Module ICT3715

INFORMATION AND COMMUNICATION TECHNOLOGY PROJECT

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STUDENT NUMBER (Student completes) | | | | | | | | | |
| 5 | 1 | 1 | 5 |  | 2 | 1 | 7 |  | 7 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IDENTITY NUMBER (Student completes) | | | | | | | | | | | | |
| 7 | 9 | 0 | 9 | 2 | 3 | 5 | 5 | 3 | 5 | 0 | 8 | 0 |

No handwritten assignments will be accepted.

**INSTRUCTIONS:**

Complete this Front Page (page 1)

Complete the Plagiarism Pledge (page 2). Your assignment will not be assessed without this.

After you have completed the front page with your information, the plagiarism pledge, and Assignment 3 with Section A and B, save the document as a PDF document.

Keep a copy of the original should there be problem with the upload.

PLAGIARISM PLEDGE BY THE STUDENT

1. I have read Unisa’s plagiarism policy.
2. I understand Unisa’s plagiarism policy.
3. I agree to abide by Unisa’s plagiarism policy.
4. I have read the direct copying, plagiarism, and “patch-writing” document.
5. I understand what direct copying, plagiarism, and “patch-writing” is.
6. I undertake to avoid copying directly, plagiarism and patch writing.
7. All academic work, written or otherwise, that I submit is expected to be the result of my own skill and labour.
8. I understand that, if I am guilty of the infringement of breach of copyright/plagiarism or unethical practice, I will be subject to the applicable disciplinary code as determined by Unisa.
9. The marker has the right to refuse to assess the assignment and the system if plagiarism is detected.

Student name and Surname: Tshililo Theophelus Makhado

Student number: 51152177

Student signature: TTM Date: 10/02/2023

**Assignment 3 [848069]**

|  |  |
| --- | --- |
| **Due date Monday 8 August 2022, 11:00 PM** |  |

**Notes:**

This is a compulsory assignment. The assignment contributes to 40% of your year mark.

**System | Online examination file submission system**

Due to the pandemic, the University of South Africa (UNISA) has decided to **conduct all examinations in an online environment.**

The **purpose of this project** is then to look at the digital environment and **design and develop a simulation of such a system, using the real-life scenario and environment.**

You thus must design and develop an online examination file submission system, that can be presented to the University to be used as an alternative to the current system.

**Note**: You are not allowed to develop any other system or use any other data that was not prescribed or provided to you.

The outcome of this assignment will form part of the design, development, and implementation

of the database and the system. Your implementation effort will be greatly reduced if you take care with the preparation phases of the system.

**Instructions:**

* Make sure that you did complete the instructions on page 1 of this document
* Complete the header and footer with your own information
* Add your practical system content to the document
* Remove everything that is in brackets []
* Keep the answers that you had for the Assignment 1 and 2 also in this document
* Make sure that your Table of Content is updated
* Save the document as PDF, e.g., 12345678\_ICT3715\_03.pdf, (replace 1234568 with your student number)
* When you are done submit via myModules 2022 on the Module Site under Assignment 3

Table of Contents

1 Assignment 3 7

1.1 Section A = MIS Reports [20] 7

1.1.1 Create the Daily MIS report (5) 8

1.1.2 Create the Weekly MIS report (5) 8

1.1.3 Create one other MIS report (5) 8

1.1.4 Create one other MIS report (5) 9

1.2 Section B = Graphical User Interfaces [20] 9

1.2.1 Student login GUI (7) 9

1.2.2 Lecturer login GUI (7) 10

1.2.3 Exam Department (Staff member) login GUI (6) 10

2 Assignment 2 11

2.1 Section A = Database [20] 11

2.1.1 Create the database for your system and importing the data from your .csv file (20) 11

2.2 Section B = Backup and Recovery for the Database and Programming code [10] 11

2.2.1 Database (5) 12

2.2.2 Programming code & Portfolio (assignments) (20) 12

3 Assignment 1 13

3.1 Section A [4] 13

3.1.1 Programming Languages (2) 13

3.1.2 Database (2) 13

3.2 Section B [10] 13

3.2.1 Cleaning the data 13

3.3 Section C [10] 13

3.3.1 Activity Diagram (7) 13

3.3.2 ERD Diagram (3) 14

3.4 Section Backup and Recovery for the Database and Programming code [6] 14

3.4.1 Backup and Recovery Software for the Database (3) 14

3.4.2 Backup and Recovery process for the Programming code and your Portfolio (assignments) (3) 14

# Assignment 3

## Section A = MIS Reports [20]

[Use Task 2 Document – Preparing for the MIS Reports to complete this section of Assignment 3.



Also refer to “19. 10. System | MIS Reporting: Dashboard” on the myModules 2022 Module Site.

|  |
| --- |
| For each of these MIS reports you have to do the following:   * In your own words, describe the reason for this MIS report. * Create the query (the SQL).   The query should be specific and the names of the tables that you use, columns, joints (PKs, FKs) etc. should be clear and precise.   * Run the query to get an output. * You need to provide the marker with each of these queries as well as screen dumps of the output (both the actual data and should you use graphics a screen dump of the graphic as well).   Again, remember, a report is NOT a looooong list of data. |

### Create the Daily MIS report (5)

Reason

To provide the Exam Department with a list of all modules with focus on the module code, module description and date of exam this information helps to identify when a particular module is being written.

Query

$stmt = $conn->prepare("SELECT examsetup.ModuleCode, ModDesc, DateExam

FROM examsetup, moduleinfo

WHERE examsetup.ModuleCode = moduleinfo.ModCode

ORDER BY ModuleCode");

$stmt->execute();

$modules = $stmt->fetchAll();

Screen dumps

Table

Description automatically generated with medium confidence

### Create the Weekly MIS report (5)

Reason

Display all lecturer’s info for the exam department and links the modules that they are leading.

Query

$stmt = $conn->prepare("SELECT Distinct(moduleleader.ModuleCode), ModDesc, name, email, DateExam

FROM moduleleader, moduleinfo, staffinfo, examsetup

WHERE moduleleader.ModuleCode = moduleinfo.ModCode

AND examsetup.ModuleCode = moduleinfo.ModCode

AND moduleleader.StaffNumber = staffinfo.StaffNumber

ORDER BY ModuleCode");

$stmt->execute();

$modules = $stmt->fetchAll();

Screen dumps

Table

Description automatically generated

### Create one other MIS report (5)

Reason

3D Bar graph of modules with most registered student

Query

const dataSource = {

chart: {

caption: "Modules with Most registered students",

subcaption: "(With Group Values)",

numbersuffix: " Students",

yaxisname: "No of registered (students)",

theme: "candy",

plottooltext: "No of $label in $seriesName is <b>$dataValue</b>"

},

FusionCharts.ready(function() {

var myChart = new FusionCharts({

type: "errorbar2d",

renderAt: "chart-container",

width: "100%",

height: '700',

dataFormat: "json",

dataSource

}).render();

});

Screen dumps

Chart, box and whisker chart

Description automatically generated

### Create one other MIS report (5)

Reason

Bar graph with top module submissions

Query

//STEP 3 - Chart Configurations

const chartConfig = {

type: 'column2d',

renderAt: 'chart-container',

width: '100%',

height: '700',

dataFormat: 'json',

dataSource: {

// Chart Configuration

"chart": {

"caption": "Top Module Submissions",

"xAxisName": "Module",

"yAxisName": "File Uploads",

"numberSuffix": "",

"theme": "candy",

},

// Chart Data

"data": chartData

}

};

FusionCharts.ready(function(){

var fusioncharts = new FusionCharts(chartConfig);

fusioncharts.render();

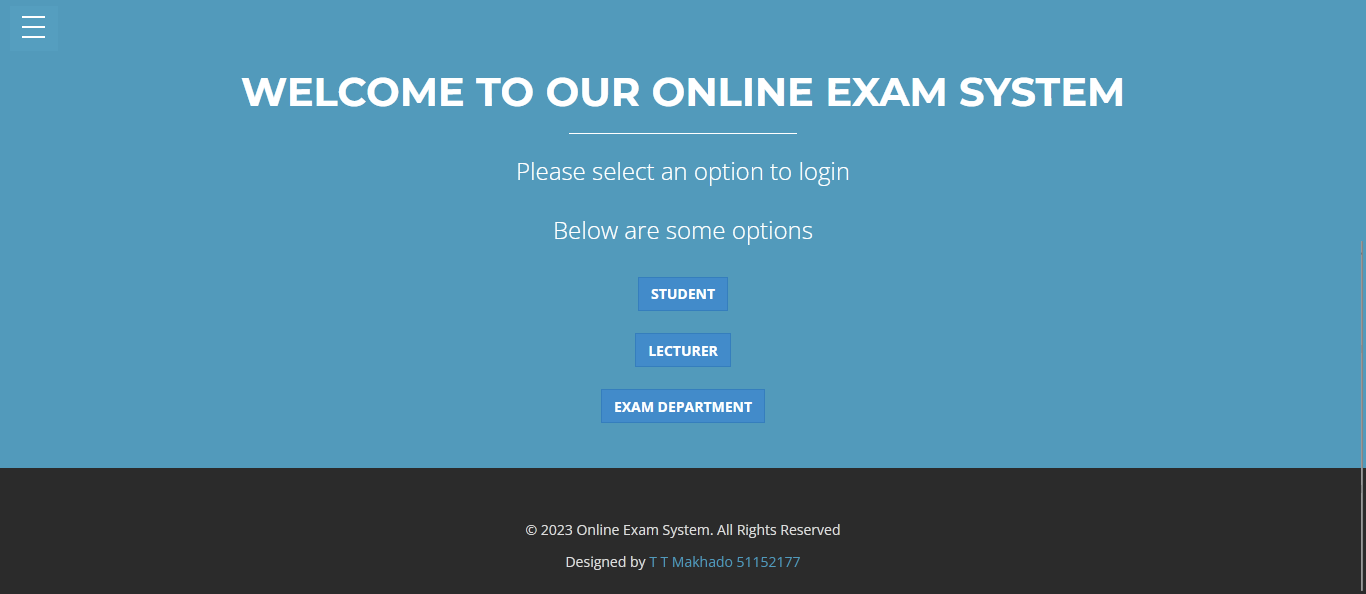
});

Screen dumps

Chart, bar chart

Description automatically generated

## Section B = Graphical User Interfaces [20]



### Student login GUI (7)

**Creating the Graphical User Interfaces for the Student Login**

Screen dumps

***Graphical user interface, website

Description automatically generated***

Code

session\_start();

require\_once 'include/conn.php';

if(ISSET($\_POST['submit'])){

if($\_POST['username'] != "" || $\_POST['password'] != ""){

$email = $\_POST['username'];

$password = $\_POST['password'];

$sql = "SELECT \* FROM `studentinfo` WHERE `email`=? AND `password`=? ";

$query = $conn->prepare($sql);

$query->execute(array($email,$password));

$row = $query->rowCount();

$fetch = $query->fetch();

if($row > 0) {

$\_SESSION['email'] = $fetch['email'];

$\_SESSION['studentnumber'] = $fetch['studentnumber'];

$\_SESSION['studentname'] = $fetch['studentname'];

header("location: stud");

} else{

echo "<script>alert('Invalid username or password')</script>";

}

}

}

### Lecturer login GUI (7)

**Creating the Graphical User Interfaces for the Lecturer Login**

Screen dumps

*Graphical user interface, website

Description automatically generated*

Code

session\_start();

require\_once 'include/conn.php';

if(ISSET($\_POST['submit'])){

if($\_POST['username'] != "" || $\_POST['password'] != ""){

$email = $\_POST['username'];

$password = $\_POST['password'];

$sql = "SELECT \* FROM `staffinfo` WHERE `email`=? AND `password`=? ";

$query = $conn->prepare($sql);

$query->execute(array($email,$password));

$row = $query->rowCount();

$fetch = $query->fetch();

if($row > 0) {

$\_SESSION['staffnumber'] = $fetch['staffnumber'];

$\_SESSION['name'] = $fetch['name'];

$\_SESSION['email'] = $fetch['email'];

header("location: lect");

} else{

echo "<script>alert('Invalid username or password')</script>";

}

}

}

### Exam Department (Staff member) login GUI (6)

**Creating the Graphical User Interfaces for the Exam Department (Staff member) Login**

Screen dumps

*Graphical user interface, website

Description automatically generated*

Code

session\_start();

require\_once 'include/conn.php';

if(ISSET($\_POST['submit'])){

if($\_POST['username'] != "" || $\_POST['password'] != ""){

$email = $\_POST['username'];

$password = $\_POST['password'];

$sql = "SELECT \* FROM `examdeptstaff` WHERE `email`=? AND `password`=? ";

$query = $conn->prepare($sql);

$query->execute(array($email,$password));

$row = $query->rowCount();

$fetch = $query->fetch();

if($row > 0) {

$\_SESSION['staffnum'] = $fetch['staffnum'];

$\_SESSION['staffname'] = $fetch['staffname'];

$\_SESSION['email'] = $fetch['email'];

header("location: exam\_dept");

} else{

echo "<script>alert('Invalid username or password')</script>";

}

}

}

1. **Assignment 2**
   1. **Section A = Database [20]**
      1. **Create the database for your system and importing the data from your .csv file (20)**

**Process**

* I started by opening the Data.csv file which was given to us
* I had to break down an analysis each table and columns needed per table
* I also analysed the data type and range for each of the columns and paid attention which are PK and FK
* I had to copy the columns needed per table on a separate worksheet which I then saved as separate csv file which represents each table
* Then using PHPMyAdmin I could easily create the database, tables and columns using the simple interface. The interface also allows to specify PK, FK and datatypes.
* Then I had to import the csv files to PHPMyAdmin to make sure the data was properly inserted.
* After this process then I extract an “SQL” file

Graphical user interface, application, table

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

**Problems**

The problems encountered where many I will break it down table by table and provide extra info

“studentmodule table”

* There was a lot of duplicate data of which I dealt with by deleting them in the csv file
* Making the 2 columns primary was simple as the interface provided that option
* Making the “ModuleCode” columns foreign key was hard, but I ran an SQL query to deal with that.
* ALTER TABLE `studentmodule` ADD PRIMARY KEY(`StudentNumber`, `ModuleCode`);

“studentinfo table”

* The only issue in this table was to create an email column and collect data from the studentnumber and add text. That issue was sort using the SQL query below
* UPDATE studentinfo SET email = StudentNumber + "@mylife.unisa.ac.za"

“staffinfo table”

* There was a lot of duplicate data of which I dealt with by deleting them in the csv file

“moduleleader table”

* There was a lot of duplicate data of which I dealt with by deleting them in the csv file
* Making the 2 columns primary was simple as the interface provided that option

“moduleinfo table”

* There was a lot of duplicate data of which I dealt with by deleting them in the csv file

“examsetup table”

* There was a lot of duplicate data of which I dealt with by deleting them in the csv file
* I also paid attention to the DATE datatype from the csv file

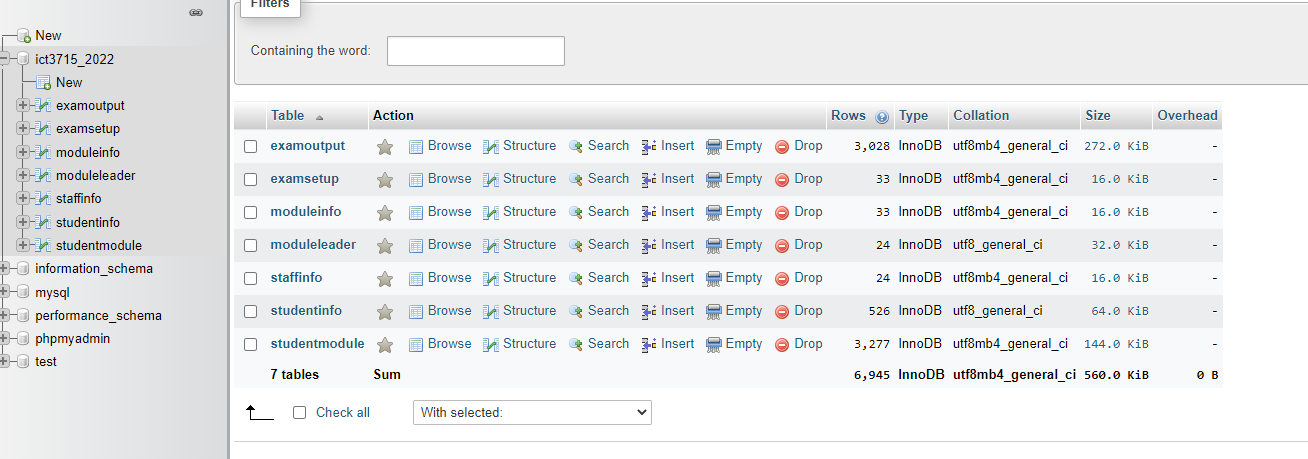
“examoutput table”

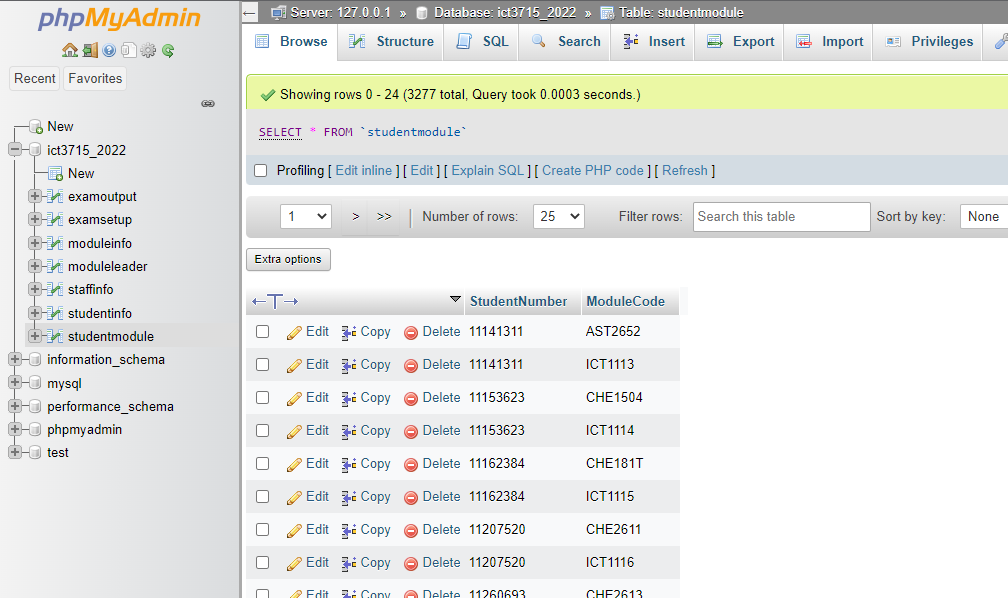
* There was a lot of duplicate data of which I dealt with by deleting them in the csv file
* I also paid attention to the DATE datatype from the csv file

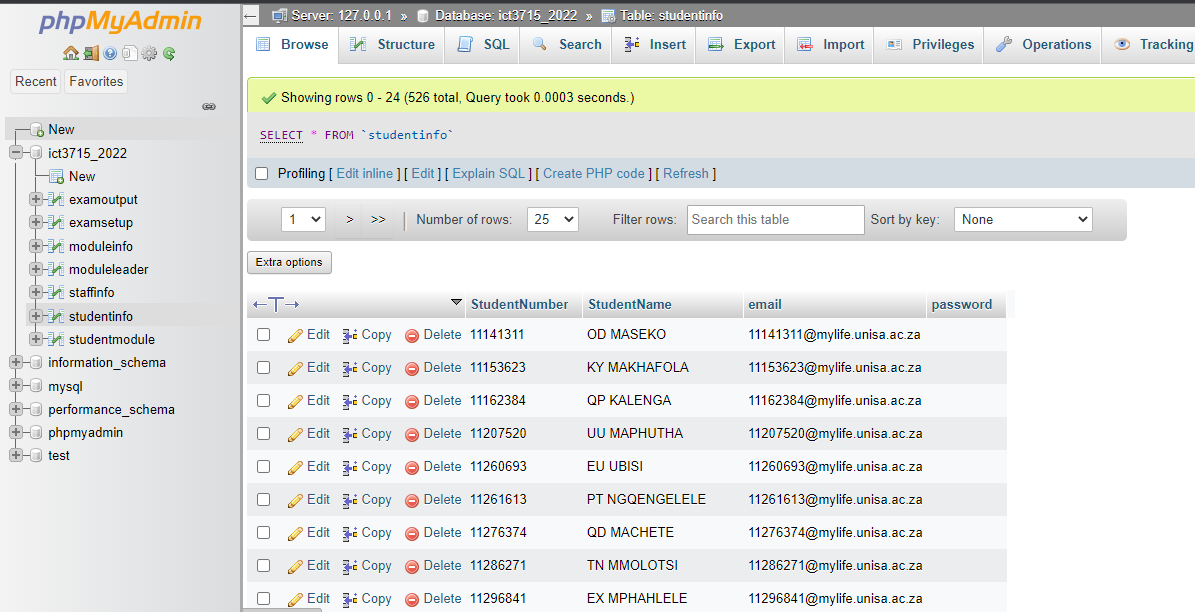
Extra

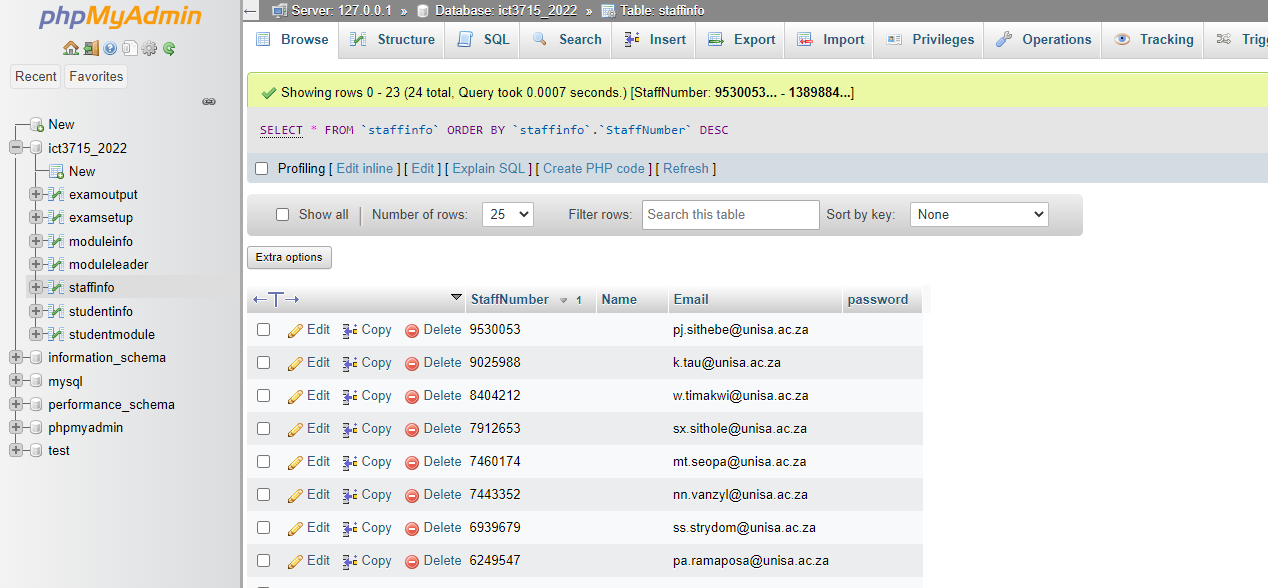
* Just paying attention to certain datatypes such as date and time; I had to set them from the csv file so when it is imported it uses the right datatype.

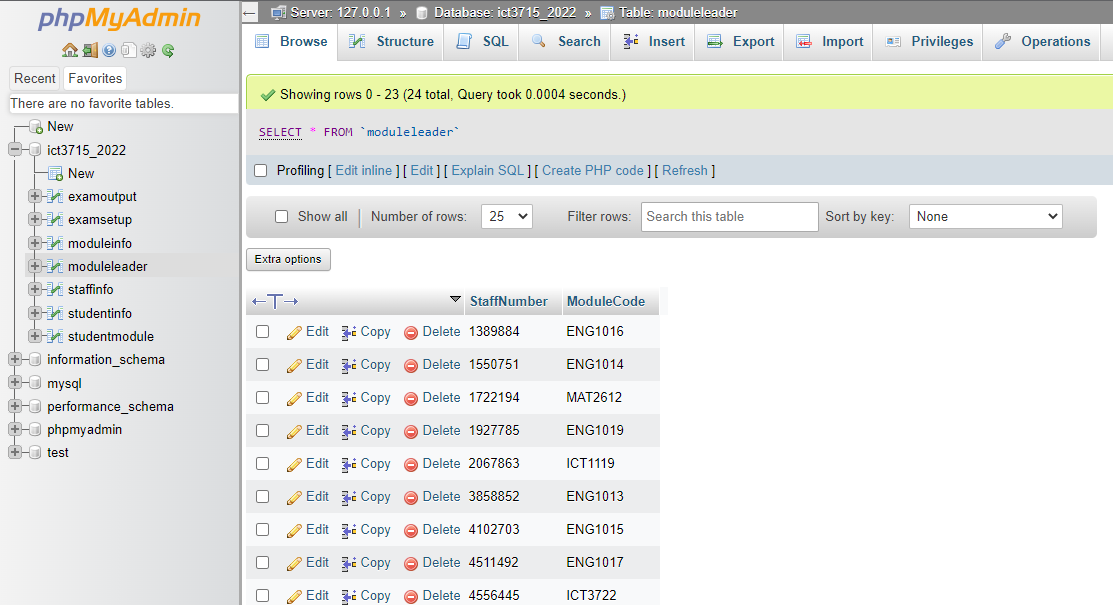
**Screenshots**

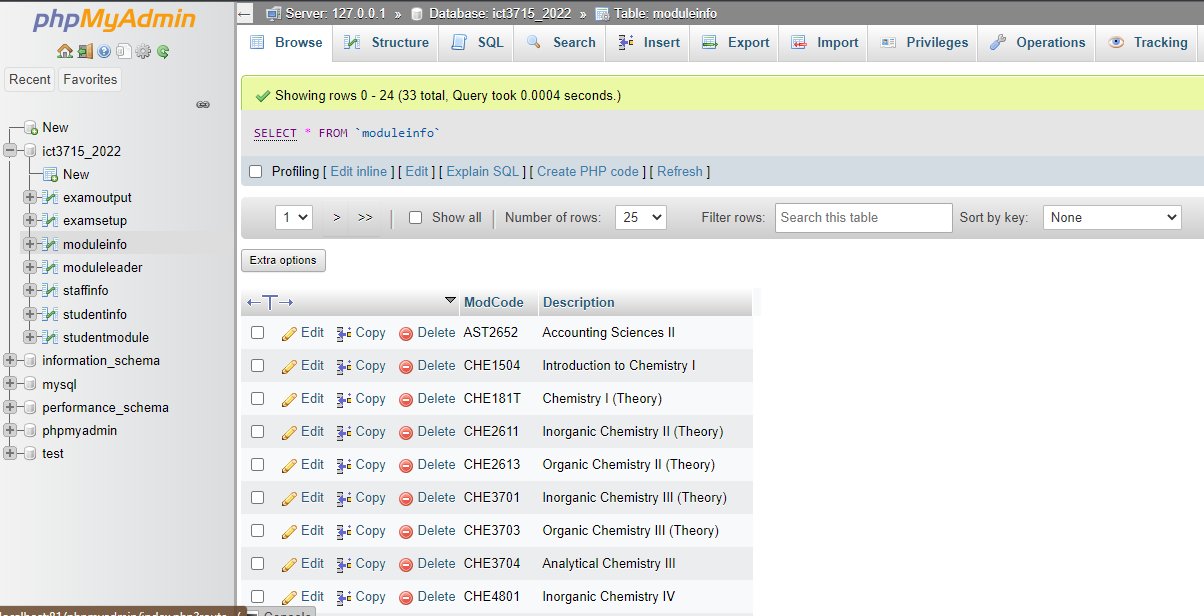


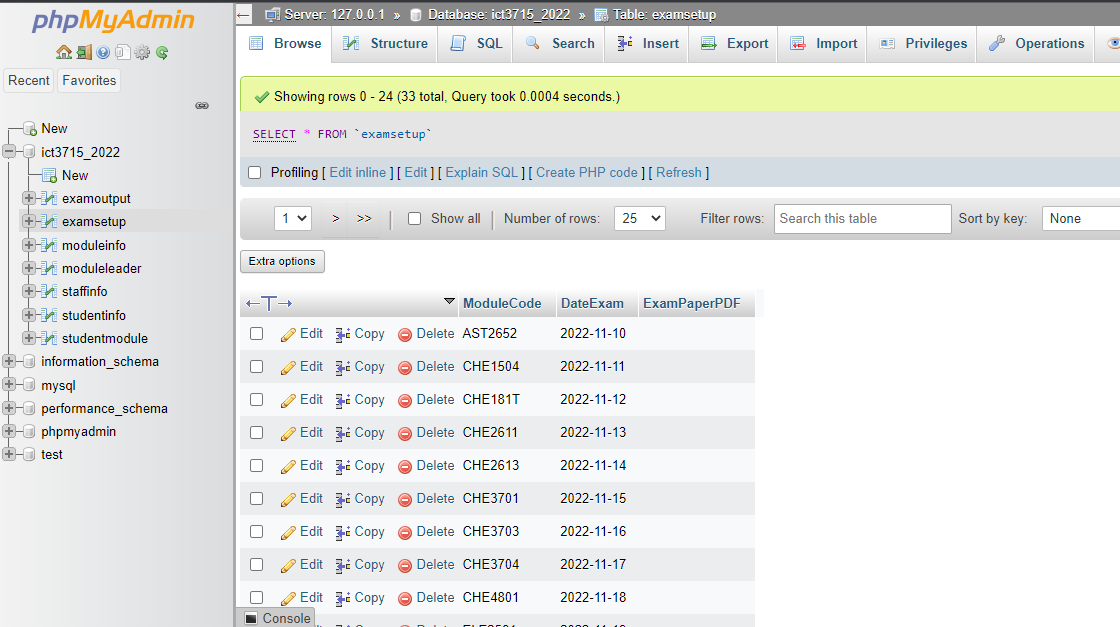


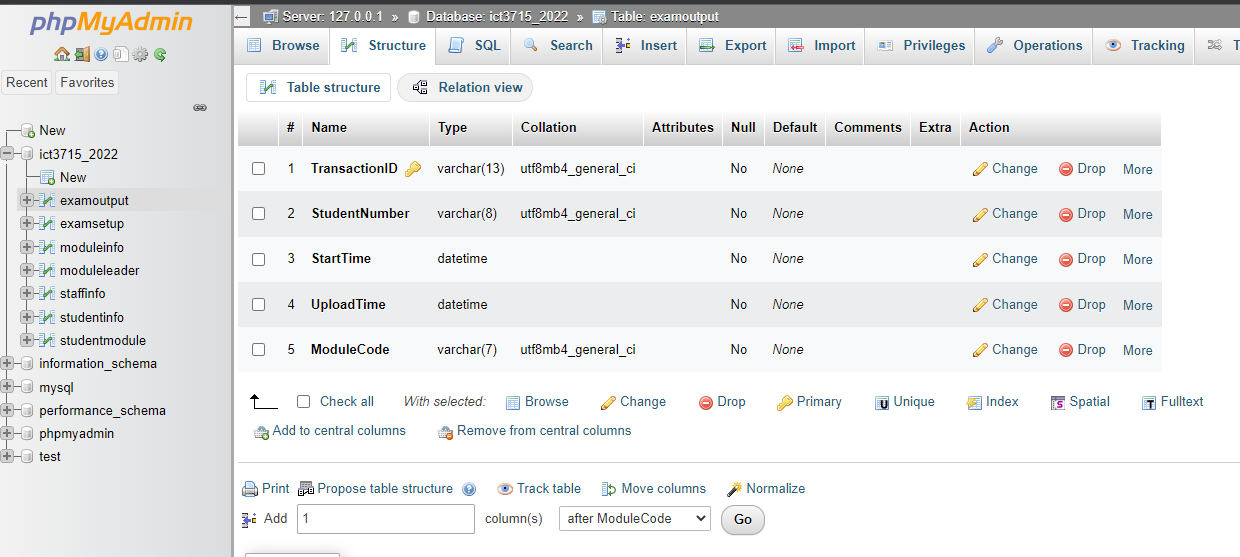








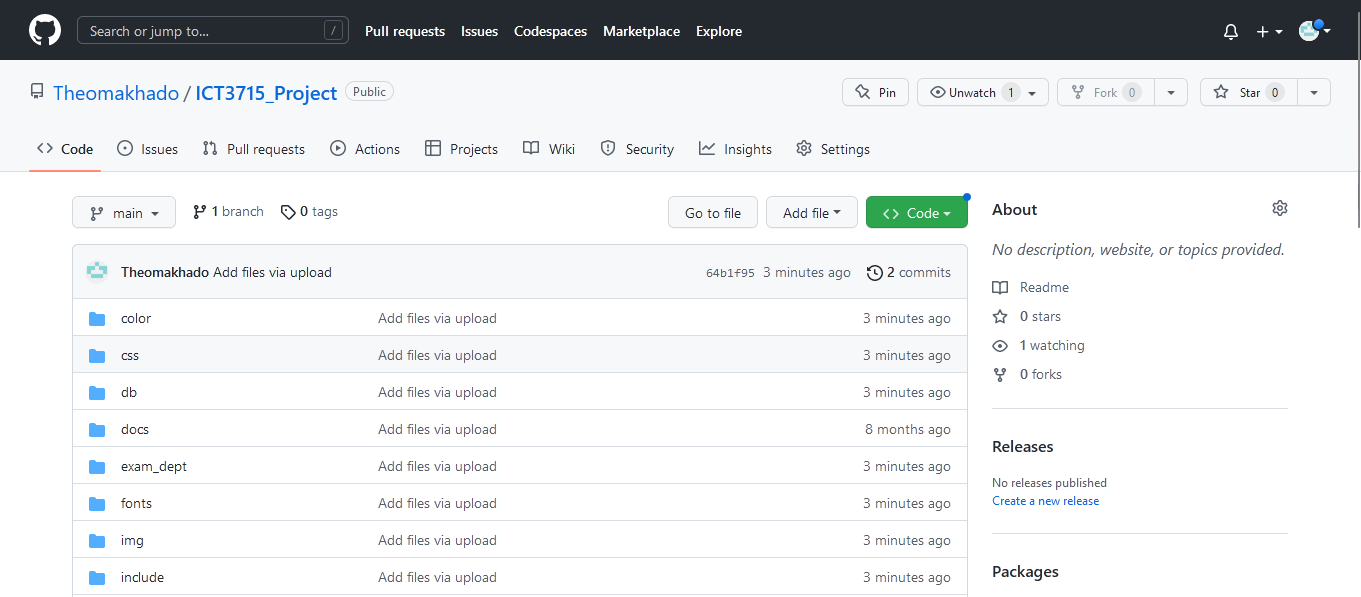




* 1. **Section B = Backup and Recovery for the Database and Programming code [10]**

Graphical user interface

Description automatically generated with medium confidence



* + 1. **Database (5)**

1. I start by opening my xampp software and running PHPMyAdmin using the localhost
2. Then extract the sql and csv file from the database and it adds it to the downloads folder
3. Then this get uploaded on a weekly basis to my GitHub accounts as well as to Google Drive for extra backup
4. GitHub being a source control software I can commit changes and go back to previous changes. Basically, it allows me to track changes made to the database file
5. 000webhost will also be used to upload the database from the local server to a cloud-based server.
6. This is done weekly to keep track of changes and make sure I have no database crashes.
   * 1. **Programming code & Portfolio (assignments) (20)**

--

-- Database: `ict3715\_2022`

--

-- --------------------------------------------------------

--

-- Table structure for table `examoutput`

--

**CREATE** **TABLE** `examoutput` **(**

`TransactionID` varchar**(**13**)** **NOT** **NULL,**

`StudentNumber` varchar**(**8**)** **NOT** **NULL,**

`StartTime` datetime **NOT** **NULL,**

`UploadTime` datetime **NOT** **NULL,**

`ModuleCode` varchar**(**7**)** **NOT** **NULL**

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

--

-- Table structure for table `examsetup`

--

**CREATE** **TABLE** `examsetup` **(**

`ModuleCode` varchar**(**7**)** **NOT** **NULL,**

`DateExam` date **NOT** **NULL,**

`ExamPaperPDF` varchar**(**150**)** **NOT** **NULL**

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

--

-- Table structure for table `moduleinfo`

--

**CREATE** **TABLE** `moduleinfo` **(**

`ModCode` varchar**(**7**)** **NOT** **NULL,**

`Description` varchar**(**100**)** **NOT** **NULL**

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

--

-- Table structure for table `moduleleader`

--

**CREATE** **TABLE** `moduleleader` **(**

`StaffNumber` varchar**(**7**)** **NOT** **NULL,**

`ModuleCode` varchar**(**7**)** **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf8**;**

--

-- Table structure for table `staffinfo`

--

**CREATE** **TABLE** `staffinfo` **(**

`StaffNumber` varchar**(**7**)** **NOT** **NULL,**

`Name` varchar**(**150**)** **NOT** **NULL,**

`Email` varchar**(**200**)** **NOT** **NULL,**

`password` text **NOT** **NULL**

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

**CREATE** **TABLE** `studentinfo` **(**

`StudentNumber` varchar**(**8**)** **NOT** **NULL,**

`StudentName` varchar**(**100**)** **DEFAULT** **NULL,**

`email` varchar**(**150**)** **DEFAULT** **NULL,**

`password` varchar**(**200**)** **DEFAULT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf8**;**

--

**CREATE** **TABLE** `studentmodule` **(**

`StudentNumber` varchar**(**8**)** **NOT** **NULL,**

`ModuleCode` varchar**(**7**)** **NOT** **NULL**

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

--

--

-- Indexes for dumped tables

--

--

-- Indexes for table `examoutput`

--

**ALTER** **TABLE** `examoutput`

**ADD** **PRIMARY** **KEY** **(**`TransactionID`**);**

--

-- Indexes for table `examsetup`

--

**ALTER** **TABLE** `examsetup`

**ADD** **PRIMARY** **KEY** **(**`ModuleCode`**);**

--

-- Indexes for table `moduleinfo`

--

**ALTER** **TABLE** `moduleinfo`

**ADD** **PRIMARY** **KEY** **(**`ModCode`**);**

--

-- Indexes for table `moduleleader`

--

**ALTER** **TABLE** `moduleleader`

**ADD** **PRIMARY** **KEY** **(**`StaffNumber`**,**`ModuleCode`**),**

**ADD** **UNIQUE** **KEY** `ModuleCode` **(**`ModuleCode`**);**

--

-- Indexes for table `staffinfo`

--

**ALTER** **TABLE** `staffinfo`

**ADD** **PRIMARY** **KEY** **(**`StaffNumber`**);**

--

-- Indexes for table `studentinfo`

--

**ALTER** **TABLE** `studentinfo`

**ADD** **PRIMARY** **KEY** **(**`StudentNumber`**);**

--

-- Indexes for table `studentmodule`

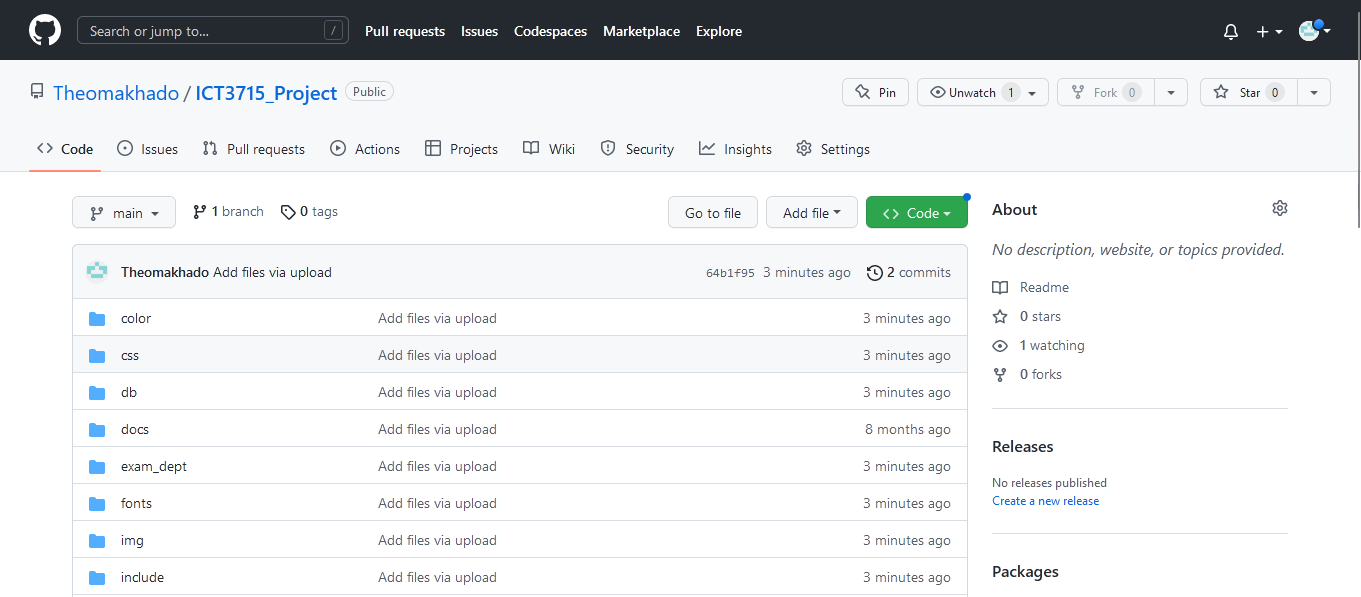
--

**ALTER** **TABLE** `studentmodule`

**ADD** **PRIMARY** **KEY** **(**`StudentNumber`**,**`ModuleCode`**);**

**COMMIT;**





Graphical user interface

Description automatically generated

1. **Assignment 1**
   1. **Section A [4]**
      1. **Programming Languages (2)**

PHP, JavaScript, jQuery, Bootstrap, Fusion Charts

* + 1. **Database (2)**

MySQL, PHPMyAdmin, MariaDB

* 1. **Section B [10]**
     1. **Cleaning the data**

A picture containing table

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated



* 1. **Section C [10]**
     1. **Activity Diagram (7)**

Diagram

Description automatically generated

* + 1. **ERD Diagram (3)**

A picture containing diagram

Description automatically generated

* 1. **Section Backup and Recovery for the Database and Programming code [6]**
     1. **Backup and Recovery Software for the Database (3)**

[GitHub, 000Webhost, mySQLWorkBench]

* + 1. **Backup and Recovery process for the Programming code and your Portfolio (assignments) (3)**

[GitHub, 000Webhost, mySQLWorkBench]

GitHub – is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.

000Webhost - Free web hosting tends to be so limited in capabilities and features that users must pay to get what they need. However, 000webhost is a free website hosting solution that provides an array of valuable features, including a website builder, WordPress support, and no ads.

mySQLWorkBench - MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system.