COM1025 Web and Database Systems

Coursework Assignment

Abdullah Zahid, 6636272 and az00379

I confirm that the submitted work is my own work. No element has been previously submitted for assessment, or where it has, it has been correctly referenced. I have clearly identified and fully acknowledged all material that is entitled to be attributed to others (whether published or unpublished) using the referencing system set out in the programme handbook.

University Accommodation Office

1 Development Environment

The drawing software I used was Draw.io. I used this to implement the tables by indicating primary keys and foreign keys where required. I also used crows foot notation, which I also added from this software.
Due to being a Mac User, I used, Mamp for the PHP related tasks. I also got the help of PHPMYADMIN which I used from MAMP. All my files were saved in 'htdocs' as a root folder. This allowed me to open files via PHP. In addition, it allowed me to implement, interpret and modify and SQL related actions.
The web browser I used was Google Chrome version 87.0.4280.141 for MAC OS.
The operating system I used was Mac OS Big Sur / Catalina.

2 EER Data Model Business Rules and Assumptions

Business Rules

1. Flat – Hall
    - 1.1. There are 2 Halls in each flat.
    - 1.2. There are 2 Flats that the university owns, each containing two halls.
    - 1.3. Resulting into 4 Halls.
    - 1.4. A Flat has the ability to add more halls, in the act, should there be a flat renovation, construction or extension.

2. Hall – Room
    - 2.1. There are 4 halls in total.
    - 2.2. Each Hall has 8 rooms.
    - 2.3. 32 Rooms in total.
    - 2.4. A Hall has the ability to add more rooms, in the act, should there be a flat/hall renovation, construction or extension.

3. Hall – Hall Inspection
    - 3.1. Hall Inspection can be done in multiple halls, though report is published on hall-by-hall basis.
    - 3.2. Hall Inspection is carried out every couple of days in each hall.
    - 3.3. Hall Inspection report should have HallID indicating where a certain Hall Inspection was carried out.

4. Staff – Hall Inspection
    - 4.1. One Staff can do many hall inspections.
    - 4.2. One Hall Inspection is only done by one staff member.
    - 4.3. Hall Inspection report should have StaffID indicating which staff did the inspection.

5. Staff – Flat
    - 5.1. Many Staff work in a flat.
    - 5.2. One flat has many staff.
    - 5.3. Staff member roles include: Receptionist, Cleaner and Security Guard, where one can also be a manager.

6. Student – Staff
    - 6.1. One Student(s) can seek advice from many staff.
    - 6.2. One Staff can help many students.

7. Student – Room
    - 7.1. One Student can only live in one room, during a semester.
    - 7.2. One Room can only occupy one person and no more than that.

8. Course – Student
    - 8.1. One student must be enrolled to at least one course but only one course at a time.
    - 8.2. A Course can have no students, or it can have many students.

9. Student – Undergraduate / Postgraduate
    - 9.1. Students are classes in to two groups: Undergraduate and Postgraduate.
    - 9.2. A Student cant be an Undergraduate and Postgraduate at the same time.
    - 9.3. A Student must be an Undergraduate or a Postgraduate whilst a student.

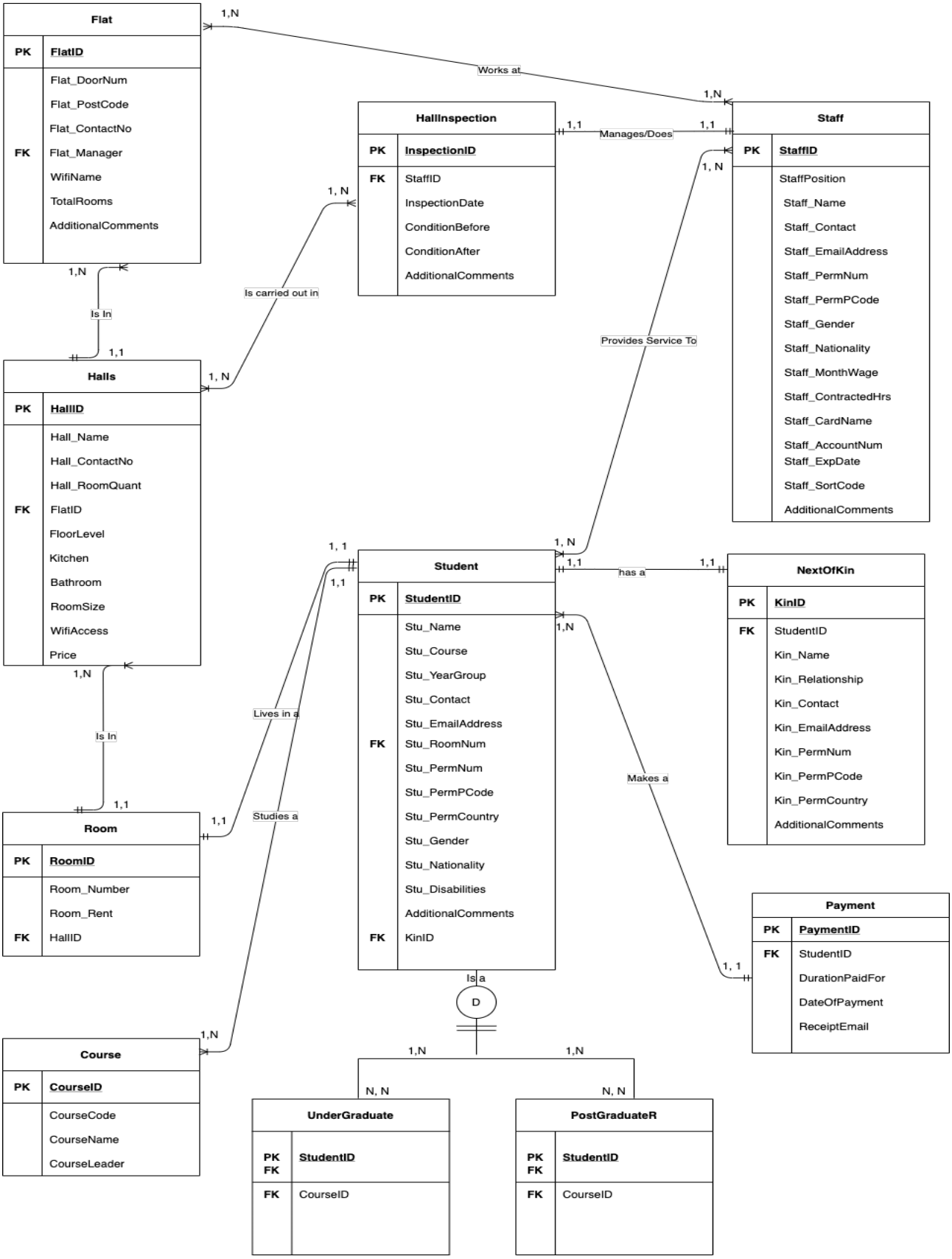
10. Student – Payment
    - 10.1. One student can only pay for only one semester at a given time.
    - 10.2. One payment can only be received by one student.

11. Student – Next of Kin
    - 11.1. One student can only have one next of kin.
    - 11.2. A next of kin must belong to a single individual.

Assumption

1. A student can have more than one email address.
2. A staff can work more than one role at once (Cleaner and security guard).
3. Hall Inspection can be carried out more than once a day or maybe no inspections a day.

3 EER Diagram



4 Logical Relational Database Schema

Database Name: **'university' (MUST BE ADDED WHEN IMPLEMENTING SQL)**
Flat(FlatID, Flat_DoorNum, Flat_PostCode, Flat_ContactNo, Flat_Manager, WifiName, TotalRooms, AdditionalComments)
PrimaryKey: FlatID
ForeignKey: Flat_Manager REFERENCES Staff(StaffID)

Halls(HallID, Hall_Name, Hall_ContactNo, Hall_RoomQuant, FlatID, FloorLevel, Kitchen, Bathroom, RoomSize, WifiAccess, DailyInspection, WeeklyInspection, Price)
PrimaryKey: HallID
ForeignKey: FlatID REFERENCES Flat(FlatID)

Room(RoomID, Room_Number, Room_Rent, HallID, FlatID)
PrimaryKey: RoomID
ForeignKey: HallID REFERENCES Halls(HallID),
FlatID REFERENCES Flat(FlatID)

HallInspection(InspectionID, StaffID, InspectionDate, ConditionBefore, ConditionAfter, AdditionalComments)
PrimaryKey: InspectionID
ForeignKey: StaffID REFERENCES Staff(StaffID)

Student(StudentID, Stu_Name, Stu_Course, Stu_YearGroup, Stu_Contact, Stu_EmailAddress, Stu_RoomNum, Stu_PermNum, Stu_PermPCode, Stu_PermCountry, Stu_Gender, Stu_Nationality, Stu_Disabilities, AdditionalComments, KinID)
PrimaryKey: StudentID
ForeignKey: Stu_RoomNumber REFERENCES Room(RoomID),
KINID REFERENCES NextOfKin(KinID)

Undergraduate(StudentID, CourseID,)
PrimaryKey: StudentID
ForeignKey: StudentID REFERENCES Student(StudentID)
ForeignKey: CourseID REFERENCES Course(CourseID)

Postgraduate (StudentID, CourseID,)
PrimaryKey: StudentID
ForeignKey: StudentID REFERENCES Student(StudentID)
ForeignKey: CourseID REFERENCES Course(CourseID)

Staff(StaffID, StaffPosition, Staff_Name, Staff_Contact, Staff_EmailAddress, Staff_PermNum, Staff_PermPCode, Staff_Gender, Staff_Nationality, Staff_MonthWage, Staff_ContractedHrs, Staff_CardName, Staff_AccountNum, Staff_ExpDate, Staff_SortCode, AdditionalComments)
PrimaryKey: StaffID
ForeignKey: None

NextofKin(KinID, StudentID, Kin_Name, Kin_Relationship, Kin_Contact, Kin_EmailAddress, Kin_PermNum, Kin_PermPCode, Kin_Country, AdditionalComments)
PrimaryKey: KinID
ForeignKey: StudentID REFERENCES Student(StudentID)

Payment(PaymentID, StudentID, Cardholder_Name, Cardholder_Number, Cardholder_ExpDate, Cardholder_,CVC, DurationPaidFor, DateOfPayment, ReceiptEmail)
PrimaryKey: PaymentID
ForeignKey: StudentID References Student(StudentID)

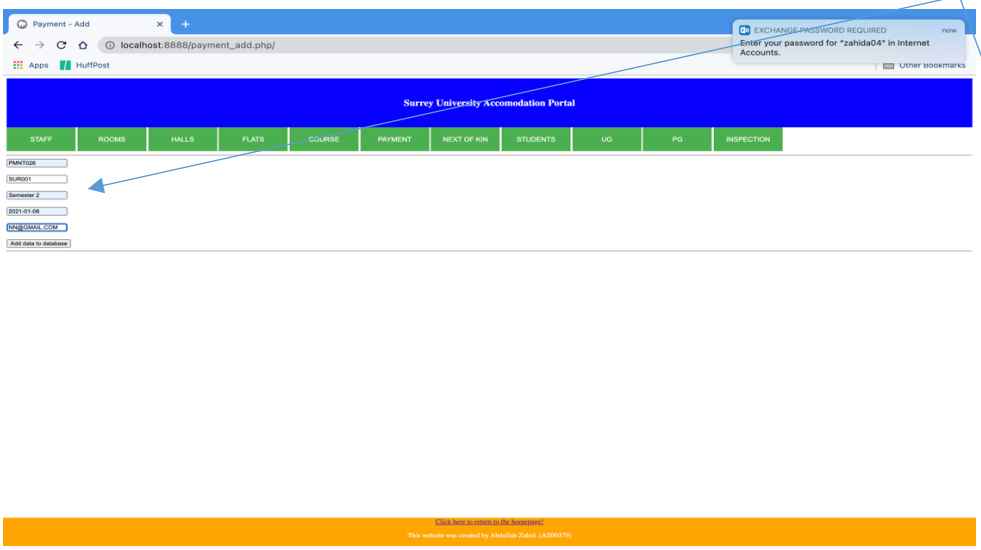
Course(CourseID, CourseCode, CourseName, CourseLeader)
PrimaryKey: CourseID
ForeignKey:None

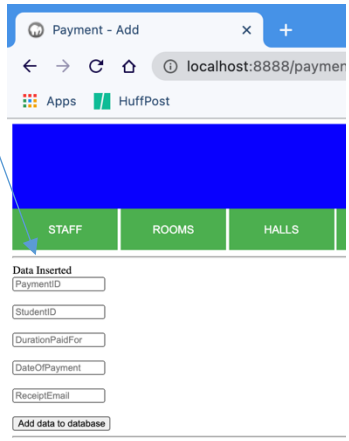
5 Website Working with MySQL Database

In this section, you should give a list of all files of your website (can be one if you decide to have a simple webpage, but remember you are not marked for front-end). Clearly state which PHP script is used to access the MySQL database and for what purposes. Include one or more screenshot of your webpage.

In my websites which I have created, we can implement the SQL database via PHP in three different ways. The ways we can implement are by **adding** an entry, viewing our database on our website and removing an **existing** entry. Please ensure that when importing COM1025_CW_AZ00379.sql (SQL Script), there should already be an **existing 'university database'** because that is the name, I used in the PHP scripts under '**dbName'**.

Here I am demonstrating the 'add' function. As you can see here, I am adding all the required variables, as shown in **screenshot 1**. When I press enter, I get a confirmation that the data has been inserted as shown in **screenshot 2**.

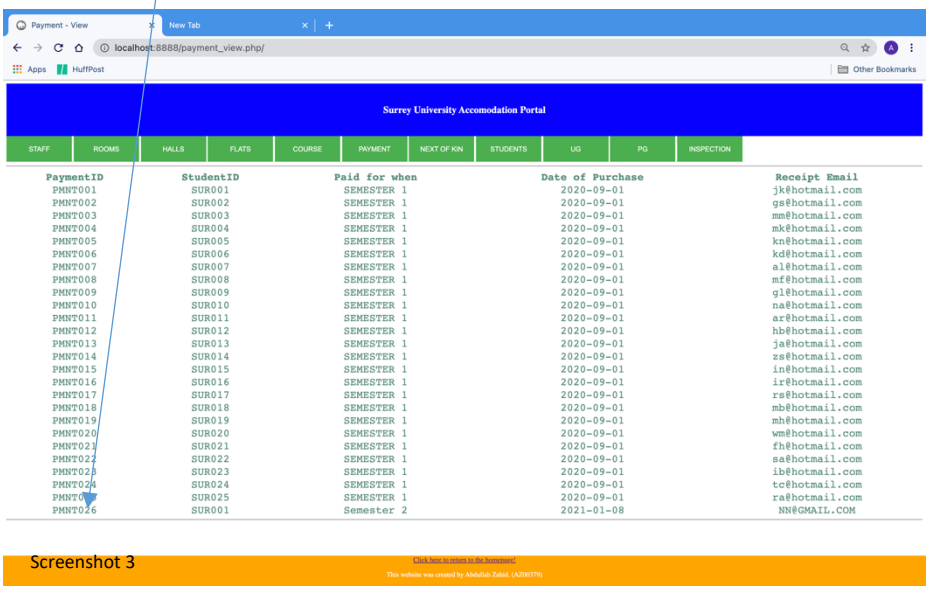




Screenshot 2

Screenshot 1

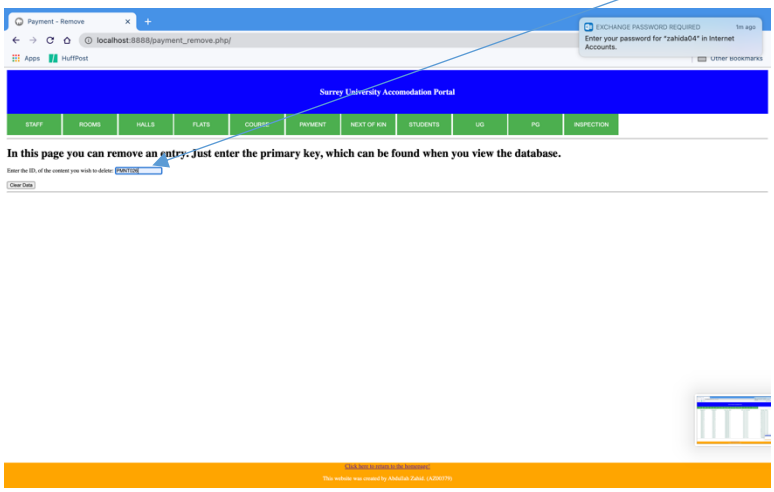
In **Screenshot 3**, we will now view what we have entered. You can see from Screenshot 1 I entered data, and it is now visible in the Payment 'view'.

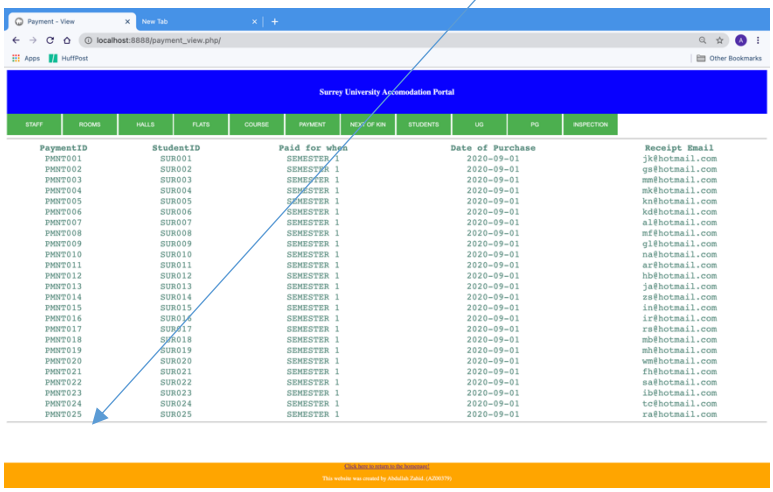


PaymentID	StudentID	Paid for when	Date of Purchase	Receipt Email
PMNT001	SUR001	SEMESTER 1	2020-09-01	jk@hotmail.com
PMNT002	SUR002	SEMESTER 1	2020-09-01	gs@hotmail.com
PMNT003	SUR003	SEMESTER 1	2020-09-01	mm@hotmail.com
PMNT004	SUR004	SEMESTER 1	2020-09-01	mk@hotmail.com
PMNT005	SUR005	SEMESTER 1	2020-09-01	kn@hotmail.com
PMNT006	SUR006	SEMESTER 1	2020-09-01	kd@hotmail.com
PMNT007	SUR007	SEMESTER 1	2020-09-01	al@hotmail.com
PMNT008	SUR008	SEMESTER 1	2020-09-01	mf@hotmail.com
PMNT009	SUR009	SEMESTER 1	2020-09-01	gl@hotmail.com
PMNT010	SUR010	SEMESTER 1	2020-09-01	na@hotmail.com
PMNT011	SUR011	SEMESTER 1	2020-09-01	ar@hotmail.com
PMNT012	SUR012	SEMESTER 1	2020-09-01	hb@hotmail.com
PMNT013	SUR013	SEMESTER 1	2020-09-01	ja@hotmail.com
PMNT014	SUR014	SEMESTER 1	2020-09-01	zs@hotmail.com
PMNT015	SUR015	SEMESTER 1	2020-09-01	in@hotmail.com
PMNT016	SUR016	SEMESTER 1	2020-09-01	ir@hotmail.com
PMNT017	SUR017	SEMESTER 1	2020-09-01	rs@hotmail.com
PMNT018	SUR018	SEMESTER 1	2020-09-01	mb@hotmail.com
PMNT019	SUR019	SEMESTER 1	2020-09-01	mh@hotmail.com
PMNT020	SUR020	SEMESTER 1	2020-09-01	wm@hotmail.com
PMNT021	SUR021	SEMESTER 1	2020-09-01	fh@hotmail.com
PMNT022	SUR022	SEMESTER 1	2020-09-01	sa@hotmail.com
PMNT023	SUR023	SEMESTER 1	2020-09-01	ib@hotmail.com
PMNT024	SUR024	SEMESTER 1	2020-09-01	tc@hotmail.com
PMNT025	SUR025	SEMESTER 1	2020-09-01	ra@hotmail.com
PMNT026	SUR001	Semester 2	2021-01-08	NN@GMAIL.COM

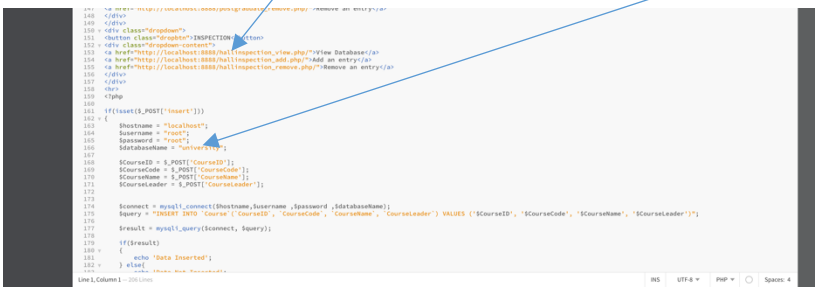
Screenshot 3

In **Screenshot 4**, I will demonstrate how to remove this certain entry. All I have had to do was enter the primary key of the entry I wanted to remove. In this instance 'I wanted to remove 'PMNT026', when I click the button a message will pop up that the data has been removed. **Screenshot 5** will demonstrate that when I go to view the page, the entry has successfully been removed.





In Screenshot 6, I will show how my anchors and hyperlinks work, and I will show the PHP script from where I insisted a 'university' database must be implemented before importing the SQL script. As you can see through arrow indication, my anchors are displayed in a format "http://localhost:8888/...", this is because for PHP to work, they need to be saved in the **'root'** folder, and my php files have to be saved under here, to be opened in a PHP suitable format. In addition, you can see here, on the PHP scripts, I entered a 'university' as database name. The database must match for the PHP to work.



6 Advanced Tasks

Advanced Tasks were not attempted

7 References

https://www.w3schools.com/	I used this website for the design of my website. I didn't apply any JavaScript as it was surplus to my requirement, but I used this website for the CSS part of my website. The CSS code I applied was modified from the content on the website to how I wanted the CSS to display.
https://www.w3schools.com/php/php_mysql_intro.asp	I used this to gain further knowledge about how PHP and SQL work together. I didn't use anything from which I applied to my website, however I used it to learn how both programming languages work together.
https://www.codecademy.com/learn/learn-sql	I used this website to learn more about SQL and how the structures work. I.e. How to display primary and foreign keys
https://www.youtube.com/watch?v=Em1Oneakz1A	This website is a YouTube video, which I would like to use as a reference. This website helped me understand how to insert data into my website. This website was very helpful in helping me design and write the code to insert any data from HTML form, which could then transfer to my SQL database, which I can then display onto my website.
https://www.youtube.com/watch?v=bHFoobciCTM	This website is another YouTube video, which I would like to use as a reference. This website helped me understand how to view data on my website. This website was very helpful in helping me design and write the code to view and read data from MySQL to my website which I created and could then display onto my website.
https://www.youtube.com/watch?v=08fYqZqtaeU	This website is another YouTube video, which I would like to use as a reference. This website helped me understand how to remove data from my website. This website was very helpful in helping me design and write the code to remove any data via a HTML form, where I enter the respective Primary key for each certain database, and I can remove data.