal Form, no two Rows of data must contain repeating group of information i.e. each set of columns must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique.

Second Normal Form (2NF):

As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails Second normal form.

Third Normal Form (3NF):

Third Normal form applies that every non-prime attribute of table must be dependent

on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So this transitive functional dependency should be removed from the table and also the table must be in Second normal form.

NORMALISED TABLES

2.3.2 ANSWERS

ANSWER_ID	USER_ID	QUIZ_ID	QUESTION_ID	OPTION_ID	IS_RIGHT

1NF
This is satisfied as all attributes are atomic

2NF
This is satisfied as there is no partial dependency

3NF
This is NOT satisfied as there is transitive dependency

{QUESTION_ID} -> {QUIZ_ID} (here exists transitive dependency)

QUESTION_ID	QUIZ_ID
Q101	QZ11
Q102	QZ11

{OPTION_ID} -> {IS_RIGHT} (here exists transitive dependency)

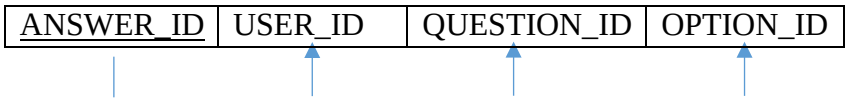
OPTION_ID	IS_RIGHT
O11	0
O12	1
O13	0
O14	0

Mapping between them

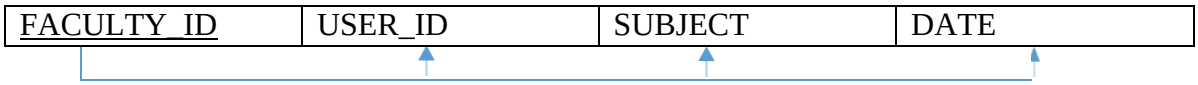
ANSWER_ID	QUESTION_ID	OPTION_ID
A12	Q101	O11
A13	Q102	O12

A14	Q103	013

Now the table is normalized.



2.3.3 FACULTY



1NF

This is satisfied as all attributes are atomic

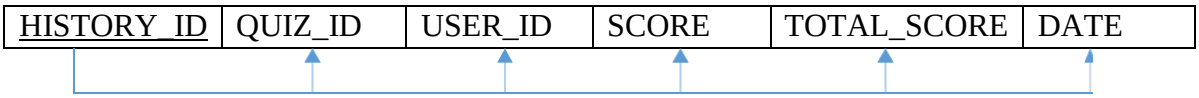
2NF

This is satisfied as there is no partial dependency

3NF

This is satisfied as there is no transitive dependency

2.3.4 HISTORY



1NF

This is satisfied as all attributes are atomic

2NF

This is satisfied as there is no partial dependency

3NF
This is satisfied as there is no transitive dependency

2.3.5 QUESTIONS

QUESTION_ID	QUESTION	Q_ID	ORDER_BY	DATE

1NF
This is satisfied as all attributes are atomic

2NF
This is satisfied as there is no partial dependency

3NF
This is satisfied as there is no transitive dependency

2.3.6 QUESTION_OPT

OPTIOIN_ID	OPTION_TXT	QUESTION_ID	IS_RIGHT	DATE

1NF
This is satisfied as all attributes are atomic

2NF
This is satisfied as there is no partial dependency

3NF
This is satisfied as there is no transitive dependency

2.3.6 QUIZ_LIST

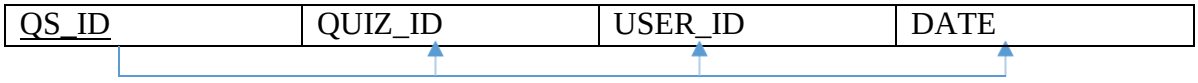
QUIZ_ID	TITLE	Q_POINTS	USER_ID	DATE

1NF
This is satisfied as all attributes are atomic

2NF
This is satisfied as there is no partial dependency

3NF
This is satisfied as there is no transitive dependency

2.3.7 QUIZ_STUDENT_LIST

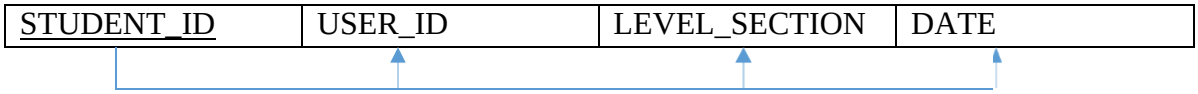


1NF
This is satisfied as all attributes are atomic

2NF
This is satisfied as there is no partial dependency

3NF
This is satisfied as there is no transitive dependency

2.3.8 STUDENTS



1NF
This is satisfied as all attributes are atomic

2NF
This is satisfied as there is no partial dependency

3NF
This is satisfied as there is transitive dependency

2.3.9 USERS

USER_ID	NAME	USER_TYPE	USERNAME	PASSWORD	STATUS	DATE

1NF

This is NOT satisfied as all attributes are not atomic.

1NF-1NF is not satisfied because user_type is multivalued attribute. So we should represent rows in such a way that there should not be more than one value for particular column, like

USER_ID	USER_TYPE
U123	2, 1
U117	3, 2

applying 1NF

USER_ID	USER_TYPE
U123	2
U123	1
U117	3
U117	2

2NF

This is satisfied as there is no partial dependency

3NF

This is satisfied as there is no transitive dependency

CHAPTER-3

FRONT END DESIGN

3. FRONT END DESIGN

HTML (Hyper Text Mark-up Language)

HTML is a standard mark-up language for creating web pages and web applications with Cascading Style Sheet (CSS) and JavaScript, it forms a triad of corner stone technologies of the World Wide Web.

CSS (Cascad*ing* Style Sheet)

CSS is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML. CSS is a corner stone technology of the World Wide Web, alongside HTML and JavaScript.

JAVASCRIPT

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Bootstrap

Bootstrap is the most popular CSS framework for developing responsive and mobile-first websites.

3.1 SCREEN LAYOUT DESIGN

HTML <form> TAG

The HTML <form> element represents a document section that contains interactive controls to submit information to a web server. It is possible to use the :valid and :invalid CSS pseudo-classes to style a <form>element. The HTTP method that the browser uses to submit the form.

Possible values are:

POST: Corresponds to the HTTP POST method; form data are included in the body of the form and sent to the server.

GET: Corresponds to the HTTP GET method; form data are appended to the action attribute URI with a '?' as separator, and the resulting URI is sent to the server. This method is used when the form has no side-effects and contains only ASCII characters. This value can be overridden by a form method attribute on a <button> or <input> element.

Action: The URI of a program that processes the form information. This value can be overridden by a form action attribute on a <button> or <input> element.

HTML <input> TAG

The HTML <input> element is used to create interactive controls for web-based forms in order to accept data from the user. An <input> work varies considerably depending on the value of its type attribute; hence the different types are covered in their own separate reference pages. If this attribute is not specified, the default type adopted type is text

3.2 CONNECTIVITY

In order to store or access the data inside a MySQL database, we first need to connect to the MySQL database server. PHP offers two different ways to connect to MySQL server: **MySQLi** (Improved MySQL) and **PDO** (PHP Data Objects) extensions.

While the PDO extension is more portable and supports more than twelve different databases, MySQLi extension as the name suggests supports MySQL database only. MySQLi extension however provides an easier way to connect to, and execute queries on, a MySQL database server. Here we have used MySQLi extension only.

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

CHAPTER-4

MAJOR MODULES

4. MAJOR MODULES

4.1 DESCRIPTION OF FUNCTIONALITIES

Login – This page helps to secure the project data and also to helps to manage users' access.

Home – This page is the welcoming page or the page where the user is redirected after logging in to the system. On this page also, quizzes summary can be seen.

Faculty – On this page, the administrator will manage the faculty list such as creating new and updating and deleting data of the faculty.

Student – At this. page, both faculty, and administrator can manage this page. This is where student data are listed and being managed.

Quiz List for Faculty – This page is where can faculty create quizzes and add students who can take the quiz. On this page, the Administrator can also manage quizzes.

Quiz List for Student – This page where student quizzes are listed.

Answer Sheet Page– This is the page where the student will answer their quiz.

Quiz Record Page – On this page, admin and faculty can see the records of the students each quiz

CHAPTER-5

IMPLEMENTATION

5. IMPLEMENTATION

5.1 DATABASE CODE

5.1.1 CREATION OF TABLES

```
 CREATE TABLE `answers` (
  `id` int(30) NOT NULL,
  `user_id` int(30) NOT NULL,
  `quiz_id` int(30) NOT NULL,
  `question_id` int(30) NOT NULL,
  `option_id` int(30) NOT NULL,
  `is_right` tinyint(1) NOT NULL COMMENT ' 1 = right, 0 = wrong',
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
CONSTRAINT ans PRIMARY KEY(ID),
CONSTRAINT ans FOREIGN KEY(QUESTION_ID) REFERENCES
 QUESTION(QUESTION_ID),
CONSTRAINT ans1 FOREIGN KEY(QUIZ_ID) REFERENCES
 QUIZ_LIST(QUIZ_ID),
CONSTRAINT ans2 FOREIGN KEY(ANSWER_ID) REFERENCES
 ANSWER(ANSWER_ID),
CONSTRAINT ans3 FOREIGN KEY(USER_ID) REFERENCES
 USER(USER_ID),
CONSTRAINT ans4 FOREIGN KEY(OPTION_ID) REFERENCES
 OPTION(OPTION_ID),
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `faculty` (
  `id` int(30) NOT NULL,
  `user_id` int(30) NOT NULL,
```

```
  `subject` varchar(100) NOT NULL,
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
CONSTRAINT fac PRIMARY KEY(ID),
CONSTRAINT fac1FOREIGN KEY(USER_ID) REFERENCES
 USER(USER_ID),
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `history` (
  `id` int(30) NOT NULL,
  `quiz_id` int(30) NOT NULL,
  `user_id` int(30) NOT NULL,
  `score` int(5) NOT NULL,
  `total_score` int(5) NOT NULL,
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
CONSTRAINT his PRIMARY KEY(ID)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `questions` (
  `id` int(30) NOT NULL,
  `question` text NOT NULL,
  `qid` int(30) NOT NULL,
  `order_by` int(11) NOT NULL`date_updated` datetime NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp(),
CONSTRAINT ques PRIMARY KEY(ID),
CONSTRAINT ques1 FOREIGN KEY(QUIZ_ID) REFERENCES
QUIZ_LIST(QUIZ_ID),
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `question_opt` (
  `id` int(30) NOT NULL,
```

```
  `option_txt` text NOT NULL,
  `question_id` int(30) NOT NULL,
  `is_right` tinyint(4) NOT NULL DEFAULT 0 COMMENT '1= right answer',
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
CONSTRAINT opt PRIMARY KEY(ID),
CONSTRAINT opt1 FOREIGN KEY(QUESTION_ID) REFERENCES
 QUESTION(QUESTION_ID),
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `quiz_list` (
  `id` int(30) NOT NULL,
  `title` varchar(200) NOT NULL,
  `qpoints` int(11) NOT NULL DEFAULT 1,
  `user_id` int(20) NOT NULL,
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
CONSTRAINT quiz PRIMARY KEY(ID),
CONSTRAINT quiz1 FOREIGN KEY(USER_ID) REFERENCES
USER(USER_ID),
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `quiz_student_list` (
  `id` int(30) NOT NULL,
  `quiz_id` int(30) NOT NULL,
  `user_id` int(30) NOT NULL,
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
CONSTRAINT qs PRIMARY KEY(ID),
CONSTRAINT qz1 FOREIGN KEY(QUIZ_ID) REFERENCES
 QUIZ_LIST(QUIZ_ID),
CONSTRAINT qz2  FOREIGN KEY(USER_ID) REFERENCES
 USER(USER_ID),
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `students` (
  `id` int(30) NOT NULL,
  `user_id` int(30) NOT NULL,
  `level_section` varchar(100) NOT NULL,
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
CONSTRAINT stu PRIMARY KEY(ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `users` (
  `id` int(30) NOT NULL,
  `name` varchar(150) NOT NULL,
  `user_type` tinyint(1) NOT NULL DEFAULT 1 COMMENT '1 = admin, 2= faculty , 3 = student',
  `username` varchar(25) NOT NULL,
  `password` varchar(25) NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 1 COMMENT ' 0 = incative , 1 = active',
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
CONSTRAINT user PRIMARY KEY(ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

5.2 Frontend Codes

Admin.php

```
<!DOCTYPE html>
<html>
<head>
        <?php include('header.php') ?>
    <?php
    session_start();
    if(isset($_SESSION['login_id'])){
        header('Location:home.php');
    }
    ?>
        <title>Admin | Simple Online Quiz System</title>
</head>

<body id='login-body' class="bg-light">

    <div class="card col-md-6 offset-md-3 text-center bg-primary mb-4">
        <h3 class="he3-responsive text-white">Simple Online Quiz System</h3>
    </div>
        <div class="card col-md-4 offset-md-4 mt-4">
            <div class="card-header-edge text-white">
                <strong>Login</strong>
            </div>
        <div class="card-body">
                <form id="login-frm">
                    <div class="form-group">
                        <label>Username</label>
                        <input type="username" name="username" class="form-control">
                    </div>
```

```
                <div class="form-group">
                    <input type="password" name="password" class="form-control">
                </div>
                <div class="form-group text-right">
                    <button class="btn btn-primary btn-block"
name="submit">Login</button>
                </div>

            </form>
        </div>
    </div>

    </body>

    <script>
        $(document).ready(function(){
            $('#login-frm').submit(function(e){
                e.preventDefault()
                $('#login-frm button').attr('disable',true)
                $('#login-frm button').html('Please wait...')

                $.ajax({
                    url:'./login_auth.php?type=1',
                    method:'POST',
                    data:$(this).serialize(),
                    error:err=>{
                        console.log(err)
                        alert('An error occured');
                        $('#login-frm button').removeAttr('disable')
                        $('#login-frm button').html('Login')
                    },
```

```
                    success:function(resp){
                        if(resp == 1){
location.replace('home.php')
                        }else{
                            alert("Incorrect username or password.")
                            $('#login-frm button').removeAttr('disable')
                            $('#login-frm button').html('Login')
                        }
                    }
                })

            })
        })
    </script>
</html>
```

home.php

```
<?php
    include 'db_connect.php';
   include 'auth.php';
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <?php include('header.php') ?>
    <title>Home | Simple Online Quiz System</title>
</head>
<body>
    <?php
```

```
include 'nav_bar.php';
?>
<div class="container-fluid admin">
    <div class="card col-md-5 offset-2">
        <div class="card-body">
            <table class="table table-striped">
                <thead>
                    <th>Quiz</th>
                    <th>Items</th>
                    <?php if($_SESSION['login_user_type'] == 3): ?>
                    <th>Status</th>
                    <?php else: ?>
                    <th>Had Taken</th>
                    <?php endif; ?>
                </thead>
                <tbody>
                    <?php
                        $where = '';
                        if($_SESSION['login_user_type'] == 2){
                            $where = " where u.id = ".$_SESSION['login_id']." ";
                        }
                        if($_SESSION['login_user_type'] == 3){
                            $where = " where q.id in (SELECT quiz_id from quiz_student_list where user_id = '".$_SESSION['login_id']."') ";
                        }
                        $qry = $conn->query("SELECT q.*,u.name as fname from quiz_list q left join users u on q.user_id = u.id ".$where." order by q.title asc ");
                            while($row= $qry->fetch_assoc()){
                                $items = $conn->query("SELECT count(id) as item_count from questions where qid = '".$row['id']."' ")->fetch_array()['item_count'];
                                $swhere ='';
```

```
                        if($_SESSION['login_user_type'] == 3)
                          $swhere= ' and user_id = '.$_SESSION['login_id'].' ';

                        $taken = $conn->query("SELECT count(id) as item_count from
answers where quiz_id = '".$row['id']."'  ".$swhere )->fetch_array()['item_count'];
                    ?>
                    <tr>
                    <td><?php echo $row['title'] ?></td>
                    <td class='text-center'><?php echo $items ?></td>
                    <?php if($_SESSION['login_user_type'] == 3): ?>
                    <td class='text-center'><?php echo $taken > 1 ? 'Taken' : 'Pending' ?></td>
                    <?php else: ?>
                    <td class='text-center'><?php echo $taken ?></td>
                    <?php endif; ?>
                    </tr>
                    <?php
                    }

                    ?>
                  </tbody>
                </table>
              </div>
            </div>
          </div>
</body>
<script>
$(document).ready(function(){
        const
})
</script>
</html>
```

Faculty.php

```
<!DOCTYPE html>
<html lang="en">
<head>
        </head>
<?php include('header.php') ?>
        <?php include('auth.php') ?>
        <?php include('db_connect.php') ?>
        <title>Faculty List</title>
</head>
<body>
        <?php include('nav_bar.php') ?>

        <div class="container-fluid admin">
                <div class="col-md-12 alert alert-primary">Faculty List</div>
                <button class="btn btn-primary bt-sm" id="new_faculty"><i class="fa fa-
plus"></i>       Add New</button>
                <br>
                <br>
                <div class="card">
                        <div class="card-body">
                                <table class="table table-bordered" id='table'>
                                        <colgroup>
                                                <col width="10%">
                                                <col width="40%">
                                                <col width="30%">
                                                <col width="20%">
                                        </colgroup>
                                        <thead>
                                                <tr>
                                                        <th>#</th>
```

```
                                        <th>Name</th>
                                        <th>Subject</th>
                                        <th>Action</th>
                                    </tr>
                                </thead>
                                <tbody>
<?php
                                $qry = $conn->query("SELECT f.*,u.name from faculty f left join users u  on f.user_id = u.id order by u.name asc ");
                                $i = 1;
                                if($qry->num_rows > 0){
                                    while($row= $qry->fetch_assoc()){
                                    ?>
                                <tr>
                                    <td><?php echo $i++ ?></td>
                                    <td><?php echo $row['name'] ?></td>
                                    <td><?php echo $row['subject'] ?></td>
                                    <td>
                                        <center>
                                         <button class="btn btn-sm btn-outline-primary edit_faculty" data-id="<?php echo $row['id']?>" type="button"><i class="fa fa-edit"></i> Edit</button>
                                        <button class="btn btn-sm btn-outline-danger remove_faculty" data-id="<?php echo $row['id']?>" type="button"><i class="fa fa-trash"></i> Delete</button>
                                        </center>
                                    </td>
                                </tr>
                                <?php
                                }
                                }
                                ?>
```

```
                                    </tbody>
                                </table>
                            </div>
                        </div>
                    </div>
                    <div class="modal fade" id="manage_faculty" tabindex="-1" role="dialog" >
                                <div class="modal-dialog modal-centered" role="document">
<div class="modal-content">
                                        <div class="modal-header">

                                            <h4 class="modal-title"
id="myModallabel">Add New Faculty</h4>
                                            <button type="button" class="close"
data-dismiss="modal" aria-label="Close"><span aria-
hidden="true">&times;</span></button>
                                        </div>
                                        <form id='faculty-frm'>
                                            <div class ="modal-body">
                                                <div id="msg"></div>
                                                <div class="form-group">
                                                    <label>Name</label>
                                                    <input type="hidden"
name="id" />
                                                    <input type="hidden"
name="uid" />
                                                    <input type="hidden"
name="user_type" value = '2' />
                                                    <input type="text"
name="name" required="required" class="form-control" />
                                                </div>
                                                <div class="form-group">
                                                    <label>Subject</label>
```

```
                                        <input type="text" name
="subject" required="" class="form-control" />
                                    </div>
                                    <div class="form-group">

        <label>Username</label>
                                        <input type="text" name
="username" required="" class="form-control" />
                                    </div>
<div class="form-group">

        <label>Password</label>
                                        <input type="password"
name="password" required="required" class="form-control" />
                                    </div>
                                </div>
                                <div class="modal-footer">
                                        <button class="btn btn-
primary" name="save"><span class="glyphicon glyphicon-save"></span> Save</button>
                                    </div>
                                </form>
                            </div>
                        </div>
                    </div>
</body>
<script>
        $(document).ready(function(){
                $('#table').DataTable();
                $('#new_faculty').click(function(){
                        $('#msg').html('')
                        $('#manage_faculty .modal-title').html('Add New Faculty')
                        $('#manage_faculty #faculty-frm').get(0).reset()
```

```
                $('#manage_faculty').modal('show')
        })
        $('.edit_faculty').click(function(){
                var id = $(this).attr('data-id')
                $.ajax({
                        url:'./get_faculty.php?id='+id,
                        error:err=>console.log(err),
                        success:function(resp){
                                if(typeof resp != undefined){
resp = JSON.parse(resp)
                                        $('[name="id"]').val(resp.id)
                                        $('[name="uid"]').val(resp.uid)
                                        $('[name="name"]').val(resp.name)
                                        $('[name="subject"]').val(resp.subject)
                                        $('[name="username"]').val(resp.username)
                                        $('[name="password"]').val(resp.password)
                                        $('#manage_faculty .modal-title').html('Edit
Faculty')
                                        $('#manage_faculty').modal('show')

                                }
                        }
                })

        })
        $('.remove_faculty').click(function(){
                var id = $(this).attr('data-id')
                var conf = confirm('Are you sure to delete this data.');
                if(conf == true){
                        $.ajax({
                        url:'./delete_faculty.php?id='+id,
```

```
                        error:err=>console.log(err),
                        success:function(resp){
                                if(resp == true)
                                        location.reload()
                        }
                })
                }
        })
        $('#faculty-frm').submit(function(e){
                e.preventDefault();
$('#faculty-frm [name="submit"]').attr('disabled',true)
                $('#faculty-frm [name="submit"]').html('Saving...')
                $('#msg').html('')

                $.ajax({
                        url:'./save_faculty.php',
                        method:'POST',
                        data:$(this).serialize(),
                        error:err=>{
                                console.log(err)
                                alert('An error occured')
                                $('#faculty-frm
[name="submit"]').removeAttr('disabled')
                                $('#faculty-frm [name="submit"]').html('Save')
                        },
                        success:function(resp){
                                if(typeof resp != undefined){
                                        resp = JSON.parse(resp)
                                        if(resp.status == 1){
                                                alert('Data successfully saved');
                                                location.reload()
```

```
        }else{
                                        $('#msg').html('<div class="alert alert-
danger">'+resp.msg+'</div>')

                                        }
                                    }
                                }
                            })
                        })
        })
</script>
</html>
```

CHAPTER-6

SNAPSHOTS

6. SNAPSHOTS

6.1 SNAPSHOTS

ADMIN PAGE: consists of an admin sign in

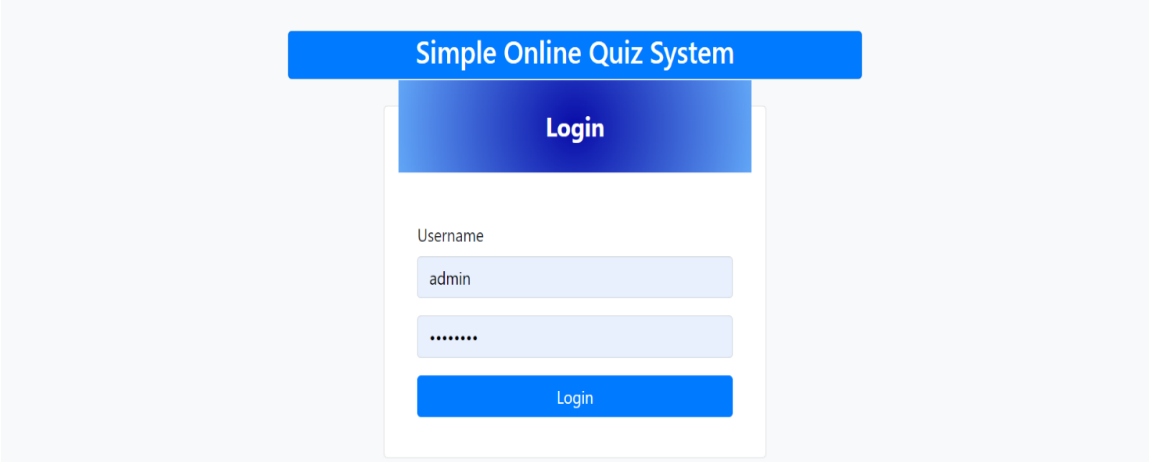


Fig 6.1.1: Admin page

HOME PAGE: consists of 3 fields quiz, items(questions), Had taken(students)

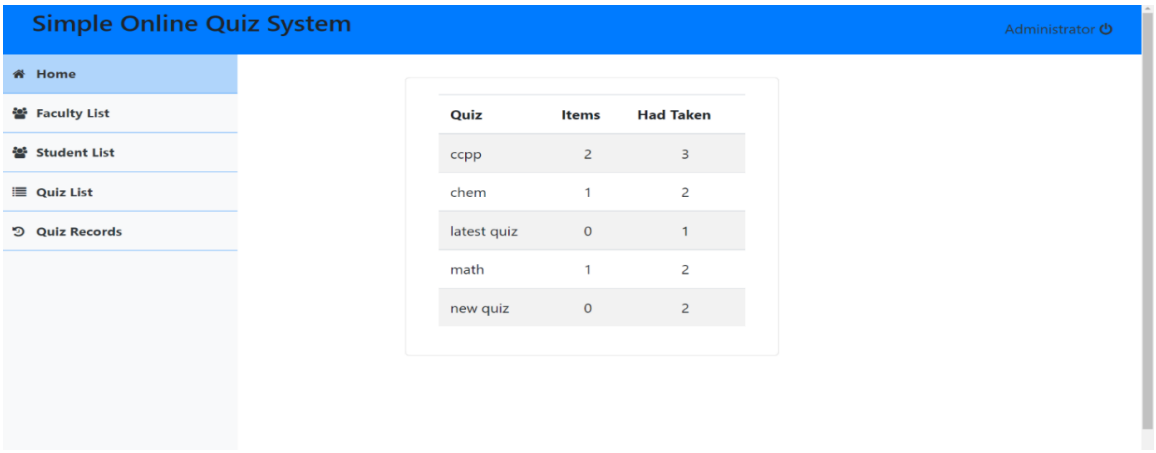


Fig 6.1.2: Home page

FACULTY LIST: displays name, subject and action.

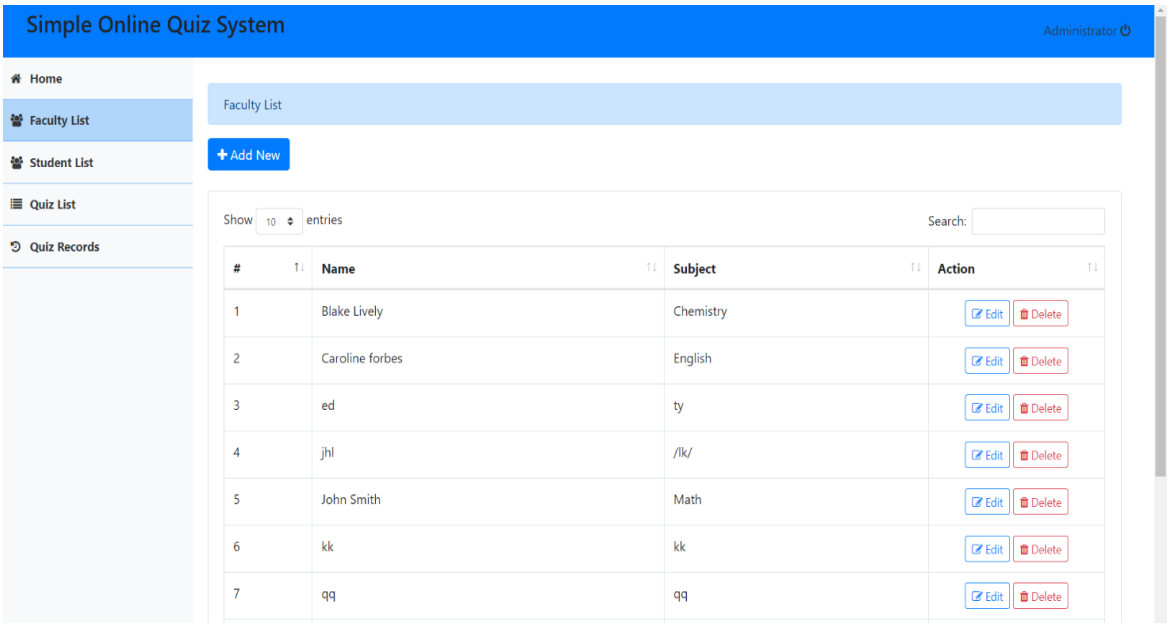


Fig 6.1.3: Faculty list

UNDER ACTION: we can delete or edit faculty members

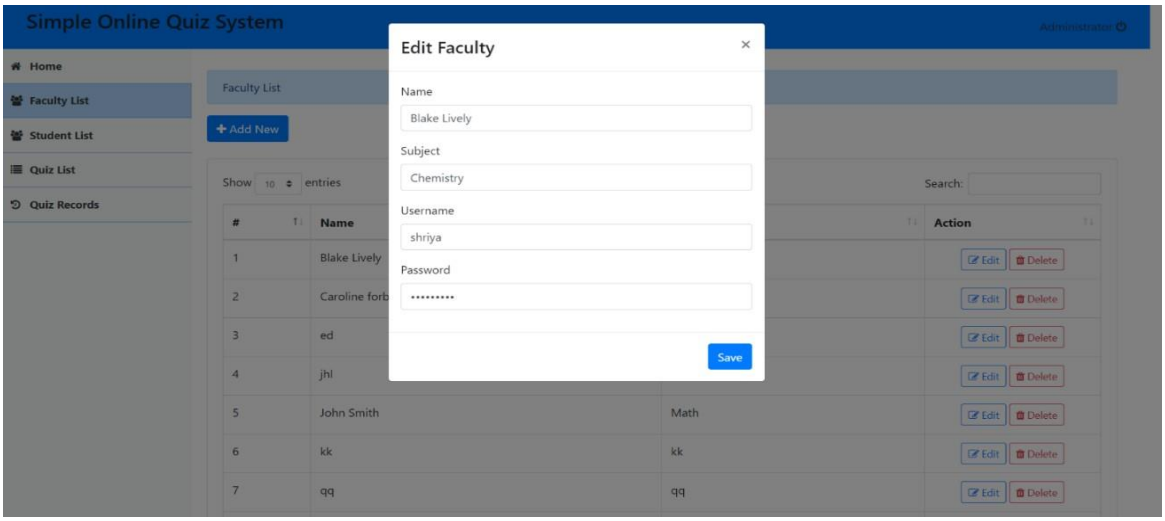


Fig 6.1.4: Editing faculty members

ADD NEW: can add new faculty member

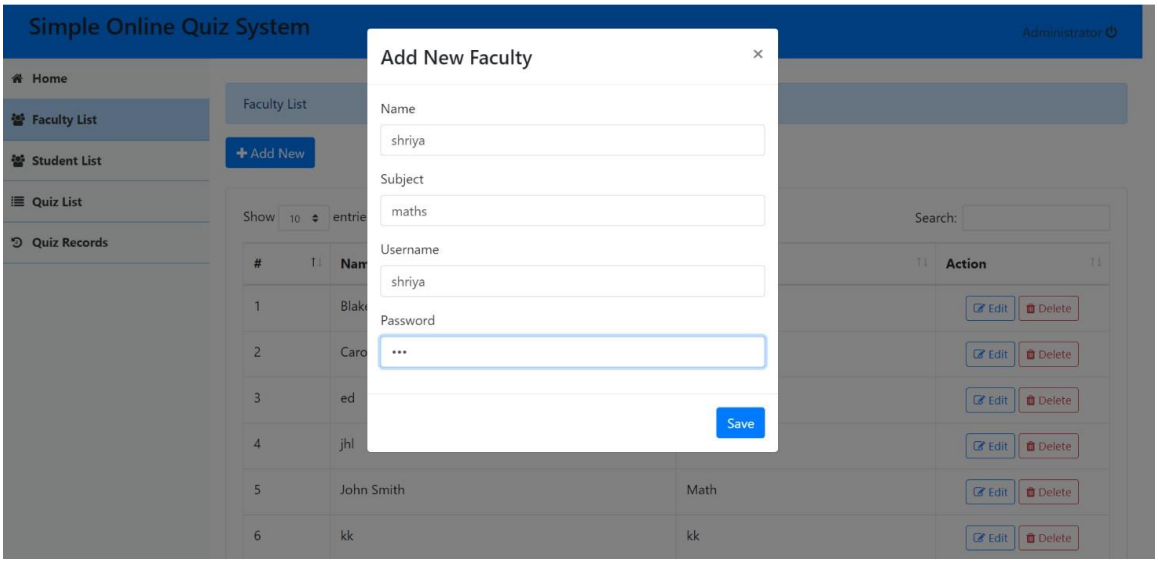


Fig 6.1.5: Adding new faculty

FACULTY LOGIN: consist of faculty login

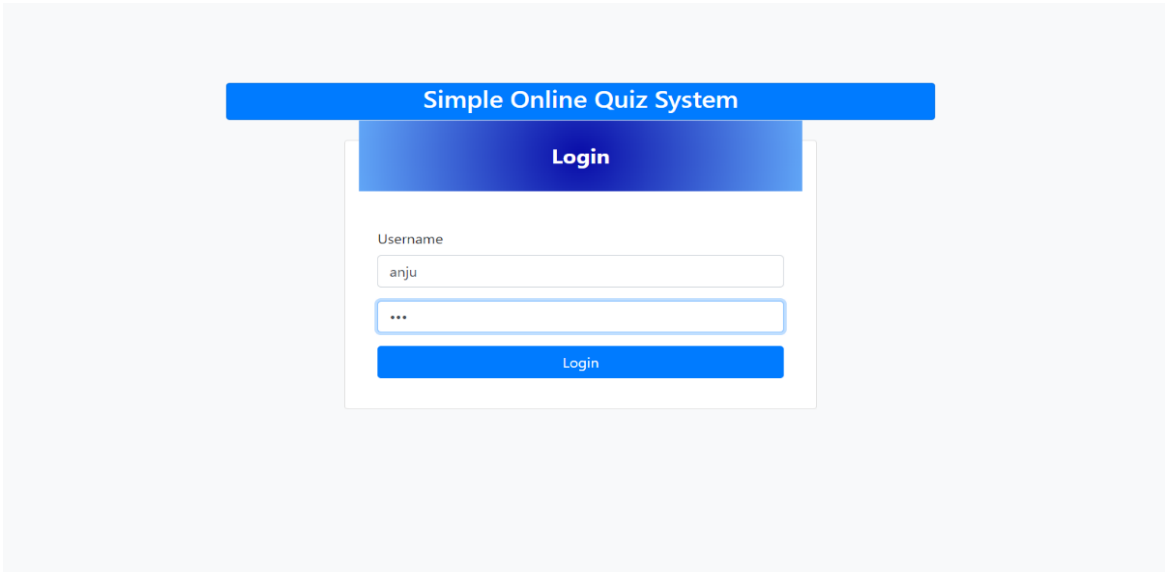


Fig 6.1.6: Faculty login

FACULTY HOME: displays of quiz(subject), item(questions), had taken(student)

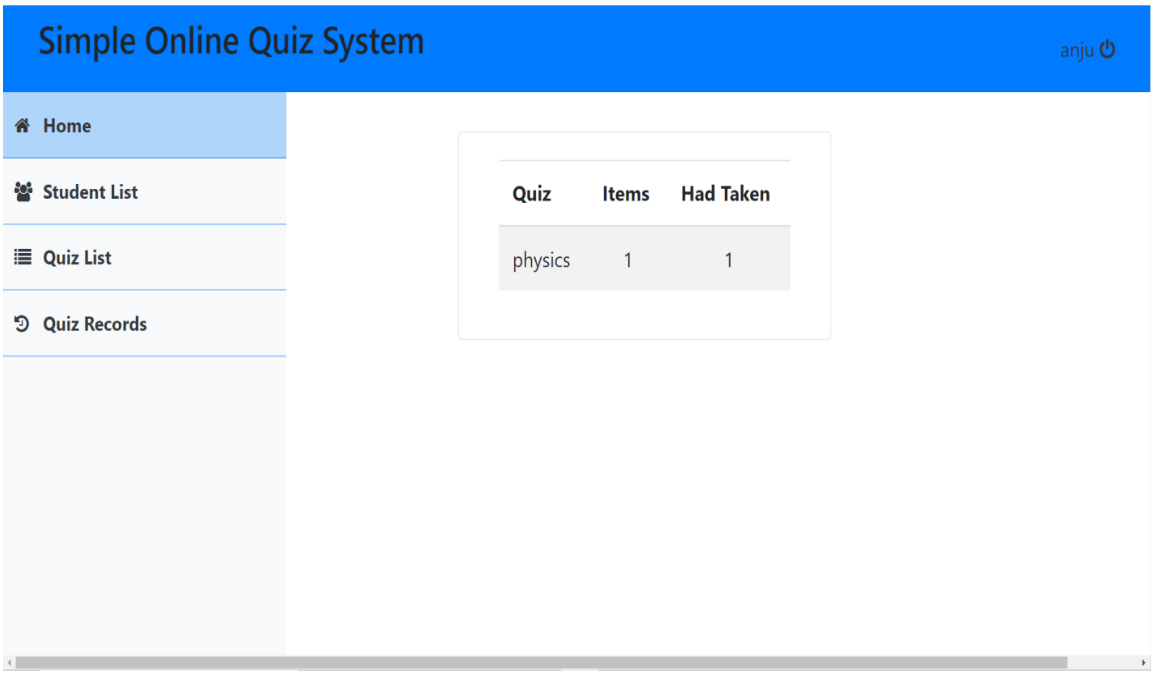


Fig 6.1.7: Faculty home page

STUDENT LOGIN: consist of student login

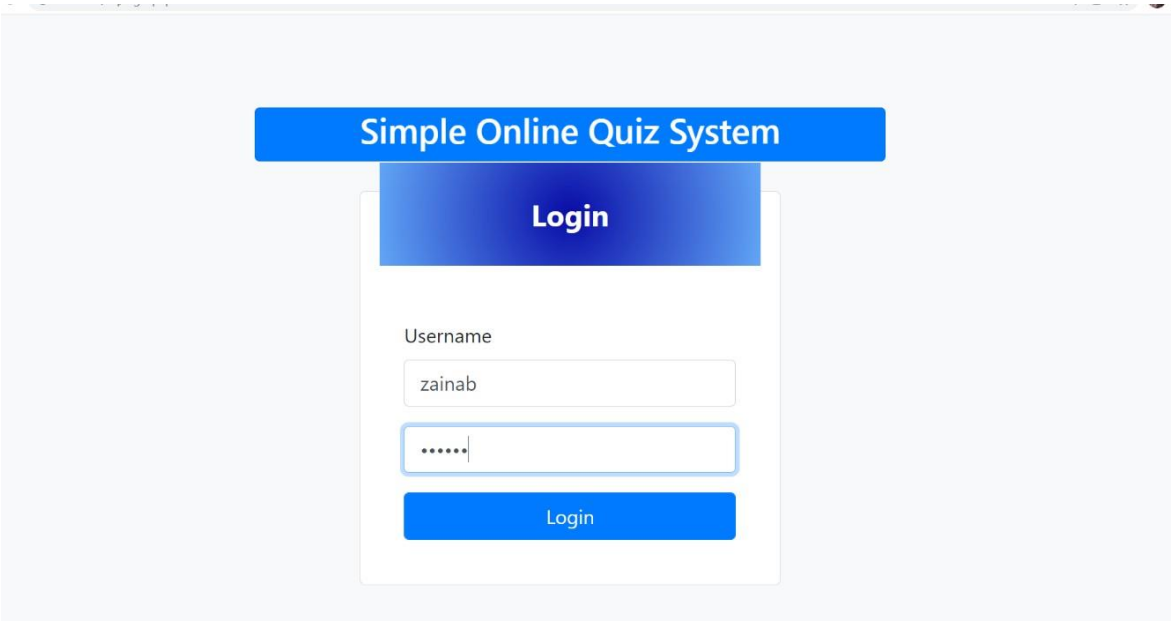


Fig 6.1.8: Student login

UNDER STUDENT QUIZ LIST: display quiz, score, status, and action of student.

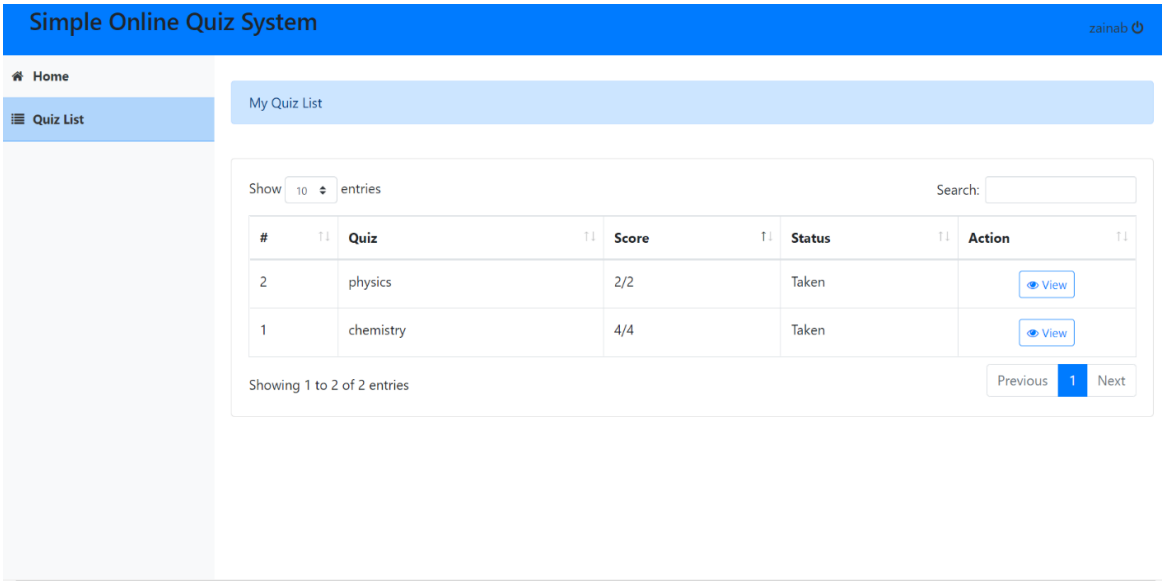


Fig 6.1.9: Student quiz list

UNDER ACTION: student can either view or attempt quiz

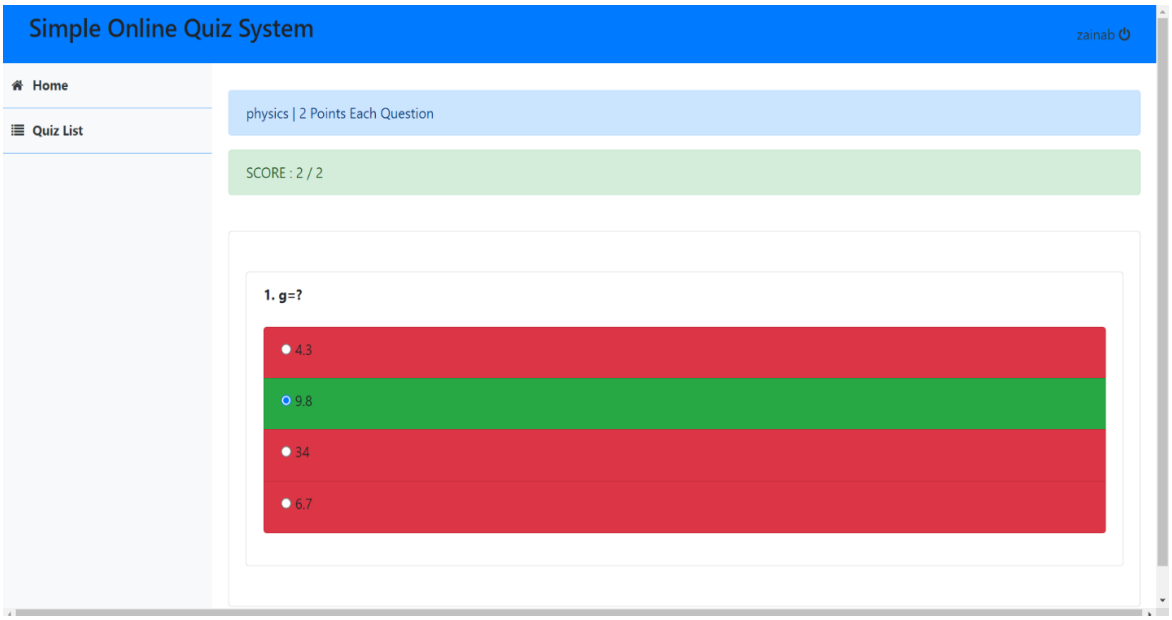


Fig 6.1.10 (a): Student viewing quiz

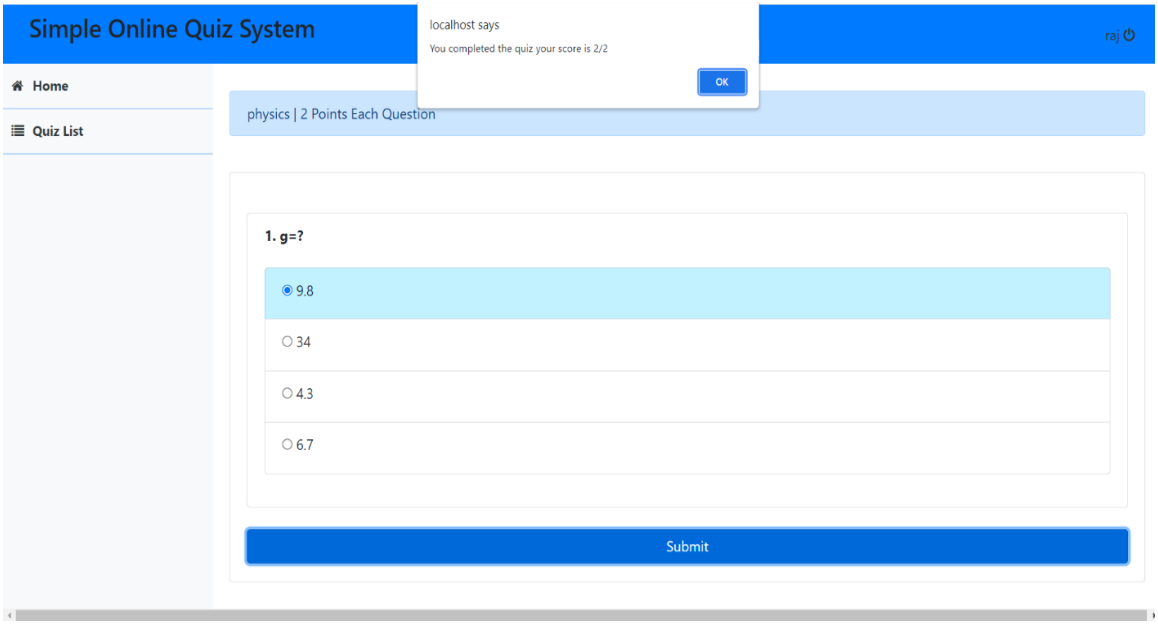


Fig 6.1.10 (b): Student attempting quiz

QUIZ LIST: displays title, item, points per item, faculty and action

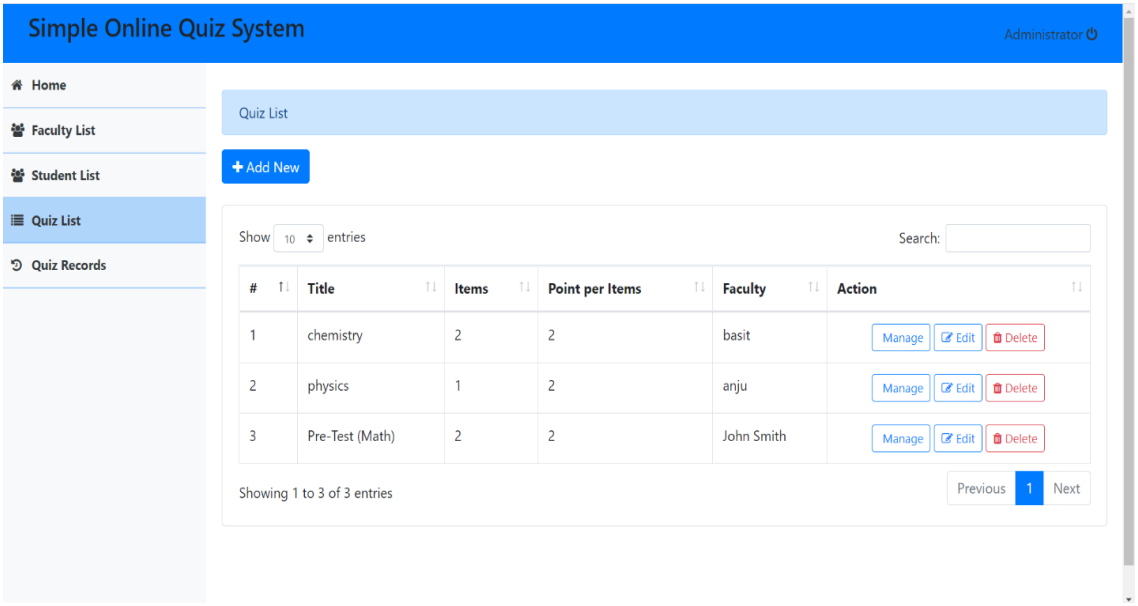


Fig 6.1.11: Quiz list

UNDER MANAGE: can add questions and students attempting those questions

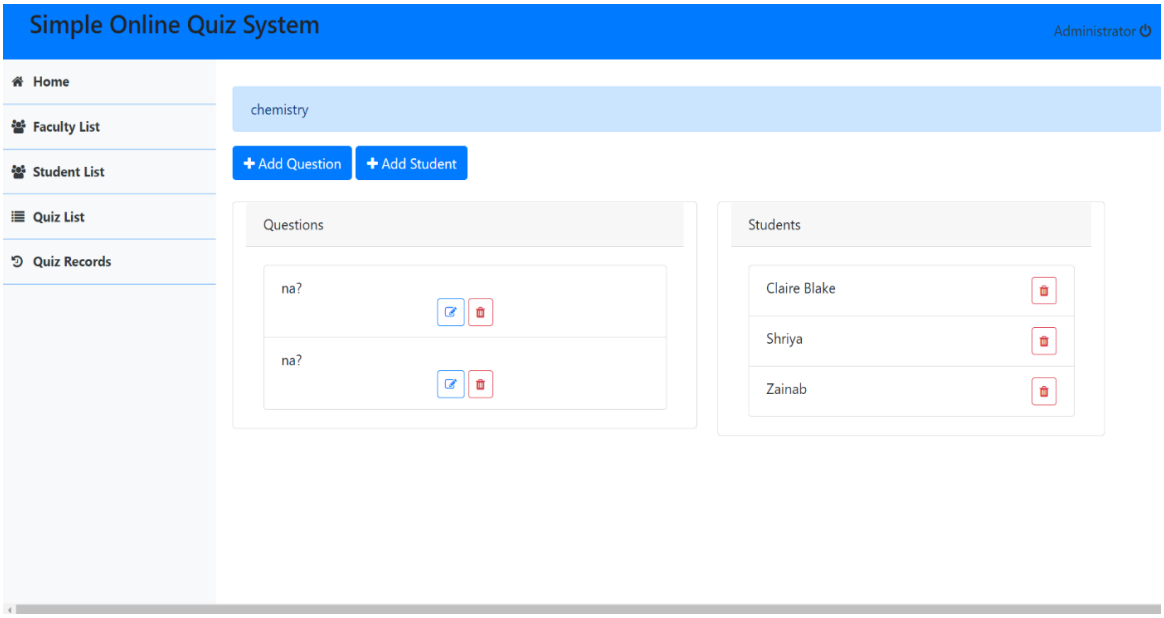


Fig 6.1.12: Adding of questions and students

QUIZ RECORD: displays student name, quiz and final score of student

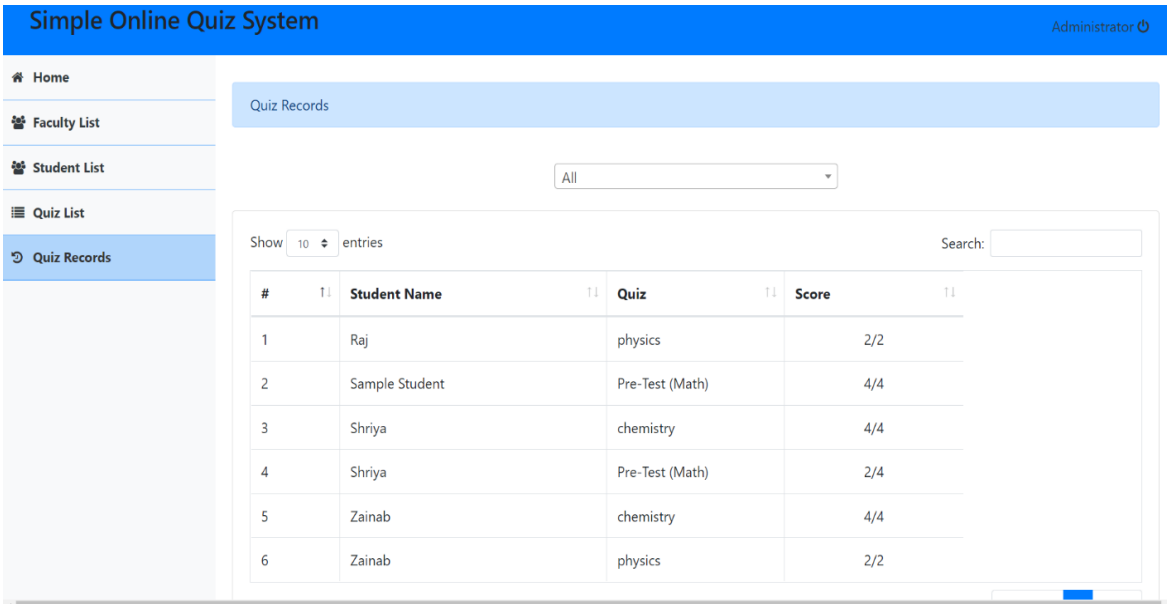


Fig 6.1.13: Quiz record

CHAPTER-7

APPLICATIONS

7. APPLICATIONS

7.1 APPLICATIONS OF ONLINE QUIZ MANAGEMENT SYSTEM

- It tracks all the information of the examination, papers, students, etc.
- Manages the information of examination.
- Shows the information and description of courses and marks.
- All the fields such as courses marks results are validated and does not take invalid values.
- This also provides the searching facilities based on various factors such as courses marks, students and results.
- Time efficient since the entire process is online.

CHAPTER-8

CONCLUSION

8. CONCLUSION

8.1 CONCLUSION

Technology is introducing new innovations day by day, thus reducing the time required to do things. The proposed system is developed for educational purposes allowing the user to prepare for multiple choice questions for different examinations conducted on national and provincial level. This system reduces the heavy paper work needed for documenting the information and makes the process efficient and transparent.

Developing this project came with enormous amount of learning, enabling me to acquire new skills as well as experimenting with my already acquired skills. It improved my skills in database management system helping me learn various perks of organized data.

CHAPTER-9

BIBLIOGRAPHY

9. BIBLIOGRAPHY

9.1 BOOK REFERENCES

1. Learn to Code HTML and CSS: Develop and Style Websites (Web Design Courses) 1st, Kindle Edition by Shay Howe

2. *PHP 6 and MySQL 5* - Larry Ullman

9.2 WEBSITE REFERENCES

HTML Learning

1. https://www.codecademy.com/

2. https://dash.generalassemb.ly/

3. https://www.w3schools.com/

PHP Learning

1. http://www.tutorialspoint.com/php/

2. https://killerphp.com

3. https://www.w3schools.com/