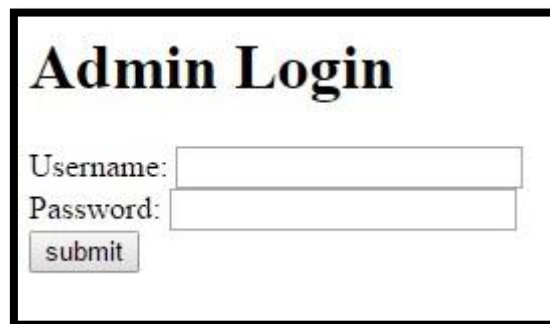


## Quiz

The QUIZ is a web application for to take online test in an efficient manner and no time wasting for checking the paper. The main objective of QUIZ is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves lot of time but also gives fast results. For students they give papers according to their convenience and time. This can be used in educational institutions as well as in corporate world. Can be used anywhere any time as it is a web based application (user location doesn't matter). No restriction that examiner has to be present when the candidate takes the test.

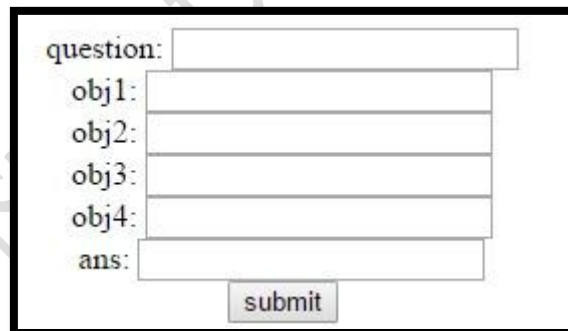
## Admin Side

In [index.php](#) the admin will get a link as admin login on click of which login block will open if admin enters proper login credentials and clicks on login button, the control will be transferred to [log.php](#) using **javascript-ajax** which will check whether entered username and password is correct, if yes it will redirect to [addQuestion.php](#)



The image shows a web form titled "Admin Login". It contains two input fields: "Username:" and "Password:". Below the password field is a "submit" button. The form is enclosed in a black border.

The [addQuestion.php](#) will have questions, option and answer filling. **Note: Interface is kept very simple.**



The image shows a web form for adding a question. It contains five input fields: "question:", "obj1:", "obj2:", "obj3:", and "obj4:". Below these is an "ans:" field and a "submit" button. The form is enclosed in a black border.

Very  
Important

**All of this should be stored in the database; the correct option should be copied in the answer tab.**

## User Side

The user should be logged in, in order to give the quiz. If the user is not registered, he should register on [register.php](#)

After Successful login, he should be able to take a quiz and can also see past test results. This is the user home page. Also provide [logout](#) option, preferably on the extreme top right corner !

## Web Tech - Quiz Project @ SCTPL

### Quiz

[take quiz](#)

**Previous test result:**

Test given on 29-03-2017. Status :PASSED

Test given on 29-03-2017. Status :FAILED

Test given on 29-03-2017. Status :FAILED

Test given on 29-03-2017. Status :FAILED

The take quiz hyperlink will take you to the **quiz\_main\_page** where the actual quiz will be given by you, now on refresh the questions should not change and the timer should not reset. On clicking submit or if the timer runs out the form should be submitted and should go to [quiz\\_result.php](#) where the results should be displayed. This is the [quiz\\_main\\_page.php](#)

hello akshay 0:58

latest version of html?

html1  
 html2  
 html4  
 html5

PHP recognizes constructors by the name.

classname()  
 \_construct()  
 function\_construct()  
 function\_\_construct()

What does PHP stand for?

Personal Home Page  
 Hypertext Preprocessor  
 Pretext Hypertext Processor  
 Preprocessor Home Page

PHP files have a default file extension of.

.html  
 .xml  
 .php  
 .ph

If Sa = 12 what will be returned when (Sa == 12) ? 5 : 1 is executed?

12  
 1  
 error  
 5

## Online (MCQ) Quiz Application – Level 1 – Internship Project

**Note:** The questions in the quiz should be random. It means that if in my db, 20 questions are present only 5 random questions should be displayed for that our query will be

*select \* from question order by RAND () limit 5 ;*

The main question arises is how we will distinguish between the questions which questions is which because ultimately we need to perform query accordingly to check if the answer provided is right or wrong?

For that we will create dynamic names for our radio buttons so that we can fetch them accordingly

```
echo "<form action='quiz_result.php' method='post'>";
foreach($r as $row)
{
    echo "<div>
        <h1>".$j.". ".$row['question']."</h1>
        <ul style='list-style:none'>
            <li><input type='radio' value='".$row['obj1']."' name='que".$i."'>".$row['obj1']."</li>
            <li><input type='radio' value='".$row['obj2']."' name='que".$i."'>".$row['obj2']."</li>
            <li><input type='radio' value='".$row['obj3']."' name='que".$i."'>".$row['obj3']."</li>
            <li><input type='radio' value='".$row['obj4']."' name='que".$i."'>".$row['obj4']."</li>
        </ul>
        <input type='hidden' name='qid".$i."' value='".$row['qid']."'>
        </div>";
        $i++;$j++;
    }
echo "<input type='submit' value='submit test'></form>";
```

This piece of code will create the questions and its objectives with different name according to the result set which are ultimately encapsulated in a form .The input type hidden is necessary as it will store the uniqueness of the question which will be later used to check the answer of the question.

Now every quiz has a timer in it which should submit the answers if the timer count becomes 0. For that the java-script code we need is

```
countdown=60*1000;
var timerId = setInterval(function(){
    $.post('time_checker.php',{count:countdown});
    countdown -= 1000;
    var min = Math.floor(countdown / (60 * 1000));
    //var sec = Math.floor(countdown - (min * 60 * 1000)); // wrong
    var sec = Math.floor((countdown - (min * 60 * 1000)) / 1000); //correct
    if (countdown <= 0) {
        //alert("30 min!");
        $('form').submit();
        clearInterval(timerId);
        //doSomething();
    } else {
        $("#countTime").html(min + " : " + sec);
    }
}, 1000); //1000ms. = 1sec.
```

## Online (MCQ) Quiz Application – Level 1 – Internship Project

This java-script code will create a timer which will update our div with **#countTime** every second i.e. our timer. If our timer goes **0** we will submit the form. *Hence the quiz will be submitted for checking.*

The **quiz\_result.php** should check if that all the answers are correct using the database and the result should be shown in %. If the % is > 60, PASSED should be displayed else FAILED. The result should be stored in the database respectively and the homepage hyperlink will move you to **user\_home\_page**. This is the **quiz\_result.php**

**All answers**

What does PHP stand for?  
your answer : Hypertext Preprocessor,corect answer : Hypertext Preprocessor

PHP recognizes constructors by the name.  
your answer : function \_\_construct(),corect answer : function \_\_construct()

If Sa = 12 what will be returned when (Sa == 12) ? 5 : 1 is executed?  
your answer : 12,corect answer : 5

PHP files have a default file extension of.  
your answer : .php,corect answer : .php

latest version of html?  
your answer : html4,corect answer : html5

**your score: 3/5, Percent:60%**

Test Status : PASSED

[homepage](#)

For checking if the answers are correct we need to fetch all the radio buttons clicked from the dynamic names we created before:

```
for($i=0;$i<5;$i++)
{
    ${'qid' . $i}=$_REQUEST['qid' . $i];
    if(isset($_REQUEST['que' . $i]))
    {
        ${'que' . $i}=$_REQUEST['que' . $i];
    }
    else
    {
        ${'que' . $i}='';
    }
}
```

We will create dynamic variables with prefix qid and que to store question id which were stored in input type hidden and the option marked respectively for all the questions.

## Online (MCQ) Quiz Application – Level 1 – Internship Project

Now all the data is collected, now we will find out which options marked are correct

```
for($i=0;$i<5;$i++)
{
    $sql1="select * from question where qid='".$${'qid'}.${i}.'" ";
    $res1=mysqli_query($con,$sql1);
    while($row=mysqli_fetch_array($res1))
    {
        echo "<div>
            <h3>".$row['question'].</h3>";
        echo "your answer : ".$${'que'}.${i};
        echo "<br>correct answer : ".$row['answer'];

        echo "</div>";
    }
    $sql="select * from question where qid='".$${'qid'}.${i}.'" and answer='".$${'que'}.${i}.'" ";
    $result=mysqli_query($con,$sql);
    if(mysqli_num_rows($result))
    {
        $count++;
    }
}
```

The first query in the loop is used for displaying the correct answer and the answer that was marked, the next query will check if the marked answer is correct or not, if it is correct we will increment the value of count which in term is the variable which stores no of correct answers. If the count is above 60% we will show passed or else failed and after that we will enter the results in test table in db.

On clicking the back button, the *quiz\_main\_page* should not be loaded instead the *user\_main\_page* should be redirected.

After giving the quiz, the *user\_home\_page* can look like this

### Quiz

[take quiz](#)

Previous test result: [logout](#)

Test given on 12-03-2017. Status :PASSED

## Database schema

### Admin

Aid
Aname
Apass

### User

Uid
Uname
Upass

### Question

Qid
Question
Obj1
Obj2
Obj3
Obj4
Ans

### Test

Tid
Uid
Timestamp
Result

Following files need to be coded: (this is just a descriptive list; a student may code the same logic with more or less files.)

1. Index.php // login page with register hyperlink
2. Register.php
3. Register\_action.php
4. Login\_action.php
5. User\_home\_page.php
6. Quiz\_main\_page.php
7. Quiz\_result.php
8. Admin.php //sign-in page
9. Admin\_action.php
10. Addquestion.php