

E-learning Website

by:

Mr. J.B. Booysen - 25060953

Semester Practical

submitted in pursuit of the degree

BACHELOR OF ENGINEERING

In

COMPUTER AND ELECTRONIC ENGINEERING

North-West -University Potchefstroom Campus

Supervisor: Mr. A. Alberts

Potchefstroom
2018

DECLARATION

We, **Juan Booysen and Ugo Louw** declare that this report is a presentation of our own original work.

Whenever contributions of others are involved, every effort was made to indicate this clearly, with due reference to the literature.

No part of this work has been submitted in the past, or is being submitted, for a degree or examination at any other university or course.

JB Booysen

Table of Contents

Table of Contents	3
List of Figures	4
1. Introduction	5
1.1 Background	5
1.2 Project Aim	5
2. Methodology	6
3. Database Design	7
3.1 Tables	8
3.2 Connections	9
4. Web programming	10
4.1 Database connection	10
4.2 Style (css)	10
4.3 Sign up page	14
4.4 Login page	17
4.5 Admin Page	18
4.5.1 Class Page	20
4.5.2 Upload Files	21
4.5.3 Student Uploads Page	24
4.5.4 Create Test Page	25
4.5.5 Mark Page	27
4.6 Student Page	28
4.6.1 Subjects Page	30
4.6.2 Study Material Page	31
4.6.3 Upload Homework Page	31
4.6.4 Test	32
4.6.5 Marks Page	35
5. Conclusion	36
6. References	36

List of Figures

Figure 1: E-Learning EER diagram	7
Figure 2: Database connection php code.....	10
Figure 3: Reset the page style.....	11
Figure 4: Css header sections	12
Figure 5: Css main sections	12
Figure 6: Css footer section.....	13
Figure 7: Css code to make the website compatible with any device	14
Figure 8: Sign up page layout.....	14
Figure 9: Sign up page different sections	15
Figure 10: Sign up php code.....	16
Figure 11: Login page layout	17
Figure 12: login php code	17
Figure 13: Admin page layout.....	18
Figure 14: Only admins can access the pages	19
Figure 15: Not Admin response	19
Figure 16: Class table	20
Figure 17: Class css & html code	20
Figure 18: Class php code to display information	21
Figure 19: Upload file before file chosen	21
Figure 20: Choose file.....	22
Figure 21: Upload files page layout.....	22
Figure 22: Upload file php code.....	22
Figure 23: Student Uploads page layout.....	24
Figure 24: Student Uploads php code	24
Figure 25: Create Test Start page layout	25
Figure 26: Add a question to the test.....	25
Figure 27: Add question html code.....	26
Figure 28: Add question php code.....	26
Figure 29: Mark page layout.....	27
Figure 30: Mark student homework php code	27
Figure 31: Student page layout.....	28
Figure 32: student page main section layout.....	28
Figure 33: Student css code	29
Figure 34: Subjects page	30
Figure 35: Mathematics subject page.....	30
Figure 36: Study Material page layout.....	31
Figure 37: Study material php code.....	31
Figure 38: Upload Homework page layout.....	31
Figure 39: Take test page layout	32
Figure 40: Take test display php code.....	32
Figure 41: Question display page	33
Figure 42: Get the questions and answers array	33
Figure 43: Displaying the questions and answers.....	33
Figure 44: Test complete display page.....	34
Figure 45: Final student test score insert	34
Figure 46: Marks page layout	35
Figure 47: Display marks for student php code.....	35

1. Introduction

1.1 Background

In the world we live in, technology is constantly improving our everyday activities. It is no different in the field of education. People are looking for more comfortable ways to build up knowledge or participate in learning courses which is easier to attend and participate in. This is where technology is becoming very important.

With E-learning websites on the internet people can participate in different courses or online programs from the comfort of their home. This fact is a key contributor to the significant increase in online enrolment on e-learning websites. Thus if someone wants to form part and create an e-learning website they have to make 100% sure that there are no errors, the website flow is very easy and every function on the websites works perfectly.

In order to create a website we have to combine different programming languages such as: php, html, css and sql. When using php and sql we ensure that the website can communicate with a database and store/retrieve information stored in the database.

1.2 Project Aim

In this practical we had to create and design a user friendly e-learning website. The website will allow different parties to log in to different sections of the website. All material uploaded to the website will be stored in a database to ensure the retrieval of the material is easily accessible. When a student registers all his details are stored in a database, however the user's password is encrypted, ensuring not even the admin can obtain access to his account.

Our website has to have the following features and functional parts in order for it to be successful:

- Different user roles (Administrator, student, etc.).
- Admin should be able to create new courses.
- Admin should be able to upload study material.
- Student should be able to upload homework done.
- The admin can create a test online.
- The student can take the test online which the admin created.
- The test should be graded online.
- The admin can grade the homework which is uploaded by the student.
- Test scores and homework scores are stored in a database for future use.
- A sensible extensible structure to facilitate the implementation of additional functionality.

All these functions listed above must be presented in an easily accessible manner and the website must be designed in such a manner that the admin as well as the student can easily browse the site.

2. Methodology

In order to ensure the practical demonstration requirements and deadline was met we followed the method discussed below:

1. Database requirements - We discussed what the requirements of the website will be and according to those requirements decided on a database design (tables that will be used in the database) and how all the different tables will link together.
2. Table layouts - After we decided on all the tables that will be required in the database, the next step was to ensure all the data will be stored neatly to be easily accessible to use again.
3. After the database was decided on, we discussed the layout of the website. How many roles we will design for and how the website will flow. We ensured the website flow is logical and very easy to understand.
4. It had to be user friendly, for even people who have zero technological capabilities.
5. The different forms/pages which the user will interact with or see will be designed using html and css code.
6. Using php code we will do all the processing work of the website, the parts which takes the user inputs and communicates with the database using SQL statements. This will be used to insert or retrieve data from different tables in the database.

3. Database Design

The database is one of the most important parts in successfully designing a functional website. Data has to be stored neatly and orderly to ensure the data can be used again. Data has to be connected in a logical manner by using primary and foreign keys.

Our EER diagram (figure 1) shows how the different tables in our database are connected. The connections between the tables are either a many-to-one, one-to-many or a one-to-one non identifying relationship.

We used the program MySQL to design the EER (Extended Entity Relationship) diagram. This is a software which helps you to design the database layout and the connections between the different tables. If you are uncertain about the SQL output you are receiving in the website, this software can be used to test the SQL statements and compare the two results with one another.

In order to run the database and see your website a local server has to be hosted. We used the software XAMPP which creates a local server on your computer and allows the database to be active and allows the website to be able to connect to the database. If both these programs are running on your computer you can forward engineer your EER diagram to your database and all your tables will already be created on the server. MYSQL connects through a dedicated port which allows it to be the host.

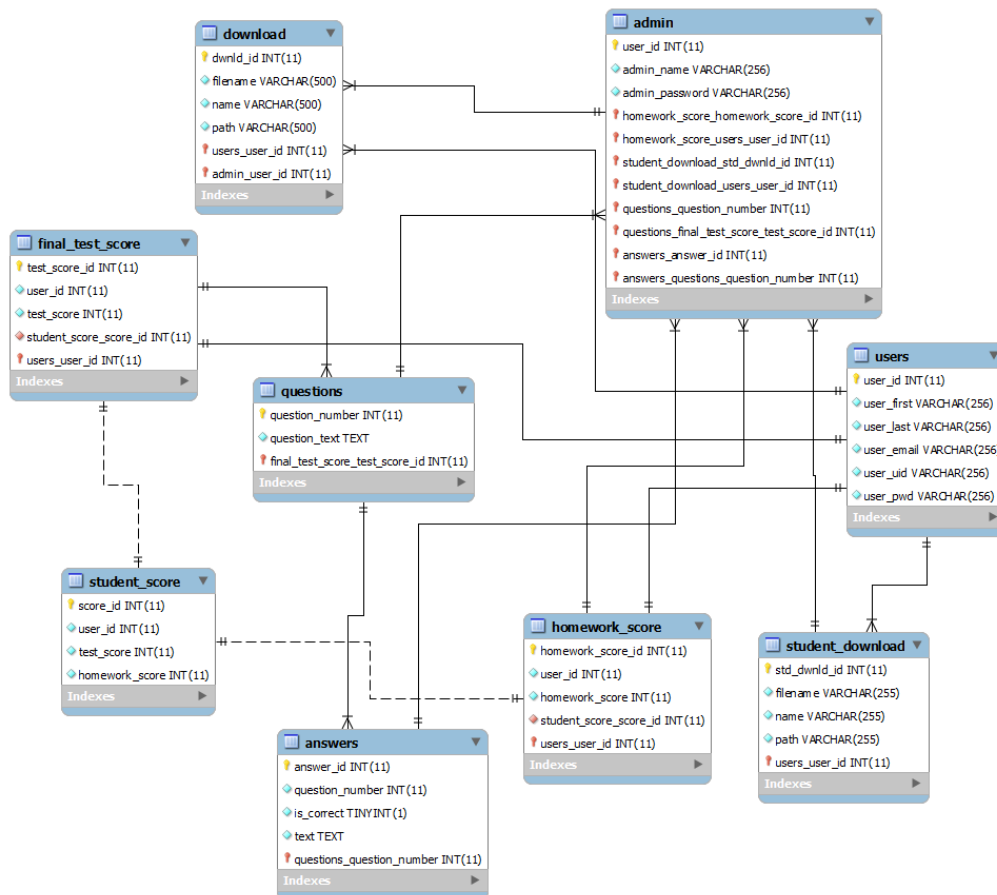


Figure 1: E-Learning EER diagram

3.1 Tables

I will discuss the need for every table below:

Users

This is the table where new students are inserted when they sign up for the courses. Also for login functions, to check if the user trying to login is actually registered.

Admin

The admin table is only there to identify the creators of the website and to showcase what their capabilities are in the website. They can upload study material, download homework grade homework and create test.

Download

The download table is used with the admin, when the admin uploads study material the filename, encrypted filename and file path where it is stored is saved. When the student opens his study material tab the download table is used again to showcase the filename of the file uploaded by the admin and gets the path to enable the student to download the files.

Student_download

As discussed above the student_download table is used in exactly the same way but the other way around. When the student uploads his homework it is saved in the student_download table. When the admin wants to grade the work it showcases the filename of the uploaded homework and uses the path to get the file.

Homework_score

The homework_score table is used by the admin to store the students homework results after it is marked, students are allowed to upload more than one attempt of the homework, and thus can receive more than one mark for the homework. That's why it is connected with the student_score table, only the most recent score for each student is stored here.

Final_test_score

Each student is only allowed to take the present test which the admin created once, thus only one score will be recorded for every student. The test are marked electronically and the score saved automatically into the table once the test is completed.

Student_score

The final test and homework score for every student is stored in this table to make the data retrieval and displaying the data for each individual student easier.

Questions

The admin can create the test online, each question is stored in this table and will be used to display the questions for the students when they take the test. Every question has multiple answers.

Answers

The admin when creating the test online provides each questions with up to 5 possible answers for the question. The admin then also indicates which answer is the correct one and the data is stored in the answers table.

3.2 Connections

The connections between the tables will be discussed below:

Student

- A student can download many files (download) uploaded by the admin (many-to-one).
- A student is only allowed to take a test once (many-to-one).
- A student r can submit many homework attempts (many-to-one).
- A student can submit many homework attempts, but only receive one mark for the homework done (one-to-one).

Admin

- An admin can upload many study materials for the student to download (many-to-one).
- An admin can grade a lot of homework for students (many-to-one).
- An admin can download many homework material from students (one-to-many).
- An admin can create many test and provide many answers to the questions (many-to-one).

Questions

- One question has multiple answers (many-to-one).
- A test has multiple questions (many-to-one).

Student score

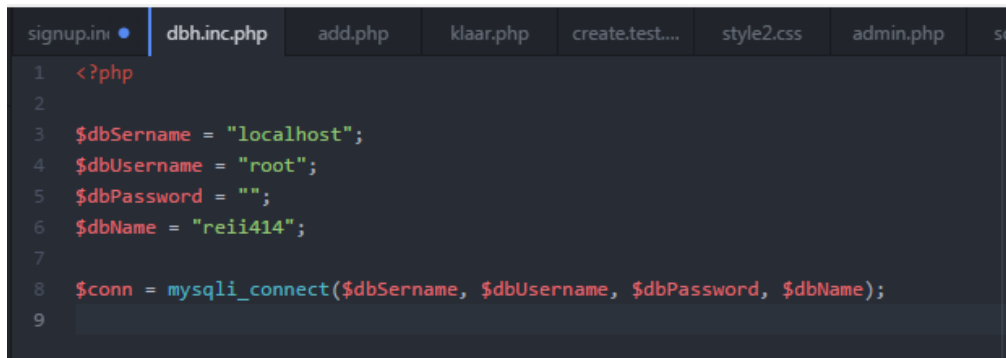
- The student score table can store a test and homework score for each student, thus it has a (one-to-one) relationship connection with the homework_score and final_test_score tables.

4. Web programming

In the section below we will discuss the creation and processing work done on each page. We will combine php, SQL, html and css code while discussing the pages.

4.1 Database connection

Firstly the page shown below is the dbh.inc.php coding page which is used to connect the different website pages with the created database and ensure communication is possible between the pages and database tables.



```
1  <?php
2
3  $dbSername = "localhost";
4  $dbUsername = "root";
5  $dbPassword = "";
6  $dbName = "reii414";
7
8  $conn = mysqli_connect($dbSername, $dbUsername, $dbPassword, $dbName);
9
```

Figure 2: Database connection php code

This is where you provide the connection variable (\$conn) with the necessary parameters to be able to connect to the database.

4.2 Style (css)

When designing the entire website most of the same styling methods was used this in several parts of this system I will refer back to these following pages when discussing the style method used to design a certain page.

```

style.css  display.mar...  math.php  geo.php  physics.php  Computer...
1  /*
2  html5doctor.com Reset Stylesheet
3  v1.6.1
4  Last Updated: 2010-09-17
5  Author: Richard Clark - http://richclarkdesign.com
6  Twitter: @rich_clark
7  */
8
9  html, body, div, span, object, iframe,
10 h1, h2, h3, h4, h5, h6, p, blockquote, pre,
11 abbr, address, cite, code,
12 del, dfn, em, img, ins, kbd, q, samp,
13 small, strong, sub, sup, var,
14 b, i,
15 dl, dt, dd, ol, ul, li,
16 fieldset, form, label, legend,
17 table, caption, tbody, tfoot, thead, tr, th, td,
18 article, aside, canvas, details, figcaption, figure,
19 footer, header, hgroup, menu, nav, section, summary,
20 time, mark, audio, video {
21     margin:0;
22     padding:0;
23     border:0;
24     outline:0;
25     font-size:100%;
26     vertical-align:baseline;
27     background:transparent;
28 }
29
30 body {
31     line-height:1;
32 }
33
34 article,aside,details,figcaption,figure,
35 footer,header,hgroup,menu,nav,section {
36     display:block;
37 }
38
39 nav ul {
40     list-style:none;
41 }
42
43 blockquote, q {
44     quotes:none;
45 }
46
47 blockquote:before, blockquote:after,
48 q:before, q:after {
49     content:'';

```

Figure 3: Reset the page style

When starting styling on any page it is good practise to reset all the current style a certain page may have and clears it. This code was obtained from a website “html5 Doctor” [1].

Each page has three sections a header, main and footer. Below we will show the header, main and footer design for the different pages.

```

110 main {
111   padding-top: 100px;
112 }
113 header {
114   position: fixed;
115   top: 0;
116   right: 0;
117   left: 0;
118   background-color: #fff;
119   width: 100%;
120   height: 100px;
121 }
122 header .header-logo {
123   font-family: Catamaran;
124   font-size: 24px;
125   font-weight: 900;
126   color: #111;
127   display: block;
128   margin: 0 auto;
129   text-align: center;
130   padding: 20px 0;
131 }
132 header nav ul {
133   display: block;
134   margin: 0 auto;
135   width: fit-content;
136 }
137 header nav ul li {
138   display: inline-block;
139   float: left;
140   list-style: none;
141   padding: 0 16px;
142 }
143 header nav ul li a {
144   font-family: Catamaran;
145   font-size: 16px;
146   color: #111;
147   text-transform: uppercase;
148 }
149 header .header-cases {
150   display: none;
151 }

```

Figure 4: Css header sections

```

186 /*Index*/
187 .index-banner {
188   width: 100%;
189   height: calc(100vh - 100px);
190   background-image: url('Web image1.jpg');
191   background-repeat: no-repeat;
192   background-position: center;
193   background-size: cover;
194   display: table;
195 }
196 .vertical-center {
197   display: table-cell;
198   vertical-align: middle;
199 }
200 .index-banner h2 {
201   font-family: Catamaran;
202   font-size: 60px;
203   font-weight: 900;
204   line-height: 70px;
205   color: #fff;
206   text-align: center;
207   text-shadow: 2px 2px 8px #111;
208 }
209 .index-banner h1 {
210   font-family: Cormorant Garamond;
211   font-size: 28px;
212   font-weight: 100;
213   font-style: italic;
214   line-height: 40px;
215   color: #fff;
216   text-align: center;
217   text-shadow: 2px 2px 8px #111;
218 }
219 .index-links div {
220   margin: 16px 16px 0;
221   width: calc(100% - 32px);
222   height: 100px;
223   background-color: #1e90ff;
224 }
225 .index-links div h3 {
226   font-family: Catamaran;
227   font-size: 28px;
228   font-weight: 600;
229   line-height: 100px;
230   color: #111;
231   text-align: center;
232 }

```

Figure 5: Css main sections

```

998 /*Footer*/
999
1000 footer {
1001   width: calc(100% - 80px);
1002   padding: 40px 40px;
1003   margin-top: 20px;
1004   background-color: #111;
1005   overflow: hidden;
1006 }
1007
1008 footer ul {
1009   width: fit-content;
1010   float: left;
1011   padding-left: 20px;
1012 }
1013
1014 footer ul li {
1015   display: block;
1016   list-style: none;
1017 }
1018
1019 footer ul a {
1020   font-family: Catamaran;
1021   font-size: 24px;
1022   color: #fff;
1023   line-height: 40px;
1024 }
1025
1026
1027 .footer-sm {
1028   width: 50px;
1029   float: right;
1030 }
1031
1032 .footer-sm img {
1033   width: 100%;
1034   margin-bottom: 10px;
1035 }

```

Figure 6: Css footer section

The code above showcases how the header, main and footer sections of the pages was styled.

We wanted to make this website easily accessible to all users from anywhere using any device. That's why when we designed this page we did it in the mobile view and using the code in figure 7 make it adjustable when viewing the website on other devices.

```

152  @media only screen and (min-width: 1000px) {
153      header .header-logo {
154          margin: 31px 0;
155          text-align: left;
156          line-height: 38px;
157          padding: 0 20px 0 40px;
158          border-right: 2px solid #111;
159          float: left;
160      }
161      header nav ul {
162          margin: 20px 0px 0px 20px;
163          float: left;
164      }
165      header nav ul li a {
166          line-height: 60px;
167      }
168      header .header-sign {
169          display: block;
170          font-family: Catamaran;
171          font-size: 16px;
172          color: #111;
173          text-transform: uppercase;
174          line-height: 38px;
175          border: 1px solid #111;
176          float: right;
177          margin-right: 40px;
178          margin-top: 30px;
179          padding: 0 20px;
180      }
181  }
182

```

Figure 7: Css code to make the website compatible with any device

4.3 Sign up page

ILearn

ABOUT US

THE TEAM

CONTACT US

SIGN UP

First name

Last name

Email

Username

Password

Sign up

LOGIN

Home

ILearn

About us

The team

Contact us

Figure 8: Sign up page layout

```

<?php
    session_start();
    ?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>E-learning</title>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800">
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <header>
            <a href="Home.php" class="header-logo">Ilearn</a>
            <nav>
                <ul>
                    <li><a href="about.html">About us</a></li>
                    <li><a href="team.html">The team</a></li>
                    <li><a href="contact.html">Contact us</a></li>
                </ul>
            </nav>
        </header>
        <main>
            <section class="main-container">
                <h2>Sign up</h2>
                <div class="main-wrapper">
                    <form class="signup-form" action="includes/signup.inc.php" method="POST">
                        <input type="text" name="first" placeholder="First name">
                        <input type="text" name="last" placeholder="Last name">
                        <input type="text" name="email" placeholder="Email">
                        <input type="text" name="uid" placeholder="Username">
                        <input type="password" name="pwd" placeholder="Password">
                        <button type="submit" name="submit">Sign up</button>
                    </form>
                    <a href="login.php">Login</a>
                </div>
            </section>
        </main>
        <div class="wrapper">
            <footer>
                <ul class="footer-links-main">
                    <li><a href="Home.php" >Home</a></li>
                    <li><a href="Ilearn.html" >Ilearn</a></li>
                    <li><a href="about.html" >About us</a></li>
                    <li><a href="team.html" >The team</a></li>
                    <li><a href="contact.html" >Contact us</a></li>
                </ul>
            </footer>
        </div>
    </body>
</html>

```

Figure 9: Sign up page different sections

Figure 9 showcases the basic layout of all the pages in the website. When we look closely it is very important to include a “link” to link the current page with the stylesheet to achieve the desired outcome shown in figure 8.

```

1  <?php
2  /* kjk of die submit knopje gedruk is voor die kode hardloop*/
3  if (isset($_POST['submit'])) {
4      include_once 'dbh.inc.php';
5
6      $first = mysqli_real_escape_string($conn, $_POST['first']);
7      $last = mysqli_real_escape_string($conn, $_POST['last']);
8      $email = mysqli_real_escape_string($conn, $_POST['email']);
9      $uid = mysqli_real_escape_string($conn, $_POST['uid']);
10     $pwd = mysqli_real_escape_string($conn, $_POST['pwd']);
11
12     //Error handlers
13     //Check if there is empty fields
14     if (empty($first) || empty($last) || empty($email) || empty($uid) || empty($pwd)) {
15         header("Location: ../sign-up.php?signup=empty");
16         exit();
17     } else {
18         //Check if input characters are valid
19         if (!preg_match("/^[a-zA-Z_-]*$/", $first) || !preg_match("/^[a-zA-Z_-]*$/", $last)) {
20             header("Location: ../sign-up.php?signup=invalid data");
21             exit();
22         } else {
23             //Check if email is valid
24             if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
25                 header("Location: ../sign-up.php?signup=email");
26                 exit();
27             } else {
28                 $sql = "SELECT * FROM users WHERE user_uid = '$uid'";
29                 $result = mysqli_query($conn, $sql);
30                 $resultCheck = mysqli_num_rows($result);
31                 if ($resultCheck > 0) {
32                     header("Location: ../sign-up.php?signup=usertaken");
33                     exit();
34                 } else {
35                     //Hashing the Password
36                     $hashedPwd = password_hash($pwd, PASSWORD_DEFAULT);
37                     //Insert the user into the database
38                     $sql = "INSERT INTO users (user_first, user_last, user_email, user_uid, user_pwd) VALUES ($first, $last, $email, $uid, '$hashedPwd')";
39                     mysqli_query($conn, $sql);
40                     header("Location: ../login.php?signup=success");
41                     exit();
42                 }
43             }
44         }
45     } else {
46         header("Location: ../sign-up.php");
47         exit();
48     }
49 }

```

Figure 10: Sign up php code

It is always good practise to do all the php processing code and SQL communication on a different page and link the page to the desired page where you want to use the code. This is the signup php and SQL code where the user inputs are received. To ensure users can't hack our system we get the data using the "mysqli_real_escape_string" function which ensures if SQL statements are type into the input boxes it is stored just like any other string and has no effect on our system.

Before any of these functions are executed we firstly test if the submit button was pressed with the "isset" function.

After the data is received from all the fields we next check if a field is empty. If no field is empty we check is if the characters entered are legit characters we allow to be entered with the "preg_match" function. At the email field we check to see if the email entered is a correct email using the "FILTER_VALIDATE_EMAIL" function. After we checked the email, we hash (encrypt) the user's password to ensure nobody could gain access to their account. If all the above mentioned fields are correct we execute the SQL statement to insert the new user into the users table in the database.

If any of the fields are incorrect the page is redirected and the fields are cleared, a fault message is showed in the url of the page and the user has a change to enter their data again.

4.4 Login page

ILearn

ABOUT US THE TEAM CONTACT US

LOGIN PAGE

Username

Password

Login

SIGN UP


Home

ILearn

About us

The team

Contact us






Figure 11: Login page layout

```
1  <?php
2  session_start();
3
4  if (isset($_POST['submit'])) {
5      include 'dbh.inc.php';
6
7      $uid = mysqli_real_escape_string($conn, $_POST['uid']);
8      $pwd = mysqli_real_escape_string($conn, $_POST['pwd']);
9      //Error handlers//Check if inputs are empty
10     if (empty($uid) || empty($pwd)) {
11         header("Location: ../Home.php?login=empty");
12         exit();
13     } else {
14         if ($uid=="admin" & $pwd=="admin") {
15             $_SESSION['u_uid'] = $uid;
16             header("Location: ../admin.php?AdminLogin=success");
17             exit();
18         } else {
19             $sql = "SELECT * FROM users WHERE user_uid='$uid'";
20             $result = mysqli_query($conn, $sql);
21             $resultCheck = mysqli_num_rows($result);
22             if ($resultCheck < 1) {
23                 header("Location: ../Home.php?login=username");
24                 exit();
25             } else {
26                 if ($row = mysqli_fetch_assoc($result)) {
27                     //De-hashing the PASSWORD
28                     $hashedPwdCheck = password_verify($pwd, $row['user_pwd']);
29                     if ($hashedPwdCheck == false) {
30                         header("Location: ../Home.php?login=password");
31                         exit();
32                     } elseif ($hashedPwdCheck == true) {
33                         //Login the user here
34                         $_SESSION['u_id'] = $row['user_id'];
35                         $_SESSION['u_first'] = $row['user_first'];
36                         $_SESSION['u_last'] = $row['user_last'];
37                         $_SESSION['u_email'] = $row['user_email'];
38                         $_SESSION['u_uid'] = $row['user_uid'];
39                         header("Location: ../student.php?login=success");
40                         exit();
41                     }
42                 }
43             }
44         }
45     } else {
46         header("Location: ../Home.php?login=error");
47         exit();
48     }
49 }
```

Figure 12: login php code

As we can see the login page has the same layout as the sign up page, thus we will not show or discuss its html or css code. It is the same as figure 3 – 7 and figure 9.

We once again check firstly if the submit (“Login”) button was pressed with the isset function before running the code below. If the button was pressed we once again access the data from the input boxes in a safe manner to ensure our site is not hackable. Then we use the SQL statement, SELECT, to fetch the data in the database which corresponds with the username entered. If the username is correct but the password does not correspond with it, access will not be granted, both have to be correct, because the password was hashed (encrypted) when storing it in the database we have to de-hash the password when trying to get access to the website.

For admin access we hardcoded the username and password for the admin, because we wanted to showcase the website can access different roles. When the login is approved by the program we assign the different session variables, “\$_SESSION” to be able to access them at any point later in the website.

4.5 Admin Page

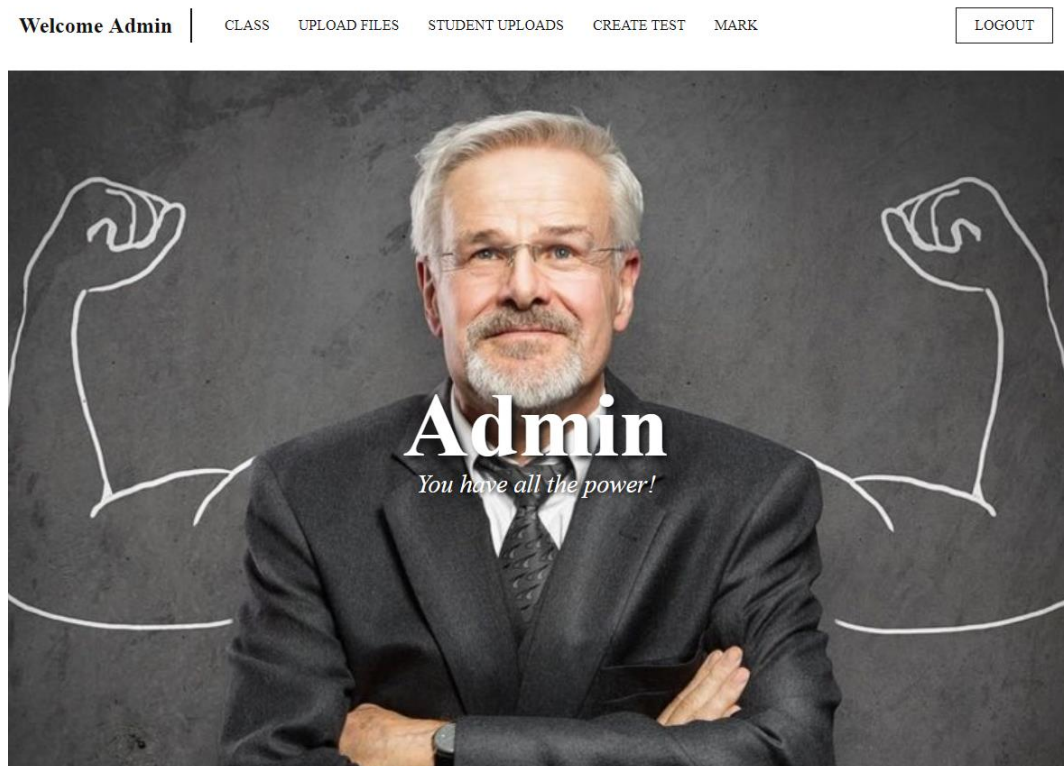


Figure 13: Admin page layout

The admin page uses exactly the same page layout design as all the above. However here we replace the input boxes and buttons with a background image. We will show how this is done in figures 31 - 33. The “Class” button displays all the students registered for the course in an orderly fashion in table form. The “Upload Files” button is where the admin will go in order to upload the

study material for the students to then download on their side. The “Student Uploads” button is where the admin will go to download the homework which the student uploaded. When the admin clicks on the “Create Test” button they can start creating an online test for the students to take and lastly the “Mark” button is where the admin can mark the homework of an individual student and upload their mark to the database.

In order to ensure the student can’t access any of the admin’s pages we included the following php code which prohibits this.

```
1  <?php
2      session_start();
3      $uid = $_SESSION['u_uid'];
4      if ($uid != 'admin') {
5          header("Location: sorry.php?Not Admin");
6          exit();
7      }
8  ?>
9
```

Figure 14: Only admins can access the pages

How this piece of codes work is, when a student or admin logs in. They get a session user id variable assigned to them as shown in figure 12. This code then checks what the user id is with the `$_SESSION['u_uid']` function. If their user id is not 'admin', which it should be, they may not get access to these pages and the following appears for them.

Sorry you are not an Admin!

Figure 15: Not Admin response

4.5.1 Class Page

CLASS TABLE

Id	First name	Last name	Email
1	Daniel	Nielson	mmtuts@gmail.com
2	Juan	Booyesen	booyesen.juanbrendan0@gmail.com
3	CW	Smith	cw.smith0@gmail.com
4	New	Name	NN@gmail.com
5	Theune	vanderWalt	theune.vdw@gmail.com
6	lekker	laag	ll@gmail.com
7	Probeer	nog n keer	PK@gmail.com
8	Die	een werk	DE@gmail.com
9	Charlie	Boon	Boon@gmail.com
10	CW	Smith	cw@gmail.com
11	MJ	Bezuidenhout	a@B.C
12	Brendan	Booyesen	booyesen.juanbrendon0@gmail.com

Figure 16: Class table

In order to display all the students registered on the website we used a combination of html, css and php code combined in one form.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>E-learning</title>
    <style type="text/css">
      table {
        border-collapse: collapse;
        width: 100%;
        color: #111;
        font-family: monospace;
        font-size: 100px;
        text-align: left;
      }
      th {
        background-color: #111;
        color: black;
        font-size: 24px;
      }
      tr:nth-child(even) {background-color: #f2f2f2}
    </style>
  </head>
  <body>
```

Figure 17: Class css & html code

The code in figure 17 was used in the php document where all the html code normally goes. Just a little bit lower on the page we set the title names and ensures they are displayed neatly on the pages. The php code is used to ensure all the information wanted is retrieved from the users table and then displayed neatly.

```

<main>
<section class="geo-links">
<div class="geo-wrapper">
<h2>Class table</h2>
<table>
<tr>
<th>Id</th>
<th>First name</th>
<th>Last name</th>
<th>Email</th>
<?php
    $sql = "SELECT user_id, user_first, user_last, user_email FROM users;";
    $result = mysqli_query($conn, $sql);
    $resultCheck = mysqli_num_rows($result);
    if ($resultCheck > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            echo "<tr><td>". $row["user_id"]. "</td><td>". $row["user_first"]. "</td><td>". $row["user_last"]. "</td><td>". $row["user_email"]. "</td></tr>";
        }
        echo "</table>";
    }
    ?>
</tr>
</table>
</div>
</section>
</main>

```

Figure 18: Class php code to display information

The “mysqli_num_rows” functions the following way, when the query is executed it receives an integer value of how many rows the table had which corresponded with the SQL query. Then we use a while function to ensure it runs for all the instances of the amount of students registered.

4.5.2 Upload Files

Admin

CLASS

STUDENT UPLOAD

CREATE TEST

MARKS

UPLOAD FILES

Choose File

No file chosen

Upload

Figure 19: Upload file before file chosen

When the upload files page is opened the admin clicks on the “choose file” button, he/she then selects a file they which the upload for the students and the file name will be shown in a display box on the screen.

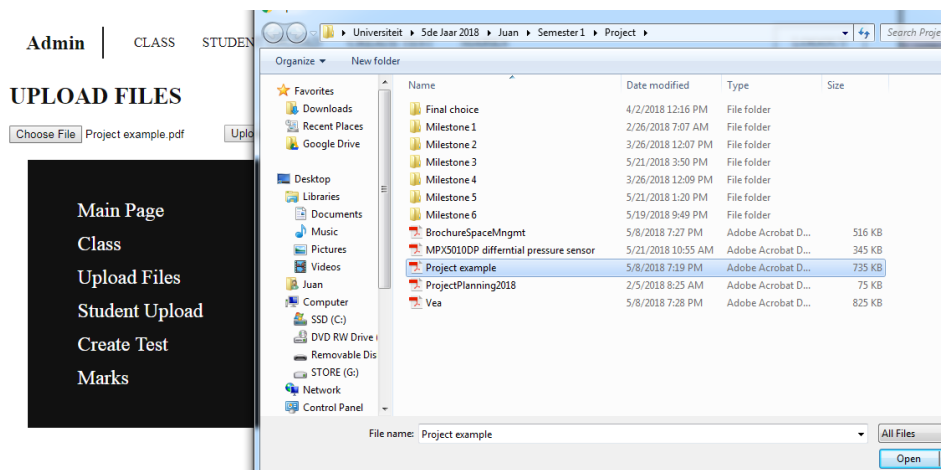


Figure 20: Choose file

If the admin is happy he/she can click on the “Upload” button and the file will be uploaded for the students to be able to download later.

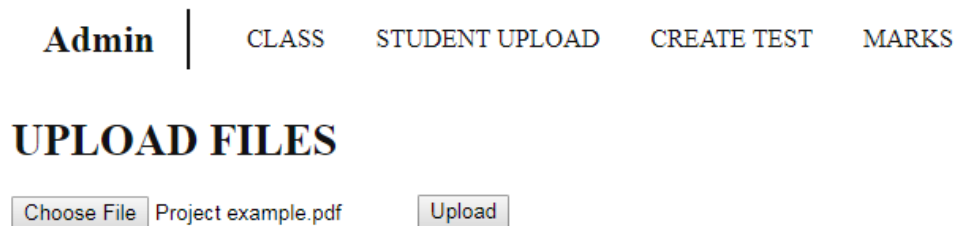


Figure 21: Upload files page layout

```

if (isset($_POST['file'])) {
    if (isset($_POST['submit'])) {
        $file = $_FILES['file'];

        $fileName = $_FILES['file']['name'];
        $fileTmpName = $_FILES['file']['tmp_name'];
        $fileSize = $_FILES['file']['size'];
        $fileError = $_FILES['file']['error'];
        $fileType = $_FILES['file']['type'];

        $fileExt = explode('.', $fileName);
        $fileActualExt = strtolower(end($fileExt));

        $allowed = array('jpg', 'jpeg', 'png', 'pdf', 'docx');

        if (in_array($fileActualExt, $allowed)) {
            if ($fileError === 0) {
                if ($fileSize < 10000000) {
                    $fileNameNew = uniqid('', true) . "." . $fileActualExt;
                    $fileDestination = 'uploads/' . $fileNameNew;
                    move_uploaded_file($fileTmpName, $fileDestination);
                    $sql = "INSERT INTO download (filename, name, path) VALUES ('$fileNameNew', '$fileName', '$fileDestination')";
                    mysqli_query($conn, $sql);
                    header("Location: upload-index.php?Upload=success");
                    exit();
                } else {
                    echo "Your file is too big!";
                }
            } else {
                echo "There was an error uploading your file!";
            }
        } else {
            echo "You cannot upload files of this type!";
        }
    }
} else {
    echo "You have not selected a file!";
}

```

Figure 22: Upload file php code

We again firstly check if a file was chosen by the admin. Then if the “Upload” button was clicked before allowing the program to run any further. We then check what the file type is, by using the “explode” function. We allow the website to be able to accept certain types of files and others not. We check if the file chosen is the correct else an error message appears on the screen.

A file may also only be a certain size. The files chosen is moved to a certain folder on the server computer and the file path is saved together with the filename in a database table. This will allow us to be able to fetch the files later when displaying them for the user to download.

4.5.3 Student Uploads Page

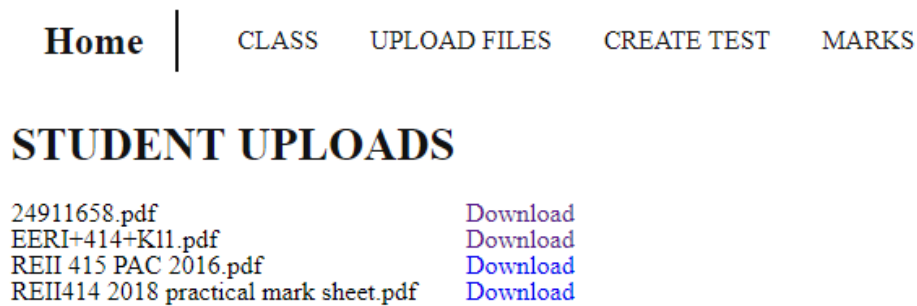


Figure 23: Student Uploads page layout

As we can see this is exactly how the “Study material” page will look like in the students side. All the files uploaded by the students to be graded are displayed here. The admin can download each file and grade them individually.

```
<main>
<section class="geo-links">
  <div class="geo-wrapper">
    <h2>Student Uploads</h2>
    <?php

      $sql = "SELECT * FROM student_download";
      $result = mysqli_query($conn,$sql);
      echo "<table>";
      while($row = mysqli_fetch_array($result)){
        echo "<tr>";
        echo "<td>"; echo $row["name"]; echo "</td>";
        echo "<td>";?><a href=?php echo $row["path"]; ?>">Download</a> <?php echo "</td>";

        echo "</tr>";
      }
      echo "</table>";
    ?>
  </div>
</section>
</main>
```

Figure 24: Student Uploads php code

The filenames and paths of the files the students uploaded is saved in a “studen_download” table in the database. All the files currently stored in the database is then displayed using the SQL SELECT statement. Using a combination of php and html code the admin can then download the files each individually by clicking on the “Download” button.

4.5.4 Create Test Page



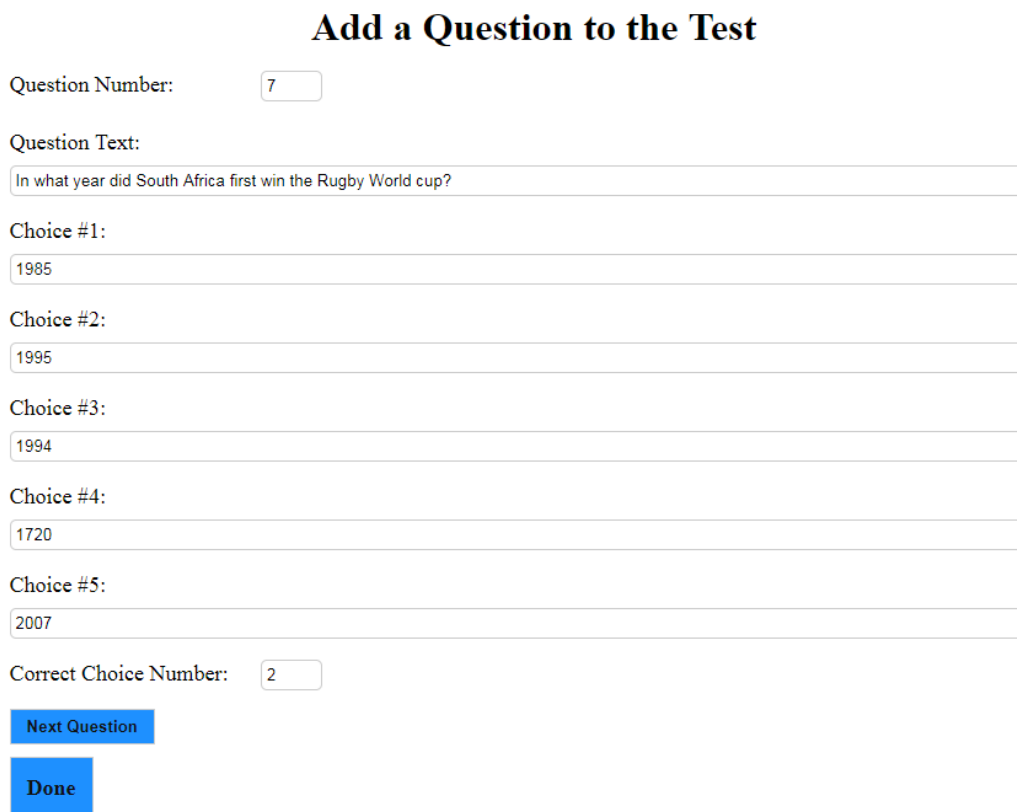
Admin | CLASS UPLOAD FILES STUDENT UPLOAD MARKS

Create Test

Create New Test

Figure 25: Create Test Start page layout

When the admin clicks on the “Create Test” page they are redirected to this page, when clicking on the “Create New Test” button they can start by creating a new test for the students to take online. We originally put this button in to clear the questions and answers tables in the database, allowing the admin to create a totally new test from the start. However this was not required for the practical and we left that function out of the website. When the button is clicked the following page appears.



Add a Question to the Test

Question Number: 7

Question Text:
In what year did South Africa first win the Rugby World cup?

Choice #1:
1985

Choice #2:
1995

Choice #3:
1994

Choice #4:
1720

Choice #5:
2007

Correct Choice Number: 2

Next Question

Done

Figure 26: Add a question to the test

The form displayed in figure 26 was obtained by using the html code in figure 27.

```

<div class="container">
<h2>Add a Question to the Test</h2>
<?php
if (isset($msg)) {
    echo '<p>'.$msg.'</p>';
}
?>
<form action="add.php" method="POST">
<p>
<label>Question Number: </label>
<input type="number" value="<?php echo $next; ?>" name="question_number">
</p>
<p>
<label>Question Text: </label>
<input type="text" name="question_text">
</p>
<p>
<label>Choice #1: </label>
<input type="text" name="choice1">
</p>
<p>
<label>Choice #2: </label>
<input type="text" name="choice2">
</p>
<p>
<label>Choice #3: </label>
<input type="text" name="choice3">
</p>
<p>
<label>Choice #4: </label>
<input type="text" name="choice4">
</p>
<p>
<label>Choice #5: </label>
<input type="text" name="choice5">
</p>
<p>
<label>Correct Choice Number: </label>
<input type="number" name="correct_choice">
</p>
<p>
<input type="submit" name="submit" value="Next Question">
</p>
<p>
<a href="admin.php" class="start">Done</a>
</p>
</form>
</div>

```

Figure 27: Add question html code

Finally the php code for the add question page is shown in figure 28.

```

<?php
if (isset($_POST['submit'])) {
    //Get the post variables
    $question_number = mysqli_real_escape_string($conn, $_POST['question_number']);
    $question_text = mysqli_real_escape_string($conn, $_POST['question_text']);
    $correct_choice = mysqli_real_escape_string($conn, $_POST['correct_choice']);
    //Choices array
    $choices = array();
    $choices[1] = mysqli_real_escape_string($conn, $_POST['choice1']);
    $choices[2] = mysqli_real_escape_string($conn, $_POST['choice2']);
    $choices[3] = mysqli_real_escape_string($conn, $_POST['choice3']);
    $choices[4] = mysqli_real_escape_string($conn, $_POST['choice4']);
    $choices[5] = mysqli_real_escape_string($conn, $_POST['choice5']);
    //Question query
    $sql = "INSERT INTO questions (question_number, question_text) VALUES ('$question_number', '$question_text')";
    $result = mysqli_query($conn, $sql);
    //Validate insert
    if ($result) {
        foreach($choices as $choice => $value){
            if ($value != '') {
                if ($correct_choice == $choice) {
                    $is_correct = 1;
                } else {
                    $is_correct = 0;
                }
                //choice query
                $sql = "INSERT INTO answers (question_number, is_correct, text) VALUES ('$question_number', '$is_correct', '$value')";
                $insert_row = mysqli_query($conn, $sql);
                //Validate INSERT
                if ($insert_row) {
                    continue;
                } else {
                    die('Error: ('.$mysqli->errno . ').'.$mysqli->error);
                }
            }
        }
        $msg = 'Question has been added';
    }
}
//Get total number of Questions
$sql = "SELECT * FROM questions ";
$result = mysqli_query($conn, $sql);
$total = mysqli_num_rows($result);
$next = $total+1;
?>

```

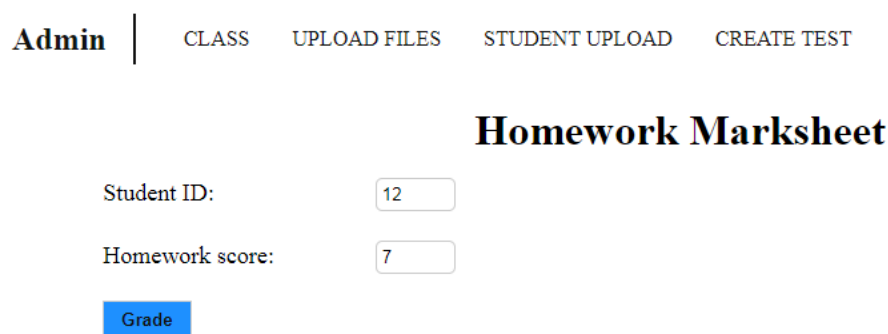
Figure 28: Add question php code

When the admin wants to add a question we again check if the submit button was clicked before executing the code. We make sure nobody can hack the system by making the input values safe to store in the database.

We then use a SQL statement to insert the question into the questions table in the database. The admin also enters all the different choices for the question and at the bottom indicates which one of the choices is the correct choice. This information is then stored into the answers table.

By clicking on the next question button the information is inserted into the different tables and stored in the database. If the Done button is clicked the admin has finished creating the test and the page exits.

4.5.5 Mark Page



Admin | CLASS | UPLOAD FILES | STUDENT UPLOAD | CREATE TEST

Homework Marksheet

Student ID:

Homework score:

Figure 29: Mark page layout

The final page which the admin can use is the mark page. By using the class page the admin can get the student id of the specific student which homework he/she wants to grade from the table. The admin then enters the student ID and their mark and the information is stored in the homework_score table as shown in figure 30.

```
<?php
include_once 'includes/dbh.inc.php';
if (isset($_POST['submit'])) {

    $student_id = mysqli_real_escape_string($conn, $_POST['student_id']);
    $homework_score = mysqli_real_escape_string($conn, $_POST['homework_score']);
    $sql = "INSERT INTO homework_score (user_id, homework_score) VALUES ('$student_id', '$homework_score')";
    mysqli_query($conn, $sql);
    $msg = 'Homework score has been added';

}
```

Figure 30: Mark student homework php code

4.6 Student Page



Figure 31: Student page layout

When a student logs into his/her account this is the front page they see. The page layout is once again the same as the other above, however in the main section we replaced the input boxes and buttons with a picture and writing as will be shown in figures 31-33.

The header tab navigates very easy to any page the student wants to access. The username, “Brendan”, always brings you back to the student main page. The “subject” button redirects the page to the current subjects that the website offer.

The “Study material” button opens up all the files the admin uploaded for the students. The students can then download the files and study them. The “Upload homework” button takes the student to a page where they can upload their homework which they have done for the admin to grade. The “Test” button allows the student to take a test if they haven’t already and they will receive a mark for their answers. Lastly the “Marks” button takes the student to a mark page where they can see their marks received for their homework and the test completed.

```
</header>
<main>
  <section class="student-banner">
    <div class="vertical-student">
      <h2>Welcome to ILearn</h2>
      <h1>Where we teach you the things you never thought you wanted to know about!<br>In a very easy manner!</h1>
    </div>
  </section>
</main>
<div class="wrapper">
  <footer>
```

Figure 32: student page main section layout

```

678  /*Student Page*/
679
680  .student-banner {
681      width: 100%;
682      height: calc(100vh - 100px);
683      background-image: url('study2.jpeg');
684      background-repeat: no-repeat;
685      background-position: center;
686      background-size: cover;
687      display: table;
688  }
689  .vertical-student {
690      display: table-cell;
691      vertical-align: middle;
692  }
693  .student-banner h2 {
694      font-family: Catamaran;
695      font-size: 100px;
696      font-weight: 900;
697      line-height: 90px;
698      color: #fff;
699      text-align: center;
700      text-shadow: 4px 4px 10px #111;
701  }
702
703  .student-banner h1 {
704      font-family: Cormorant Garamond;
705      font-size: 28px;
706      font-weight: 100;
707      font-style: italic;
708      line-height: 40px;
709      color: #fff;
710      text-align: center;
711      text-shadow: 2px 2px 8px #111;
712  }
713

```

Figure 33: Student css code

In the css code we can see where we include the background image.

4.6.1 Subjects Page

The subjects page displays all the courses available on the website. We forgot to make the subjects adjustable, add new courses or take away others, thus these subjects are fixed and the only four we provide.

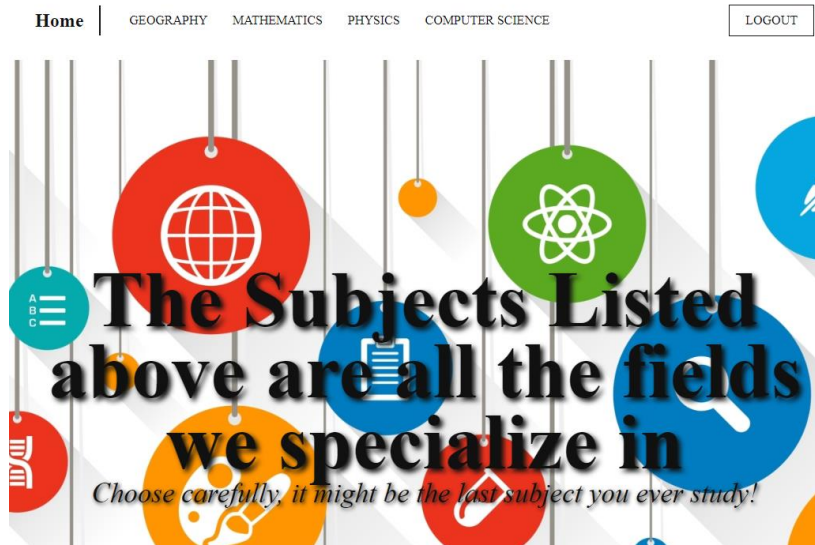


Figure 34: Subjects page

When you click on any one of the subjects in the header it only provides a description of what that subject is about.



Figure 35: Mathematics subject page

4.6.2 Study Material Page

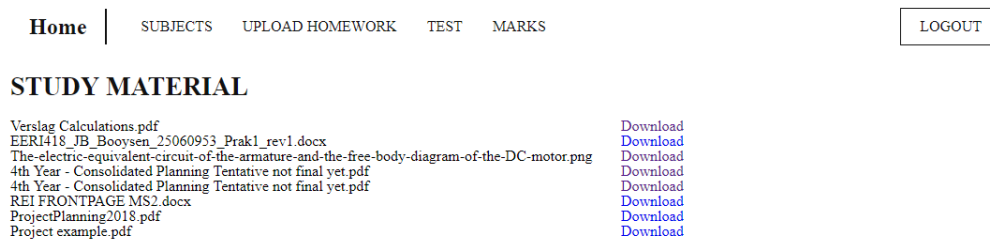


Figure 36: Study Material page layout

The study material page is where the files uploaded by the admin to study is displayed and made available to download. We can see that the file we uploaded in figure 21 is displayed here and ready to download.

```
<main>
  <section class="geo-links">
    <div class="geo-wrapper">
      <h2>Study Material</h2>
      <?php
        $sql = "SELECT * FROM download";
        $result = mysqli_query($conn,$sql);
        echo "<table>";
        while($row = mysqli_fetch_array($result)){
          echo "<tr>";
          echo "<td>"; echo $row["name"]; echo "</td>";
          echo "<td>";?><a href="<?php echo $row["path"]; ?>">Download</a> <?php echo "</td>";

          echo "</tr>";
        }
        echo "</table>";
      ?>
    </div>
  </section>
</main>
```

Figure 37: Study material php code

Figure 37 shows how the files that was uploaded and stored in a file on the server computer as well as the file path in a table in the database is fetched and displayed on this page, available to download and use as the student wishes.

4.6.3 Upload Homework Page

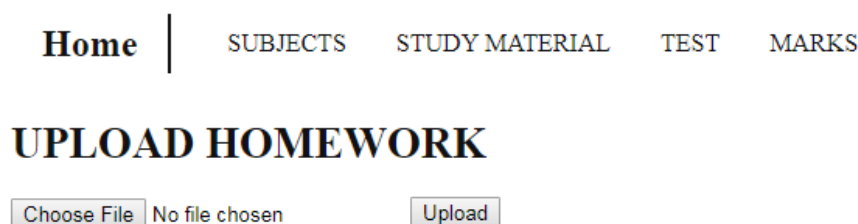


Figure 38: Upload Homework page layout

The upload homework page uses exactly the same principles as the upload files page on the admin's side. The program checks if the "choose file" button is clicked, if the file is the correct type, if the file is too large and then if all the above mentioned facts are correct the file gets upload to a folder on the server computer and the path where the file is stored is saved together with the filename in the database.

4.6.4 Test

The Test button takes the student to the page where they can participate and take a test created by the admin. The page shows the amount of questions in the test, what type of test it is and the estimated time it should take the student to complete the test.

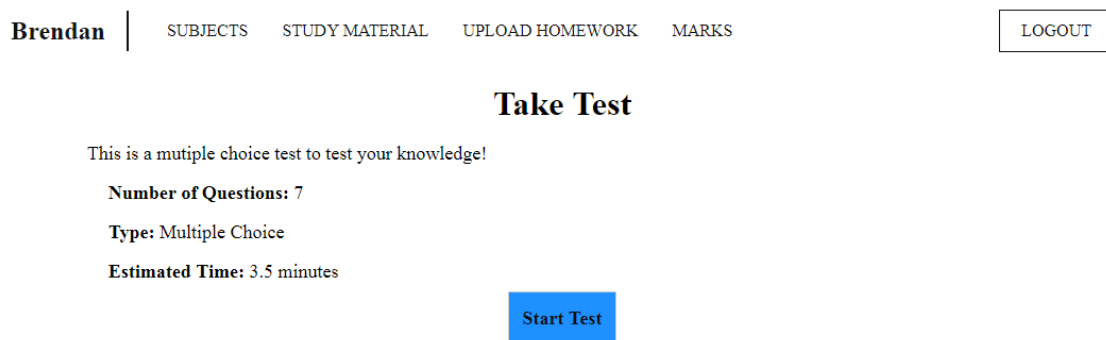


Figure 39: Take test page layout

```
<?php
    session_start();
    $_SESSION['score'] = 0;
?>
<?php

    include_once 'includes/dbh.inc.php';
    $user = $_SESSION['u_id'];
    //Get total number of Questions
    $sql = "SELECT * FROM questions ";
    $result = mysqli_query($conn, $sql);
    $total = mysqli_num_rows($result);

    $sql = "SELECT * FROM final_test_score WHERE user_id = '$user'";
    $result = mysqli_query($conn, $sql);
    $resultCheck = mysqli_num_rows($result);

    if ($resultCheck > 0) {
        header("Location: klaar.php");
        exit();
    }
}
```

Figure 40: Take test display php code

The php code above is used to calculate the total amount of questions stored in the database. Also a student is only allowed to take the test once, we check here if the students ID is already stored in the test_score table, and if it is, they will not be able to take the test again.

Question 7 of 7

In what year did South Africa first win the Rugby World cup?

- ☐ 1985
- ☒ 1995
- ☐ 1994
- ☐ 1720
- ☐ 2007

Submit Answer

Figure 41: Question display page

All the questions are displayed in the following manner with radio buttons. Each time you are satisfied with your answer you click submit answer and the answer is saved and cross-referenced with the correct answer in the database. If the answers match 1 point is added to the students end score result.

```
<?php
include_once 'includes/dbh.inc.php';
//Set question Number
$number = (int)$GET['n'];
//Get $total
$sql = "SELECT * FROM questions";
$result = mysqli_query($conn, $sql);
$total = mysqli_num_rows($result);
//Get question
$sql = "SELECT * FROM questions WHERE question_number = $number";
$result = mysqli_query($conn,$sql);
$question = mysqli_fetch_assoc($result);
//Get choices
$sql = "SELECT * FROM answers WHERE question_number = $number";
$choices = mysqli_query($conn,$sql);
?>
```

Figure 42: Get the questions and answers array

The php code above in figure 42 is what we used to obtain all the questions and all the answers and store them in the \$question and \$choices variables.

```
<main>
  <div class="container">
    <div class="current">
      Question <?php echo $question['question_number']; ?> of <?php echo $total; ?>
    </div>
    <p class="question">
      <?php echo $question['question_text']; ?>
    </p>
    <form action="process.php" method="POST">
      <ul class="choices">
        <?php while ($row = mysqli_fetch_assoc($choices)): ?>
          <li><input name="choice" type="radio" value="<?php echo $row['answer_id']; ?>" /><?php echo $row['text']; ?></li>
        <?php endwhile; ?>
      </ul>
      <button type="submit" name="submit" class="submit">Submit Answer</button>
      <input type="hidden" name="number" value="<?php echo $number; ?>">
    </form>
  </div>
</main>
```

Figure 43: Displaying the questions and answers

In figure 43 we used a combination of php and html code to display all the questions and answers in a neat and orderly fashion.

After the student has submitted his/her answer to the last question figure 44 appears and displays the students final score out of the amount of questions answered.

You're Done!

Well done! You have completed the test

Final Score: 7

Figure 44: Test complete display page

```
<?php
session_start();
include 'login.inc.php';

?>
<?php
include_once 'includes/dbh.inc.php';
$user = $_SESSION['u_id'];
$score = $_SESSION['score'];
$sql = "INSERT INTO final_test_score (user_id, test_score) VALUES ('$user ', '$score')";
$result = mysqli_query($conn, $sql);
?>

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Test</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800">
<link rel="stylesheet" href="style2.css">
</head>
<body>
<header>
<a href="student.php" class="header-logo">
<?php echo $_SESSION['u_id']; ?>
</a>
<nav>
<ul>
<li><a href="subject.php">Subjects</a></li>
<li><a href="study.material.php">Study material</a></li>
<li><a href="homework-index.php" >Upload homework</a></li>
<li><a href="display.mark.php">Marks</a></li>
</ul>
<a href="Home.php" class="header-sign">Logout</a>
</nav>
</header>
<main>
<div class="container">
<h2>You're Done!</h2>
<p>Well done! You have completed the test</p>
<p>Final Score: <?php echo $_SESSION['score']; ?></p>
</div>
</main>
```

Figure 45: Final student test score insert

The final score the student received is displayed for them to see and then stored into the database for future use.

4.6.5 Marks Page

Brendan		SUBJECTS	STUDY MATERIAL	UPLOAD HOMEWORK	TEST
MARKS					
Homework Result			Test Result		
7			7		

Figure 46: Marks page layout

After the admin has graded the student's homework and the student has completed the test, their results for both is displayed in table format. Clearly the marking system works, because in figure 29 the admin gave user 12 (Brendan) 7 for his homework and he just received 7 out of 7 for his test. Thus both marks display correctly.

```
<main>
<section class="geo-links">
  <div class="geo-wrapper">
    <h2>Marks</h2>
    <table>
      <tr>
        <th>Homework Result</th>
        <th>Test Result</th>
      </tr>
      <tr>
        <td>
          <?php
            $sql = "SELECT user_id, homework_score FROM homework_score WHERE user_id = '$user'";
            $sql2 = "SELECT user_id, test_score FROM final_test_score WHERE user_id = '$user'";
            $result = mysqli_query($conn, $sql);
            $result2 = mysqli_query($conn, $sql2);
            $resultCheck = mysqli_num_rows($result);
            $resultCheck2 = mysqli_num_rows($result2);
            if ($resultCheck > 0 || $resultCheck2 > 0) {
              $row = mysqli_fetch_assoc($result);
              $row1 = mysqli_fetch_assoc($result2);
              echo "<tr><td>". $row["homework_score"]."</td><td>".$row1["test_score"]."</td></tr>";
            }
          </td>
        </tr>
      </table>
    </div>
  </section>
</main>
```

Figure 47: Display marks for student php code

The php code shown in figure 47 is what we used to get the homework score and test score of the student currently online and displaying the results in a neat and orderly format.

5. Conclusion

When looking at the final product, the website was a success and could do all the functions that was set out for us to do except to add or remove new courses. To add such a function into the website would not be too difficult, however we ran out of time. The website can still be improved on greatly and all the finer details can be sorted out. Html, php and css language were very effective languages to program with and we learned a lot when working with them.

We had never before programed with either of these languages. The setup of the database was a success and the EER diagram really help to figure out how all the different tables in the database will connect to one another.

We had two main pages, a student and admin and both of those connected to each of their subpages. When a user logged in a session was started which allowed the user only to get access to their allowed pages and information meant to be displayed for them.

The admin could do all his/her tasks except for the course creation. The student could do all the tasks we were provided with to create and design for them.

The website was designed in such a manner that it could be used on any device platform. The input boxes, buttons, header, main section and footer all adjust to the page when the device's screen is small or big.

6. References

[1] R. Clark, "HTML5 Restet Stylesheet," p. 1, 2009.