

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA
FACULTY OF AUTOMATION AND COMPUTER SCIENCE

DATABASE INDIVIDUAL PROJECT

at

DATABASES

Sem. **II**, Year **2021-2022**

Student Name: Tudor Ada Maria

Group Number: 30322

Contents

Chapter 1. Introduction

1. Tools Used.....	1
2. Project theme.....	1
3. Database Creation.....	1
4. Database Diagram.....	7

Chapter 2. WorkFlow

1. Registration.....	8
2. Information Pages	15
3. Quiz.....	22
4. Admin.....	25

Chapter 3. .css Files.....43

BIBLIOGRAPHY.....60

Chapter 1. Introduction

Tools used

For this project we will use XAMPP for the localserver of this project [1] and we will create a folder in ../xampp/htdocs. From here Xampp runs the files on the local server. Xampp uses MariaDB as its database language that has a few differences compared to MySQL, for more information about MariaDB and its syntax visit [2].

For creating and manipulating the database we will use MySQL Workbench. For how to connect a database with MySQL Workbench follow the tutorial at [3].

As a text editor we will use Atom [4].

Project Theme

The project is a web application connected to a database that is meant as a rough demo for an informative site where users can test their knowledge through a quiz. In addition, special users that have an administrator key can access an additional page that serves as a graphical interface for viewing and manipulating the database.

Database Creation

For this project 5 tables are created using the queries from the figures :

- Users : the users tables will store the credentials for logging in and it will be populated through a registration page. Fig.1.1
- Admin_keys : this table will store the keys of the users that are admins and with they values stored in key one may access the graphical interface for the database Fig.1.2. Between admin_keys and users there is a one-to-one relationship as admins must first be users.
- Questions : this table will store the questions of the quiz which will have a unique identification qid that will also be found in quiz_options and quiz_answer were it will serve as a foreign key creating a one-to-many relation between these tables.Fig.1.3
- Quiz_option : will store the options for each question.Fig.1.4
- Quiz_answer : will store the answers for each question.Fig.1.5s

Fig.1.1

```
1 CREATE TABLE `users` (  
2   `id` int(11) NOT NULL AUTO_INCREMENT,  
3   `username` varchar(100) NOT NULL,  
4   `email` varchar(100) NOT NULL,  
5   `password` varchar(100) NOT NULL,  
6   PRIMARY KEY (`id`)  
7 ) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=latin1
```

Fig.1.2

```
1 CREATE TABLE `admin_keys` (  
2   `admin_id` int(11) NOT NULL,  
3   `key` int(11) NOT NULL,  
4   KEY `fk_type_admin` (`admin_id`),  
5   CONSTRAINT `fk_type_admin` FOREIGN KEY (`admin_id`) REFERENCES `users` (`id`) ON DELETE CASCADE ON  
   UPDATE CASCADE  
6 ) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Fig.1.3

```
1 CREATE TABLE `questions` (  
2   `qid` int(11) NOT NULL AUTO_INCREMENT,  
3   `question` varchar(150) NOT NULL,  
4   `is_enabled` int(11) NOT NULL,  
5   PRIMARY KEY (`qid`)  
6 ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1
```

Fig.1.4

```
1 CREATE TABLE `quiz_options` (  
2   `option_id` int(11) NOT NULL,  
3   `qid` int(11) NOT NULL,  
4   `option` varchar(150) NOT NULL,  
5   `is_enabled` int(11) NOT NULL,  
6   KEY `fk_qid` (`qid`),  
7   CONSTRAINT `fk_qid` FOREIGN KEY (`qid`) REFERENCES `questions` (`qid`) ON DELETE CASCADE ON UPDATE  
   CASCADE  
8 ) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

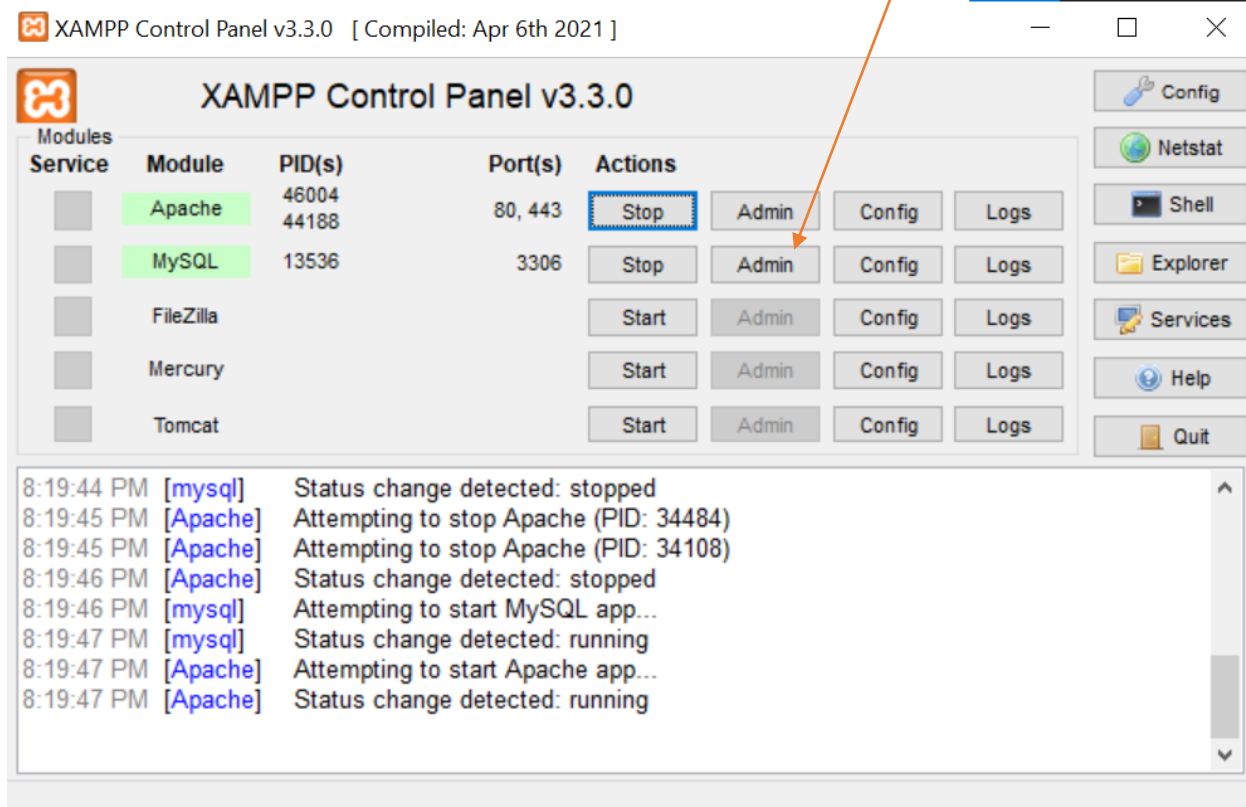
Fig.1.5

```
1 CREATE TABLE `quiz_answer` (  
2   `qa_id` int(11) NOT NULL AUTO_INCREMENT,  
3   `qid` int(11) NOT NULL,  
4   `option_number` int(11) NOT NULL,  
5   PRIMARY KEY (`qa_id`),  
6   KEY `fk_qid_a` (`qid`),  
7   CONSTRAINT `fk_qid_a` FOREIGN KEY (`qid`) REFERENCES `questions` (`qid`) ON DELETE CASCADE ON  
   UPDATE CASCADE  
8 ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1
```

The tables will be populated as the following examples show :

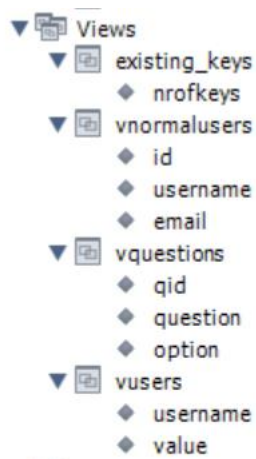
- Users : INSERT INTO `users`(`id`,`username`,`email`,`password`) VALUES ('[value-1]','[value-2]','[value-3]','[value-4]')
- Admin_keys : INSERT INTO `admin_keys`(`admin_id`,`key`) VALUES ('[value-1]','[value-2]')
- Questions : INSERT INTO `questions`(`qid`,`question`,`is_enabled`) VALUES ('[value-1]','[value-2]','[value-3]')
- Quiz_answer : INSERT INTO `quiz_answer`(`qa_id`,`qid`,`option_number`) VALUES ('[value-1]','[value-2]','[value-3]')
- Quiz_question : INSERT INTO `quiz_options` (`option_id`,`qid`,`option`,`is_enabled`) VALUES (1, 1, '6', 1);

Or for those with primary keys one can directly access the xampp admin for data insertion



Note that in the picture above Apache and MySQL server are running. This is mandatory for the localhost/Project files to be accessed.

In order to manipulate the database easier we will create the following views :



```

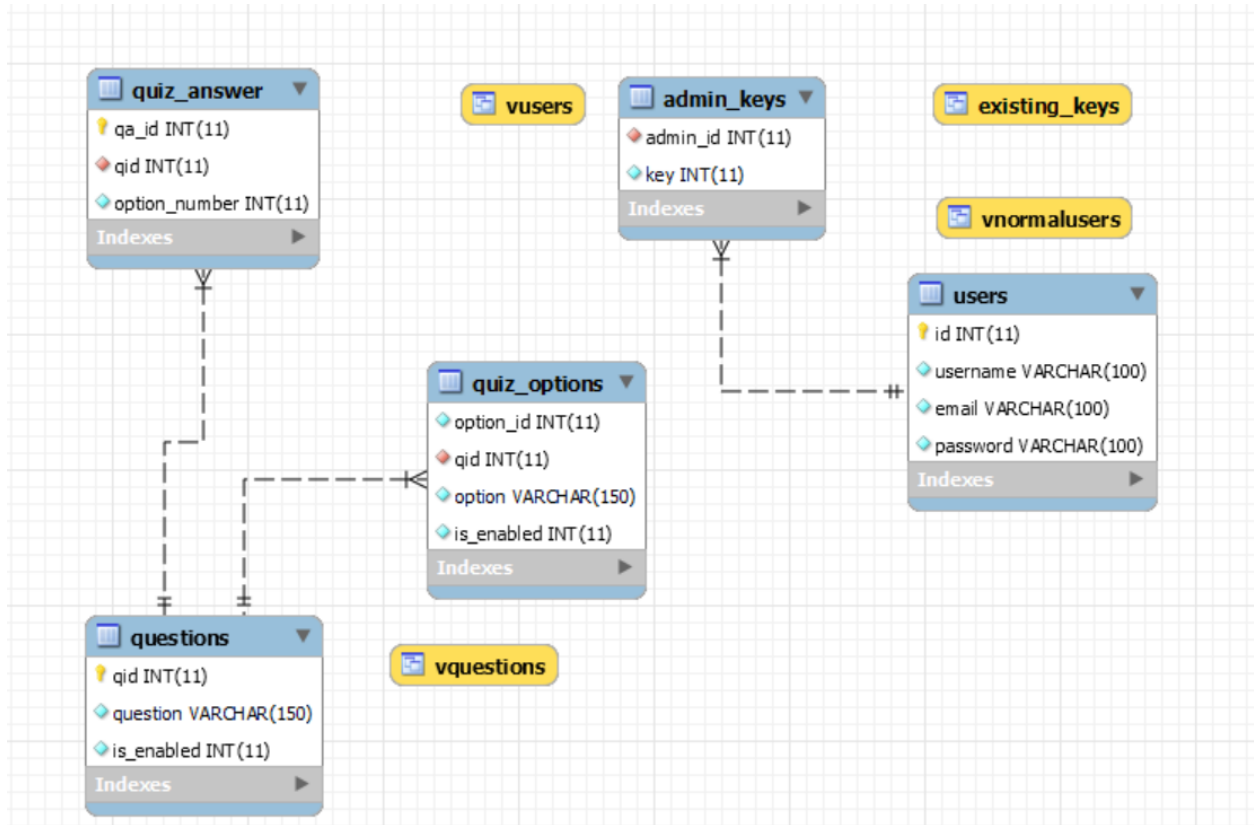
create VIEW `existing_keys` AS select count(keys) AS
`nrofkeys` from `admin_keys`

create view `vNormalUsers` as select id,username,email
from users
where id not in(select admin_id from admin_keys) ;

create view `vquestions` as select qid,question,option
from questions inner join quiz_options on
qid.question=qid.quiz_options

create view `vusers` as select username,email from users
  
```

Database Diagram



Chapter 2. Workflow

Registration

Here we will take step by step each code piece and its equivalent .

First we will create a connection to the database and some verifications through the file server.php. This connection will be used by login.php and register.php .

The registration system was based on [6].

For a better understanding of html and php in general [7] , [8].

```

1
2 <?php
3 // "Sessions or session handling is
4 // a way to make the data available
5 // across various pages of a web application"
6
7 session_start();
8
9 // variabilele pe care le vom folosi
10 $username = "";
11 $email     = "";
12 $errors    = array();
13
14 // le connection
15 $db = mysqli_connect('localhost', 'root', '', 'loginpage');
16
17 // REGISTER USER
18 // isset verifica daca exista si are atribuire o variabila
19 // aici vede daca reg_user din form ii folosit
20
21 if (isset($_POST['reg_user'])) {
22     // receive all input values from the form
23     // mysqli_real_escape_string elimina special char
24
25     $username = mysqli_real_escape_string($db, $_POST['username']);
26     $email     = mysqli_real_escape_string($db, $_POST['email']);
27     $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
28     $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);
29
30     // CORRECT YES OR NO ?
31     // by adding (array_push()) corresponding error unto $errors array
32     // pe scurt : adauga la array de strings cu avertismente
33
34     if (empty($username)) { array_push($errors, "Username is required"); }
35     if (empty($email)) { array_push($errors, "Email is required"); }
36     if (empty($password_1)) { array_push($errors, "Password is required"); }
37     if ($password_1 != $password_2) {
38         array_push($errors, "The two passwords do not match");
39     }

```



```

39 }
40
41 //mysqli_query(DATABASE,QUERY)
42 //sa nu existe deja
43
44 $user_check_query = "SELECT * FROM users WHERE username='$username' OR email='$email' LIMIT 1";
45 $result = mysqli_query($db, $user_check_query);
46 $user = mysqli_fetch_assoc($result);
47
48 if ($user) { // if user exists
49     if ($user['username'] === $username) {
50         array_push($errors, "Username already exists");
51     }
52
53     if ($user['email'] === $email) {
54         array_push($errors, "email already exists");
55     }
56 }
57
58 // Finally, register user if there are no errors in the form
59 if (count($errors) == 0) {
60
61     //encrypt the password before saving in the database
62     //pt niste securitate desi mno
63     //dai un forward dupa logout si tot acolo esti...
64     // Note : check at a later date how to solve
65
66     $password = md5($password_1);
67
68     $query = "INSERT INTO users (username, email, password)
69         VALUES('$username', '$email', '$password')";
70     mysqli_query($db, $query);
71     $_SESSION['username'] = $username;
72     $_SESSION['success'] = "You are now logged in";
73

```

```

72     $_SESSION['success'] = "You are now logged in";
73
74     // check :
75     //https://oxylabs.io/blog/http-headers-explained
76     //this sends u to index.php
77     header('location: index.php');
78 }
79 } //END REGISTER IF
80
81 // LOGIN USER
82 //if login_user is set
83 if (isset($_POST['login_user'])) {
84     $username = mysqli_real_escape_string($db, $_POST['username']);
85     $password = mysqli_real_escape_string($db, $_POST['password']);
86
87     if (empty($username)) {
88         array_push($errors, "Username is required");
89     }
90     if (empty($password)) {
91         array_push($errors, "Password is required");
92     }
93
94     //if everything works dandy
95     if (count($errors) == 0) {
96         $password = md5($password);
97         $query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
98         $results = mysqli_query($db, $query);
99         if (mysqli_num_rows($results) == 1) {
100             $_SESSION['username'] = $username;
101             $_SESSION['success'] = "You are now logged in";
102             header('location: index.php');
103         }else {
104             array_push($errors, "Wrong username/password combination");
105         }
106     }
107 } // END LOGIN IF
108
109 //end php
110 ?>
111

```

Now we will include this in register.php

```
1
2 <?php include('server.php') /*Trebuie inclus pt conexiune */ ?>
3
4 <!DOCTYPE html>
5 <html>
6
7 <head>
8 <title>RegisterPage</title>
9 <link rel="stylesheet" type="text/css" href="styleLR.css">
10 <link rel="stylesheet" type="text/css" href="errors.css">
11 </head>
12
13 <body>
14 <!--Caputul-->
15 <div class="header">
16 <h2>Register</h2>
17 </div>
18 <!--The form -->
19 <form method="post" action="register.php">
20
21 <?php include('errors.php'); ?>
22
23 <!--USERNAME-->
24 <div class="input-group">
25 <label>Username</label>
26 <input type="text" name="username" value="<?php echo $username; ?>">
27 </div>
28 <!--EMAIL-->
29 <div class="input-group">
30 <label>Email</label>
31 <input type="email" name="email" value="<?php echo $email; ?>">
32 </div>
33 <!--password_1-->
34 <div class="input-group">
35 <label>Password</label>
36 <input type="password" name="password_1">
37 </div>
38 <!--password_2-->
39 <div class="input-group">
40 <label>Confirm password</label>
41 <input type="password" name="password_2">
42 </div>
```

```

43
44     <!--Aci este butonul care are rol de check if suntem la Register in server-->
45     <div class="input-group">
46         <button type="submit" class="btn" name="reg_user">Register</button>
47     </div>
48     <!--Daca e deja membru trimite prin link la login page -->
49     <p>
50         Already a member? <a href="login.php">Sign in</a>
51     </p>
52
53 </form>
54
55 </body>
56 </html>
57

```

Register

Username

Email

Password

Confirm password

Register

Already a member? [Sign in](#)

If already registered we have the login.php file.

```
1
2 <?php include('server.php') ?>
3 <!DOCTYPE html>
4
5 <html>
6
7 <head>
8     <title>LoginPage</title>
9     <link rel="stylesheet" type="text/css" href="styleLR.css">
10     <link rel="stylesheet" type="text/css" href="errors.css">
11 </head>
12
13 <body>
14
15     <div class="header">
16         <h2>Login</h2>
17     </div>
18
19     <form method="post" action="login.php">
20
21         <?php include('errors.php'); ?>
22
23         <!--username-->
24         <div class="input-group">
25             <label>Username</label>
26             <input type="text" name="username" >
27         </div>
28
29         <!--password-->
30         <div class="input-group">
31             <label>Password</label>
32             <input type="password" name="password">
33         </div>
34
35         <!--Login_user-->
36         <div class="input-group">
37             <button type="submit" class="btn" name="login_user">Login</button>
38         </div>
39
40         <!--send to register through link -->
41         <p>
42             Not yet a member? <a href="register.php">Sign up</a>
43         </p>
44
45     </form>
```

The image shows a login form with a purple header containing the word "Login". Below the header, there are two input fields: "Username" and "Password". A purple "Login" button is positioned below the password field. At the bottom, there is a link that says "Not yet a member? [Sign up](#)".

Notice that in both files we include an errors.php. That is because in server.php after creating the database connection we have multiple conditions that if not met they will push a message in an error array. When completing the form incorrectly these errors are called from the array.

Here is the errors.php

```
1  <?php if (count($errors) > 0) : ?>
2      <div class="error">
3          <?php foreach ($errors as $error) : ?>
4              <p><?php echo $error ?></p>
5          <?php endforeach ?>
6      </div>
7  <?php endif ?>
```

Information Pages

After logging in the user will be directed to an index.php page as HOME that will provide a short explanation about the site and some useful links. On top of these page there will be a navigation bar towards the other pages : History, Types, QUIZ, The logout option and the admin.

```
<?php
    session_start();

    if (!isset($_SESSION['username'])) {
        $_SESSION['msg'] = "You must log in first";
        header('location: login.php');
    }
    if (isset($_GET['logout'])) {
        session_destroy();
        unset($_SESSION['username']);
        header("location: login.php");
    }
?>

<!DOCTYPE html>

<html>
<head>
    <title>Home</title>
    <link rel="stylesheet" type="text/css" href="content.css">
    <link rel="stylesheet" type="text/css" href="NavigationBar.css">
    <link rel="stylesheet" type="text/css" href="styleLinks.css">

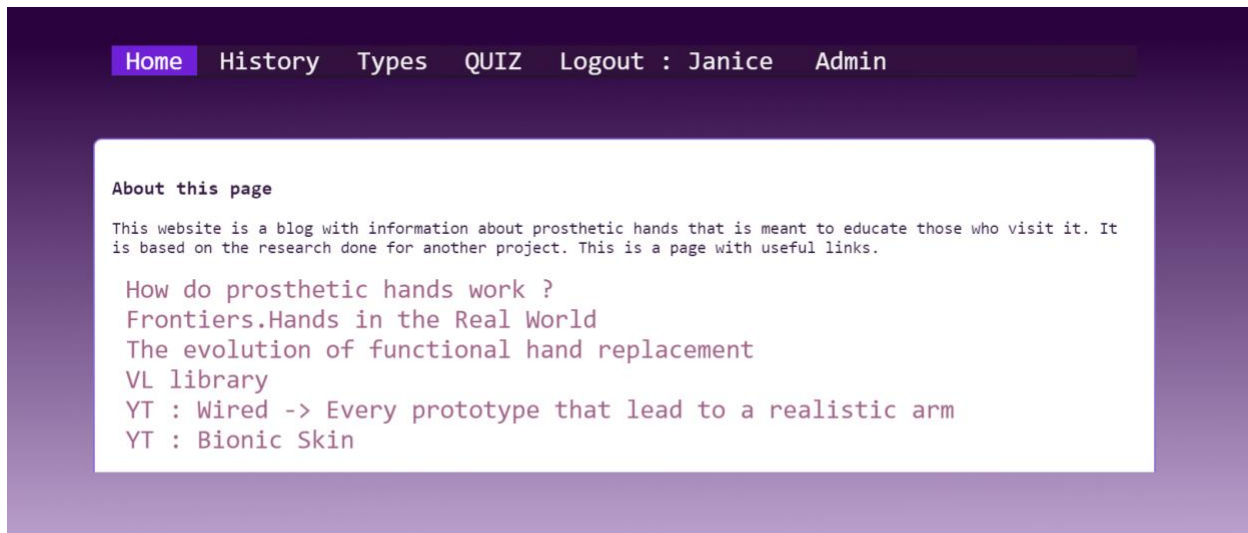
</head>
<body>
```



```

73     YT : Wired -> Every prototype that lead to a realistic arm
74     </a>
75 </div>
76 <div class="linky">
77     <a href="https://www.youtube.com/watch?v=6RzSjhrwYXs">
78     YT : Bionic Skin
79     </a>
80 </div>
81
82 </div>
83
84 </body>
85 </html>
86

```



Below we will have pageHow.html that will provide a gallery by providing inside in iframe afile Gallery.html which will access the folder images inside the project.

```

1
2
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <title>History</title>
7   <link rel="stylesheet" type="text/css" href="content.css">
8   <link rel="stylesheet" type="text/css" href="NavigationBar.css">
9
10 </head>
11 <body>
12
13 <!-- TOP NAVIGATION BAR -->
14
15 <div class="topnav">
16   <a href="index.php">Home</a>
17   <a class="active" href="pageHow.html">History</a>
18   <a href="pageTypes.html">Types</a>
19   <a href="quiz.php">QUIZ</a>
20   <?php if (isset($_SESSION['success'])) : ?>
21     <a href="login.php">
22       Logout
23     </a>
24   <?php endif ?>
25   <a href="pageAdmin.php">Admin</a>
26 </div>
27 <!-- // -->
28
29 <!-- GALLERYYY -->
30 <div class="scrisu">
31
32 <iframe width=100% height="450px" src="Gallery.html"
33   title="History">
34 </iframe>
35
36 </div>
37
38 <!-- Content -->

```

```

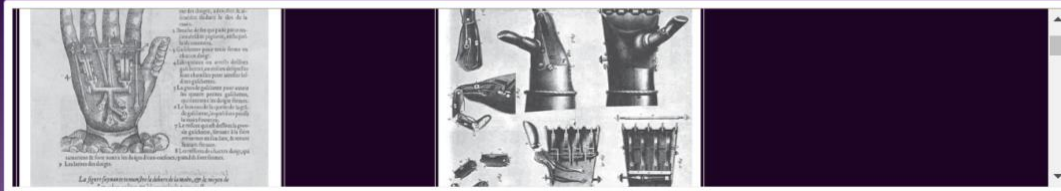
37
38 <!--Content-->
39 <div class="scr1su">
40
41 <h3>Short History of the begining of prosthetic hands</h3>
42 <p>
43 From as early as 77 A.D. when Marcus Sergius, a Roman general,
44 lost his hand in battle and had it replaced with an artificial one that
45 he could use to ride a horse and fight in battle, people have used prosthetic
46 hands to function. Throughout the ages, advancements in artificial hands have
47 improved from body-powered prostheses used after World War I to modern ones found
48 today that use electric power and even intuitive control to make the prostheses work.
49 </p>
50 <p>
51 Among the most famous examples of an early hand prosthesis was the iron hand of
52 German knight Götz von Berlichingen. After Götz lost his hand during the Siege of
53 Landshut (circa 1505) in Bavaria, an artisan fashioned him an iron hand with digits
54 that could be flexed and extended passively at the metacarpophalangeal,
55 proximal interphalangeal and distal interphalangeal joints,
56 as well as the thumb interphalangeal joint.
57 </p>
58 On strapping on the prosthesis, Götz was able to hold reins, grip weapons and return
59 to battle.
60 The device was modelled as an extension of battle armour rather than a human arm and,
61 due to its weight, needed to be attached to Götz's armour with thick leather straps.
62 </p>
63 <p>
64 One of the first descriptions of a noncombative hand prosthesis was in 1600 by Italian
65 surgeon Giovanni Tommaso Minadoi, who described an amputee who could remove his hat,
66 untie a purse and even write with a quill.
67 </p>
68 <p>
69 In the 16th century, French military surgeon Ambroise Paré drew the first detailed
70 design of a spring-loaded prosthetic hand, nicknamed 'Le Petit Lorrain' after the
71 craftsman who fashioned it. Paré also drew a prosthetic arm for an above-elbow
72 amputation.
73 </p>
74 <p>
75 The concept of an 'automatic' body-powered upper limb prosthesis was pioneered
76 by German dentist Peter Baliff in 1818. Using transmission of tension through
77 leather straps, Baliff's device enabled the intact muscles of the trunk and shoulder
78 girdle to elicit motion in a terminal device attached to the amputation stump.
79 For the first time, an amputee was able to operate his prosthesis with fluid body motions,
80 rather than as a distinct foreign object. In the 1860s, the Comte de Beaufort in
81 France adapted the design for use by wounded soldiers. A shoulder harness with a
82 strap buttoned to the trousers was passed through a loop to the contralateral axilla
83 and missing limb, allowing an amputee to manipulate the strap tension to open and close
84 a double spring hook, or flex and extend the thumb on a simple hand with fused fingers
85 </p>
86
87 <p>
88 <h3> <a href="https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4128433/#:~:text=One%20of%20the%20earliest
89 </p>
90
91 </div>
92
93 </body>
94
95 </html>
96

```

Gallery.html

```
1 <!DOCTYPE html>
2
3 <head>
4   <title>Gallery</title>
5   <style>
6     body{
7       background-color: #100424;
8       font-family : consolas ;
9     }
10    img {
11      border: 1px solid #ddd;
12      border-radius: 4px;
13      padding: 5px;
14      max-width: 100%;
15      height: auto;
16    }
17  </style>
18
19
20 </head>
21
22 <body>
23
24   <img src = "images/Ambroise Pare.jpg" alt = "design made by Ambroise Pare">
25   <img src = "images/Götz von Berlichingen.jpg" alt = "The mechanical hand of the knight Gotz">
26   <img src = "images/Gotz_2.jpg" alt = "Gotz hand">
27
28
29   <img src = "images/work tools.jpg" alt = "ww1 soldiers with prosthetics">
30   <img src = "images/hook.jfif" alt = "The still very used format of a claw hand ">
31   <img src = "images/1916.jpg" alt = "A hand design from 1916">
32
33
34   <img src = "images/hybrid.jfif" alt = "model of a hybrid prosthetic">
35   <img src = "images/myoelectric.jfif" alt = "myoelectric hand">
36   <img src = "images/hannnd.jfif" alt = "A tendon based design">
37
38
39   <img src = "images/body-powered.jpg" alt = "Body-powered prosthetic">
40   <img src = "images/bionic.jfif" alt = "bionic hand ">
41   <img src = "images/R.jfif" alt = "bionic hand ">
42
43
44 </body>
45
46
47 </html>
```

[Home](#) [History](#) [Types](#) [QUIZ](#) [Logout](#) [Admin](#)



Short History of the beginning of prosthetic hands

From as early as 77 A.D. when Marcus Sergius, a Roman general, lost his hand in battle and had it replaced with an artificial one that he could use to ride a horse and fight in battle, people have used prosthetic hands to function. Throughout the ages, advancements in artificial hands have improved from body-powered prostheses used after world war I to modern ones found today that use electric power and even intuitive control to make the prostheses work.

Among the most famous examples of an early hand prosthesis was the iron hand of German knight Götz von Berlichingen. After Götz lost his hand during the Siege of Landshut (circa 1386) in Bavaria, an artisan fashioned him an iron hand with digits that could be flexed and extended passively at the metacarpophalangeal, proximal interphalangeal and distal interphalangeal joints, as well as the thumb interphalangeal joint.

On strapping on the prosthesis, Götz was able to hold reins, grip weapons and return to battle. The device was modelled as an extension of battle armour rather than a human arm and, due to its weight, needed to be attached to Götz's armour with thick leather straps.

One of the first descriptions of a noncompetitive hand prosthesis was in 1686 by Italian surgeon Giovanni Tommaso Minardi, who described an amputee who could remove his hat, untie a purse and even write with a quill.

In the 18th century, French military surgeon Ambroise Paré drew the first detailed design of a spring-loaded prosthetic hand, nicknamed "Le Petit Lorrain" after the craftsman who fashioned it. Paré also drew a prosthetic arm for an above-elbow amputation.

The concept of an "automatic" body-powered upper limb prosthesis was pioneered by German dentist Peter Saliff in 1838. Using transmission of tension through leather straps, Saliff's device enabled the intact muscles of the trunk and shoulder girdle to assist motion in a terminal device attached to the amputated stump. For the first time, an amputee was able to operate his prosthesis with fluid body motions, rather than as a distinct foreign object. In the 1860s, the Comte de Beaufort in France adopted the design for use by wounded soldiers. A shoulder harness with a strap buttoned to the trousers was passed through a loop to the contralateral axilla and missing limb, allowing an amputee to manipulate the strap tension to open and close a double spring hook, or flex and extend the thumb on a simple hand with fused fingers.

[Information Source](#)

The pageTypes.html is simply an information page

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>pageHow</title>
5   <link rel="stylesheet" type="text/css" href="content.css">
6   <link rel="stylesheet" type="text/css" href="NavigationBar.css">
7 </head>
8
9 <body>
10
11 <!-- TOP NAVIGATION BAR -->
12 <div class="topnav">
13   <a href="index.php">Home</a>
14   <a href="pageHow.html">History</a>
15   <a class="active" href="pageTypes.html">Types</a>
16   <a href="quiz.php">QUIZ</a>
17   <?php if (isset($_SESSION['success'])) : ?>
18     <a href="login.php">
19       Logout
20     </a>
21   <?php endif ?>
22   <a href="pageAdmin.php">Admin</a>
23 </div>
24 <!-- Content -->
25
26 <div class = "scrisu">
27
28   <h1><strong>Passive</strong></h1>
29   <h3>
30     This mostly nonfunctional type of prosthetic hand is used mainly for cosmetic purposes.
31     Some of these passive hands can have fingers that the user can bend or reposition if desired.
32     The passive artificial hand is made to match the other hand as much as possible and
33     is then fitted with a vinyl or silicone glove that can match the person's skin tone.
34     In addition, the gloves are customizable to match what their real hand would look like with matching veins and fingernails.
35   </h3>
36   <h1><strong>Body-Powered</strong></h1>
37   <h3>
38     A simple form of movable prosthetic hands uses body power to control the hand.
39     A series of cables attached to the artificial hand and to a harness that the user
40     wears let the person open or close the hand by moving their shoulder or upper arm.
41     This system uses a voluntary opening of the hand triggered by the user, and then the
42     hand voluntarily closes once pressure is released through the cable system.
43   </h3>
44   <h1><strong>Myoelectric</strong></h1>
45   <h3>
46     This functional type of prosthetic hand is called a myoelectric prosthesis.
47     The muscles of the affected arm work together with electrodes fitted inside of the prosthetic hand.
48     When the person contracts the muscles of the limb, electrical outputs send a message to the
49     artificial hand to open or close it. Conveniently, electric-powered hands also let the user change
50     the speed at which they can open and close the hand. With the addition of a wrist rotator,
51     the person can also twist and turn their wrist.
52   </h3>
53   <h1><strong>Hybrid</strong></h1>
54   <h3>
55     For those who have lost a substantial amount of their limb, a hybrid prosthesis is a viable option.
56     Using a combination of body power and electrical power control, a hybrid prosthesis is best used for
57     a transhumeral prosthetic that can move the elbow, hands, and fingers simultaneously.
58   </h3>
59
60 </div>
61
```

By now it is noticeable that the top navigation bar part of the code is pretty similar however it changes slightly depending on the active page.

Quiz

[5]. The QUIZ page is constructed using three .php files : quiz.php, score.php and quizzclass.php

Quiz.php

```
1
2 <html>
3 <head>
4   <title>Quiz</title>
5   <link rel="stylesheet" type="text/css" href="NavigationBar.css">
6   <link rel="stylesheet" type="text/css" href="content.css">
7   <link rel="stylesheet" type="text/css" href="input.css">
8 </head>
9
10 <body>
11
12 <!-- TOP NAVIGATION BAR-->
13
14 <div class="topnav">
15   <a href="index.php">Home</a>
16   <a href="pageHow.html">History</a>
17   <a href="pageTypes.html">Types</a>
18   <a class="active" href="quiz.php">QUIZ</a>
19   <?php if (isset($_SESSION['success'])) : ?>
20     <a href="login.php">
21       Logout
22     </a>
23   <?php endif ?>
24
25 </div>
26
27
28 <?php
29   include 'quizzclass.php';
30   $db = new Quiz();
31   $questions = $db->get_questions();
32 ?>
33
34 <div class="scripu">
35   <h1>Multiple Choice Questions Answers</h1>
36   <p>Please fill the details and answers the all questions-</p>
37
38   <form action="score.php" method="post">
39
40     <?php
41     foreach($questions as $ques) {
42       $options = $db->quiz_options($ques[0]);
43     ?>
44
45     <h4><?php echo $ques[1]; ?></h4>
46     <div class="input-group-text">
47       <ol>
48         <?php
49         foreach($options as $option) {
50           echo "<li><input type='radio' name='". $option[1]."' value='". $option[0]."' required/> ". $option[2]. "</li>";
51         }
52       </ol>
53     </div>
54   <?php } ?>
55 </div>
56
57 <div class="form-group">
58   <input type="submit" value="Submit" name="submit" class="btn"/>
59 </div>
60
61 </form>
62
63 </div>
64
65 </body>
66
67 </html>
68
69
```

Quizclass.php

```
1  <?php
2
3  class Quiz {
4      // Database credentials
5      private $host      = 'localhost';
6      private $username = 'root';
7      private $password = '';
8      private $database = 'loginpage';
9      public $db;
10
11     public function __construct(){
12         if(!isset($this->db)){
13             // Connect to the database
14             try {
15                 $this->db = new mysqli($this->host, $this->username, $this->password, $this->database);
16             }catch (Exception $e){
17                 $error = $e->getMessage();
18                 echo $error;
19             }
20         }
21     }
22
23     public function get_questions(){
24         $select = "SELECT * FROM `questions` where is_enabled = '1' ";
25         $result = mysqli_query($this->db , $select);
26         return mysqli_fetch_all($result);
27     }
28
29     public function quiz_options($qid) {
30         $select = "SELECT * FROM `quiz_options` where qid = '$qid' AND is_enabled = '1' ";
31         $result = mysqli_query($this->db , $select);
32         return mysqli_fetch_all($result);
33     }
34
35     public function answer($qid) {
36         $select = "SELECT * FROM `quiz_answer` where qid = '$qid' ";
37         $result = mysqli_query($this->db , $select);
38         return mysqli_fetch_all($result);
39     }
40 }
41
42 ?>
```

This file defines a class quiz and provides the methods to extract the data from the database. It also creates a connection to said database.

Score.php

This part of the code does not function properly. It is meant to check the answers provided with those in the database and offer a score to the users, however here the data collection from the post is not yet implemented correctly.

```
1  <?php
2
3  //not working properly
4  include 'quizclass.php';
5  $db = new Quiz();
6  $score = 0;
7  $questions = $db->get_questions();
8
9  foreach($questions as $ques) {
10     $options = $db->quiz_options($ques[0]);
11     foreach($options as $option) {
12         if(isset($_POST[$option[1]] )) echo $_POST[$option[1]];
13     }
14 }
15
16 /*foreach($_POST as $data)
17 {
18     echo $data;
19
20     // $answer = $db->answer($k);
21     //if($answer[0][2] == $v) { // option is correct
22     // $score++;
23     //}
24 }
25 echo $score
26
27 */
28 ?>
29
```


[Home](#) [History](#) [Types](#) [QUIZ](#)

Multiple Choice Questions Answers

Please fill the details and answers the all questions-

How many types of prosthetic hands are there ?

- ☐ 6
- ☐ 7
- ☐ 8
- ☐ 9

Who is Gotz?

- ☐ a german knight
- ☐ a blacksmith
- ☐ an english knighth
- ☐ a roman genaral

When did the concept of automatic limbs appear ?

- ☐ 77 AD
- ☐ 1585
- ☐ 1818
- ☐ 1688

What is bionic skin?

- ☐ skin made for droids
- ☐ skin for prosthetics
- ☐ impossible
- ☐ nothing yet

Submit

Admin

The Admin page is by far the most complex as it incorporates multiple others as iframes.

First the user will be met with a form that will ask for a key. Administrators have these keys stored in the admin_keys table from the database. This page is the pageAdmin.php and it uses the checkKey.php file to verify the key.

If the key that is introduced is correct then the admin will be taken to the interface for the admins, adminInterface.php.

pageAdmin.php

```
1  <?php include('checkKey.php') ?>
2  <!DOCTYPE html>
3
4  <html>
5
6  <head>
7      <title>AdminPage</title>
8      <link rel="stylesheet" type="text/css" href="styleLR.css">
9      <link rel="stylesheet" type="text/css" href="NavigationBar.css">
10
11 </head>
12
13 <body>
14
15
16 <div class="topnav">
17     <a href="index.php">Home</a>
18     <a href="pageHow.html">History</a>
19     <a href="pageTypes.html">Types</a>
20     <a href="quiz.php">QUIZ</a>
21
22     <?php if (isset($_SESSION['success'])) : ?>
23         <a href="login.php">
24             Logout : <?php echo $_SESSION['username']; ?>
25         </a>
26     <?php endif ?>
27     <a class="active" href="pageAdmin.php">Admin</a>
28 </div>
29
30
```

```

1  <?php include('checkKey.php') ?>
2  <!DOCTYPE html>
3
4  <html>
5
6  <head>
7      <title>AdminPage</title>
8      <link rel="stylesheet" type="text/css" href="styleLR.css">
9      <link rel="stylesheet" type="text/css" href="NavigationBar.css">
10
11 </head>
12
13 <body>
14
15
16 <div class="topnav">
17     <a href="index.php">Home</a>
18     <a href="pageHow.html">History</a>
19     <a href="pageTypes.html">Types</a>
20     <a href="quiz.php">QUIZ</a>
21
22     <?php if (isset($_SESSION['success'])) : ?>
23         <a href="login.php">
24             Logout : <?php echo $_SESSION['username']; ?>
25         </a>
26     <?php endif ?>
27     <a class="active" href="pageAdmin.php">Admin</a>
28 </div>
29
30

```

checkKey.php

```

2
3  <?php
4
5  session_start();
6
7  $key="";
8  $db = mysqli_connect('localhost', 'root', '', 'loginpage');
9
10 if (isset($_POST['admin_check'])){
11
12     $key=mysqli_real_escape_string($db, $_POST['key']);
13
14     $check_exists="SELECT * FROM `admin_keys` WHERE admin_keys.key='$key'";
15     $result = mysqli_query($db, $check_exists);
16     $admin = mysqli_fetch_assoc($result);
17
18     if ($admin) { // if admin exists
19         header('location:adminInterface.php');
20     }
21
22 }
23
24 >>
25

```

adminInterface.php provides the admin with multiple choices of data viewing or manipulation by using iframes.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Admins</title>
5   <link rel="stylesheet" type="text/css" href="content.css">
6   <link rel="stylesheet" type="text/css" href="NavigationBar.css">
7
8
9 </head>
10 <body>
11
12 <!--TOP NAVIGATION BAR-->
13
14 <div class="topnav">
15   <a href="index.php">Home</a>
16   <a href="pageHow.html">History</a>
17   <a href="pageTypes.html">Types</a>
18   <a href="quiz.php">QUIZ</a>
19   <?php if (isset($_SESSION['success'])) : ?>
20     <a href="login.php">
21       Logout
22     </a>
23   <?php endif ?>
24
25   <a class="active" href="adminInterface.php">Admin</a>
26 </div>
27
28 <div class = "scrisu">
29   <iframe width=100% height="450px" src="ViewEverything.php" title="View"></iframe>
30 </div>
31
32 <div class = "scrisu">
33   <iframe width=100% height="450px" src="create-form.php" title="Create Records"></iframe>
34 </div>
35 <div class = "scrisu">
36   <iframe width=100% height="450px" src="user-tabel.php" title="Update/Delete Records"></iframe>
37 </div>
38
39 <div class = "scrisu">
40   <iframe width=100% height="450px" src="question-tabel.php" title="View Questions Records"></iframe>
41 </div>
42
43
44 </body>
45
46 </html>
47

```

For the first iframe we have ViewEverything.php. Here the admin sees all the users with their information and the total number of admin keys that represent the first users in the tabel.

```
<?php

include('View_All.php');
?>

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Operations</title>
<link rel="stylesheet" type="text/css" href="tabelStyle.css">
<link rel="stylesheet" type="text/css" href="styleLinks.css">
</head>

<table border="1">
  <caption> All Users , Admin Keys :

    <?php
      foreach($fetchCount as $da){
        echo $da['nrofkeys'];
      }
    ?>

  </caption>
  <thead>
  <tr>
    <th>S.N</th>
    <th>username</th>
    <th>mail</th>

    </tr>
  </thead>
  <tbody>
```

```

<?php

include('View_All.php');
?>

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Operations</title>
<link rel="stylesheet" type="text/css" href="tabelStyle.css">
<link rel="stylesheet" type="text/css" href="styleLinks.css">
</head>

<table border="1">
  <caption> All Users , Admin Keys :

    <?php
      foreach($fetchCount as $da){
        echo $da['nrofkeys'];
      }
    ?>

  </caption>
  <thead>
  <tr>
    <th>S.N</th>
    <th>username</th>
    <th>mail</th>

  </tr>
</thead>
<tbody>

```

viewAll.php

```
1  <?php
2
3  $db = mysqli_connect('localhost', 'root', '', 'loginpage');
4
5  $fetchData= fetch_data($db);
6  $fetchCount= fetch_Count($db);
7
8  // fetch query
9  function fetch_data($db){
10     $query="SELECT * from vusers ORDER BY username DESC";
11     $exec=mysqli_query($db, $query);
12     if(mysqli_num_rows($exec)>0){
13
14         $row= mysqli_fetch_all($exec, MYSQLI_ASSOC);
15         return $row;
16
17     }else{
18         return $row=[];
19     }
20 }
21
22 function fetch_Count($db){
23     $query="SELECT * from existing_keys ";
24     $exec=mysqli_query($db, $query);
25     if(mysqli_num_rows($exec)>0){
26
27         $row= mysqli_fetch_all($exec, MYSQLI_ASSOC);
28         return $row;
29
30     }else{
31         return $row=[];
32     }
33 }
34
35 ?>
36
```


All Users , Admin Keys : 2		
S.N	USERNAME	MAIL
1	phhattjokke	pizza@gmail.com
2	Maria	tudor.ada2001@gmail.com
3	Maria	uiedu@sguqd.com
4	Janice	carvona2001@gmail.com

[9] The second application for the admin is to add users with a creation form using create-form.php and create-script.php

Create-form.php

```

1  <?php
2
3
4  include('create-script.php');
5  >>
6
7  <!DOCTYPE html>
8  <html>
9  <head>
10 <meta name="viewport" content="width=device-width, initial-scale=1">
11 <title>Create Form</title>
12 <link rel="stylesheet" type="text/css" href="styleLR.css">
13
14 </head>
15 <body>
16 <!--====form section start====-->
17
18 <div class="input-group">
19
20 <div class="header">
21 <h2>Create Form</h2>
22
23
24 </div>
25
26 <p style="color:red"><?php if(!empty($msg)){echo $msg; }></p>
27
28 <form method="post" action="">
29 <label>UserName</label>
30 <input type="text" placeholder="Enter UserName" name="username" required>
31 <label>Email Address</label>
32 <input type="email" placeholder="Enter Email Address" name="email_address" required>
33 <label>Password</label>
34 <input type="pass" placeholder="Enter Password" name="pass" required>
35
36
37 <button type="submit" class="btn" name="create">Submit</button>
38
39 </form>
40 </div>
41 </div>
42 <!--====form section start====-->
43
44
45 </body>
46 </html>
47

```

Create-script.php

```
1  ?php
2
3  $db = mysqli_connect('localhost', 'root', '', 'loginpage');
4
5
6  if(isset($_POST['create'])){
7
8      $msg=insert_data($db);
9
10 }
11
12 // insert query
13 function insert_data($db){
14
15     $username= legal_input($_POST['username']);
16     $email_address= legal_input($_POST['email_address']);
17     $pass= md5(legal_input($_POST['pass']));
18
19
20     $query="INSERT INTO users (username,email,password) VALUES ('$username','$email_address','$pass')";
21     $exec= mysqli_query($db,$query);
22     if($exec){
23
24         $msg="Data was created sucessfully";
25         return $msg;
26
27     }else{
28         $msg= "Error: " . $query . "<br>" . mysqli_error($db);
29     }
30 }
31
32 // convert illegal input to legal input
33 function legal_input($value) {
34     $value = trim($value);
35     $value = stripslashes($value);
36     $value = htmlspecialchars($value);
37     return $value;
38 }
39 ?>
40
```



The screenshot displays a web application interface for creating a new user. The page features a purple header bar with the text "Create Form". Below this, the form is contained within a white box. It includes three text input fields: "UserName" with the placeholder "Enter Username", "Email Address" with the placeholder "Enter Email Address", and "Password" with the placeholder "Enter Password". A purple "Submit" button is positioned at the bottom left of the form area. The entire interface is set against a dark purple background.

[9] the third application is an Update/delete one of the users however it has no access to the admins. We will use the user-tabel.php, update-form.php, read-script.php, update-script.php and delete-script.php

User-tabel.php

```
1  <?php
2
3  include('read-script.php');
4  ?>
5
6
7  <!DOCTYPE html>
8  <html>
9  <head>
10 <meta name="viewport" content="width=device-width, initial-scale=1">
11 <title>Operations</title>
12 <link rel="stylesheet" type="text/css" href="tabelStyle.css">
13 <link rel="stylesheet" type="text/css" href="styleLinks.css">
14 </head>
15 <body>
16
17
18 <div class="table-data">
19
20 <table border="1">
21   <caption> Users  <a href="user-tabel.php">Refresh</a></caption>
22   <thead>
23     <tr>
24       <th>S.N</th>
25       <th>Username</th>
26       <th>Email Address</th>
27       <th>Edit</th>
28       <th>Delete</th>
29     </tr>
30   </thead>
31   <tbody>
32
```

```

1  <?php
2
3  include('read-script.php');
4  ?>
5
6
7  <!DOCTYPE html>
8  <html>
9  <head>
10 <meta name="viewport" content="width=device-width, initial-scale=1">
11 <title>Operations</title>
12 <link rel="stylesheet" type="text/css" href="tabelStyle.css">
13 <link rel="stylesheet" type="text/css" href="styleLinks.css">
14 </head>
15 <body>
16
17
18 <div class="table-data">
19
20 <table border="1">
21   <caption> Users  <a href="user-tabel.php">Refresh</a></caption>
22   <thead>
23     <tr>
24       <th>S.N</th>
25       <th>Username</th>
26       <th>Email Address</th>
27       <th>Edit</th>
28       <th>Delete</th>
29     </tr>
30   </thead>
31   <tbody>
32

```

Read-script.php

```
1  <?php
2
3  $db = mysqli_connect('localhost', 'root', '', 'loginpage');
4
5  $fetchData= fetch_data($db);
6
7  // fetch query
8  function fetch_data($db){
9      $query="SELECT * from vnormalusers ORDER BY id DESC";
10     $exec=mysqli_query($db, $query);
11     if(mysqli_num_rows($exec)>0){
12
13         $row= mysqli_fetch_all($exec, MYSQLI_ASSOC);
14         return $row;
15
16     }else{
17         return $row=[];
18     }
19 }
20 ?>
21
```

Update-form.php

```

1  <?php
2
3  include('update-script.php');
4  ?>
5
6
7  <!DOCTYPE html>
8  <html>
9  <head>
10 <meta name="viewport" content="width=device-width, initial-scale=1">
11 <title>PHP CRUD Operations</title>
12
13 </head>
14 <body>
15 <!--=====form section start=====-->
16
17 <div class="user-detail">
18
19 <div class="form-title">
20 <h2>Create Form</h2>
21 </div>
22
23 <p style="color:red">
24
25 <?php if(!empty($msg)){echo $msg; }?>
26 </p>
27 <form method="post" action="">
28 <label>UserName</label>
29 <input type="text" placeholder="Enter Username" name="username" required value="<?php echo isset($editData) ? $editData['username'] : '' ?>">
30
31 <label>Email Address</label>
32 <input type="email" placeholder="Enter Email Address" name="email_address" required value="<?php echo isset($editData) ? $editData['email'] : '' ?>">
33
34 <button type="submit" name="update">Submit</button>
35 </form>
36 </div>
37 </div>
38 <!--=====form section start=====-->
39 </body>
40 </html>
41

```

Update-script.php

```

1  <?php
2
3  $db = mysqli_connect('localhost', 'root', '', 'loginpage');
4
5
6  if(isset($_GET['edit'])){
7
8      $id= $_GET['edit'];
9      $editData= edit_data($db, $id);
10 }
11
12 if(isset($_POST['update']) && isset($_GET['edit'])){
13
14     $id= $_GET['edit'];
15     update_data($db,$id);
16
17 }
18
19 function edit_data($db, $id)
20 {
21     $query= "SELECT * FROM users WHERE id= $id";
22     $exec = mysqli_query($db, $query);
23     $row= mysqli_fetch_assoc($exec);
24     return $row;
25 }
26

```

```

26
27 // update data query
28 function update_data($db, $id){
29
30     $username= legal_input($_POST['username']);
31     $email_address= legal_input($_POST['email_address']);
32
33
34     $query="UPDATE users
35             SET username='$username',
36                 email='$email_address'
37             WHERE id=$id";
38
39     $exec= mysqli_query($db,$query);
40
41     if($exec){
42         header('location:user-tabel.php');
43
44     }else{
45         $msg= "Error: " . $query . "<br>" . mysqli_error($db);
46         echo $msg;
47     }
48 }
49
50 // convert illegal input to legal input
51 function legal_input($value) {
52     $value = trim($value);
53     $value = stripslashes($value);
54     $value = htmlspecialchars($value);
55     return $value;
56 }
57 ?>
58

```

Delete-script.php

```

1  php
2
3  $db = mysqli_connect('localhost', 'root', '', 'loginpage');
4  if(isset($_GET['delete'])){
5
6      $id= $_GET['delete'];
7      delete_data($db, $id);
8
9  }
10
11  // delete data query
12  function delete_data($db, $id){
13
14      $query="DELETE from users WHERE id=$id";
15      $exec= mysqli_query($db,$query);
16
17      if($exec){
18          header('location:user-tabel.php');
19      }else{
20          $msg= "Error: " . $query . "&lt;br&gt;" . mysqli_error($db);
21          echo $msg;
22      }
23  }
24  ?&gt;
25
</pre

```

Users [Refresh](#)

S.N	USERNAME	EMAIL ADDRESS	EDIT	DELETE
1	Maria	wiedu@sgwqd.com	Edit	Delete
2	phhattjokke	pizza@gmail.com	Edit	Delete

Finally we have an application to view all the questions and their options by using question-tabel.php and read-question-script.php. these work on a similar principle as those above

Question-tabel.php

```
1  <?php
2
3  include('read-questions-script.php');
4  ?>
5
6  <!--The head and style sheets used-->
7  <!DOCTYPE html>
8  <html>
9  <head>
10 <meta name="viewport" content="width=device-width, initial-scale=1">
11 <title>Operations</title>
12 <link rel="stylesheet" type="text/css" href="tabelStyle.css">
13 <link rel="stylesheet" type="text/css" href="styleLinks.css">
14 </head>
15 <body>
16
17 <!--Tabel-->
18
19 <table border="1">
20   <caption> Questions  <a href="question-tabel.php">Refresh</a></caption>
21   <thead>
22     <tr>
23       <th>S.N</th>
24       <th>Question</th>
25       <th>Option</th>
26
27     </tr>
28   </thead>
29   <tbody>
30
```

```

29 </tbody>
30
31 <?php
32     if(count($fetchData)>0){
33         $sn=1;
34         foreach($fetchData as $data){
35
36             ?>
37             <tr>
38                 <td><?php echo $sn; ?></td>
39                 <td><?php echo $data['question']; ?></td>
40                 <td><?php echo $data['option']; ?></td>
41             </tr>
42
43             <?php
44                 $sn++; }
45             }else{
46                 ?>
47                 <tr>
48                     <td colspan="7">No Data Found</td>
49                 </tr>
50             <?php
51                 }
52                 ?>
53             </tbody>
54         </table>
55
56
57 </body>
58 </html>
59

```

Read-question-script.php

```

1  <?php
2
3  $db = mysqli_connect('localhost', 'root', '', 'loginpage');
4
5  $fetchData= fetch_data($db);
6
7  // fetch query
8  function fetch_data($db){
9      $query="SELECT * from vquestions ORDER BY qid DESC";
10     $exec=mysqli_query($db, $query);
11     if(mysqli_num_rows($exec)>0){
12
13         $row= mysqli_fetch_all($exec, MYSQLI_ASSOC);
14         return $row;
15
16     }else{
17         return $row=[];
18     }
19 }
20 ?>
21

```

CSS FILES

The .css files provide the stylesheets for this project either by applying them to existing types of html entities or by creating new variable classes.

- tabelStyle.css
- styleLink.css
- styleLR.css
- reset.css (when needing to clear a style)
- NavigationBar.css
- Input.css
- Errors.css
- Content.css

tabelStyle.php

```
*{  
  
    font-family: consolas;  
  
}  
  
/*applies to the body tag*/  
  
body {  
  
    font-size: 120%;  
  
    background: #f1dcfe;  
  
    /*make a picture BG*/  
  
    background-image: url('BG.svg');  
  
    background-repeat: no-repeat;  
  
    background-attachment: fixed;  
  
    background-size: cover;  
  
}  
  
th, td {  
  
    padding: 8px;  
  
}
```

```
table {  
  
    border-collapse: collapse;  
  
    text-align: center;  
  
        vertical-align: middle;  
  
    border: 2px solid #8969d0;  
  
}
```

```
thead {  
  
    background-color: white;  
  
    color: #8969d0;  
  
    font-size: 0.875rem;  
  
        text-transform: uppercase;  
  
        letter-spacing: 2%;  
  
}
```

```
caption {  
  
    font-weight: bold;  
  
    font-size: 24px;  
  
    text-align: left;  
  
    color: #8969d0;  
  
    margin-bottom: 16px;
```

```
}
```

```
tbody tr:nth-child(odd) {  
    background-color: #ecb3ff;  
}
```

```
tbody tr:nth-child(even) {  
    background-color: #e6ccff;  
}
```

```
.titleT {  
  
    color: white;  
    background: #8a50d9;  
    text-align: left;  
    border: 2px solid #8969d0;  
    border-bottom: none;  
    border-radius: 10px 10px 0px 0px;  
    padding: 20px;  
}
```

styleLink.css

```
.linky {
```

```
margin: auto;
```

```
}
```

```
.linky a {
```

```
color: #FF0099;
```

```
text-align: center;
```

```
padding: 14px 16px 14px 16px;
```

```
text-decoration: none;
```

```
font-size: 30px;
```

```
}
```

```
.linky a:visited{
```

```
color: #A26389;
```

```
}
```

```
.linky a:hover {
```

```
color: #FF00F9;
```

```
}
```

```
.linky a:active {
```

```
    color: #AD00FF;
```

```
}
```

styleLR.css

```
/*applies to everything by default*/
```

```
*{
```

```
    font-family: consolas;
```

```
}
```

```
/*applies to the body tag*/
```

```
body {
```

```
    font-size: 120%;
```

```
    background: #f1dcfe;
```

```
/*make a picture BG*/
```

```
background-image: url('BG.svg');
```

```
background-repeat: no-repeat;
```

```
background-attachment: fixed;
```

```
background-size: cover;
```

```
}
```



```
/*BUTTONS*/
```

```
/*class = "btn" */
```

```
.btn {
```

```
padding: 10px;
```

```
font-size: 15px;
```

```
color: white;
```

```
background: #9368eb;
```

```
border: none;
```

```
border-radius: 5px;
```

```
}
```

```
/*Only for login and register page*/
```

```
/*INPUT*/
```

```
/* div -> class = "input-group" : These are the text boxes for input*/
```

```
.input-group {
```

```
margin: 10px 0px 10px 0px;
```

```
}
```

```
.input-group label {  
  
    display: block;  
  
    text-align: left;  
  
    margin: 3px;  
  
}  
  
.input-group input {  
  
    height: 30px;  
  
    width: 93%;  
  
    padding: 5px 10px;  
  
    font-size: 16px;  
  
    border-radius: 5px;  
  
    border: 2px solid #7527a5;  
  
}
```

```
/*HEADER*/
```

```
/*This will be the style for the div -> class = "header"*/
```

```
.header {  
  
    width: 60%;  
  
    margin: 70px auto 0px;  
  
    color: white;  
  
    background: #8a50d9;
```

```

text-align: left;

border: 2px solid #8969d0;

border-bottom: none;

border-radius: 10px 10px 0px 0px;

padding: 20px;

}

/*CONTENT*/

/*This will be the style for the div,form */

form, .content {

width: 60%;

margin: 0px auto;

padding: 20px;

border: 1px solid #c79cea;

background: white;

border-radius: 0px 0px 10px 10px;

}

```

Reset.css

```

/* CSS reset */

body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,form,fieldset,input,textarea,p,blockquote,th,td {

margin:0;

padding:0;

```

```

}

html,body {

    margin:0;

    padding:0;

}

table {

    border-collapse:collapse;

    border-spacing:0;

}

fieldset,img {

    border:0;

}

input{

    border:1px solid #b0b0b0;

    padding:3px 5px 4px;

    color:#979797;

    width:190px;

}

address,caption,cite,code,dfn,th,var {

    font-style:normal;

    font-weight:normal;

}

ol,ul {

```

```

        list-style:none;

    }

    caption,th {

        text-align:left;

    }

    h1,h2,h3,h4,h5,h6 {

        font-size:100%;

        font-weight:normal;

    }

    q:before,q:after {

        content:"";

    }

    abbr,acronym { border:0;

    }

```

NavigationBar.css

```

/*TOP NAVIGATION BAR*/

.topnav {

    width: 70%;

    margin: 8% auto 0px;

    overflow: hidden;

    background-color: #32113f;

    border-bottom : 2px solid #1d1725;

```

```
}
```

```
.topnav a {
```

```
    color: #f2f2f2;
```

```
    text-align: center;
```

```
    padding: 14px 16px 14px 16px;
```

```
    text-decoration: none;
```

```
    font-size: 30px;
```

```
}
```

```
.topnav a:hover {
```

```
    background-color: #ddd;
```

```
    color: black;
```

```
}
```

```
.topnav a.active {
```

```
    background-color: #7020d6;
```

```
    color: white;
```

```
}
```

Input.css

```
.input-group {  
    margin: 10px 0px 10px 0px;  
}  
  
.input-group label {  
    display: block;  
    text-align: left;  
    margin: 3px;  
}  
  
.input-group input {  
    height: 30px;  
    width: 93%;  
    padding: 5px 10px;  
    font-size: 16px;  
    border-radius: 5px;  
    border: 2px solid #7527a5;  
}  
  
/*CONTENT*/  
  
/*This will be the style for the div,form */  
  
form, .content {  
    width: 60%;
```

```
margin: 0px auto;

padding: 20px;

border: 1px solid #c79cea;

background: white;

border-radius: 0px 0px 10px 10px;

}
```

```
/*BUTTONS*/
```

```
/*class = "btn" */
```

```
.btn {

padding: 10px;

font-size: 15px;

color: white;

background: #9368eb;

border: none;

border-radius: 5px;

}
```

Errors.css

```
/*ERRORS*/
```

```
.error {

width: 92%;
```



```
margin: 0px auto;

padding: 10px;

border: 1px solid #a94442;

color: #a94442;

background: #f2dede;

border-radius: 5px;

text-align: left;

}
```

```
/*SUCCESES*/
```

```
.success {

color: #30dd8b;

background: #d8f0e5;

border: 1px solid #3c7665;

margin-bottom: 20px;

}
```

Content.css

```
*{

font-family: consolas;

}
```

*/*applies to the body tag*/*

body {

font-size: 120%;

background: #f1dcfe;

*/*make a picture BG*/*

background-image: url('BG.svg');

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

.scrisu{

width: 70%;

margin: 70px auto 0px;

color: #321741;

background: white;

text-align: left;

border: 2px solid #8969d0;

border-bottom: none;

```
border-radius: 10px 10px 0px 0px;  
padding: 20px;  
}
```

BIBLIOGRAPHY

1. Download Xampp, <https://www.apachefriends.org/download.html>
2. MariaDB documentation, <https://mariadb.com/>
3. MySql Workbench connection, <https://www.inmotionhosting.com/support/website/connect-database-remotely-mysql-workbench/>
4. Download and Install Atom text editor, <https://atom.io/>
5. Quiz tables, <https://www.etutorialspoint.com/index.php/342-how-to-create-a-multiple-choice-quiz-in-php-and-mysql>
6. Login and Registration Page, <https://youtu.be/fsQoA7NMhsU>
7. Free Tutorials from W3Schools, <https://www.w3schools.com/>
8. Free Tutorials from tutorialspoint, <https://www.tutorialspoint.com/index.htm>
9. CRUD interface , <https://codingstatus.com/crud-operations-in-php-using-mysql/#:~:text=Create%20the%20CRUD%20Operations%20Using%20PHP%20Step%20By,DELETE%20%E2%80%93%20Delete%20Data%20from%20MySQL%20Table.%20>