Computer Science and Creative Technologies

Coursework or Assessment Specification

Module Details

Module Code	UFCFES-30-1
Module Title	Web Development and Databases
Module Leader	Zaheer Khan
Module Tutors	Zaheer Khan, Barkha Javed, Shelan Jeawak
Year	2021-22
Component/Element Number	A
Total number of assessments for this module	1
Weighting	100%
Element Description	Project: Design and development of a Website

Dates

Date issued to students	11 October 2021
Submission Date	5 May 2022
Submission Place	Blackboard
Submission Time	14:00
Date to be returned to students	June 2022
Submission Notes	Each student is required to submit project deliverables via Blackboard.

Feedback

Feedback provision will be	Summative written feedback via blackboard. Formative verbal feedback during practical/seminar session;

Contents

Section 1: Overview of Assessment

In addition to generally supporting your learning on this module, this project work assesses the following module learning outcomes (taken from the module specification):

- *Demonstrate the ability to select and use web development techniques and concepts to develop dynamic and responsive websites (MO1)*
- *Design and develop static web sites to solve simple problems (MO2)*
- *Identify and assess web security issues in a website (MO3)*
- *Demonstrate a basic understanding of legal, ethical, social and professional requirements when designing a web application (MO4)*
- *Design and develop data management solutions for a web application (MO5)*

The assessment is designed to allow you to use your learning and play to your strengths. You can always seek advice and/or interim verbal feedback from your tutors during practical sessions.

Each student is expected to submit full website. You should submit all source code (e.g. website project folder from htdocs folder or /var/www folder or flask app folder) plus relevant Database dump. You are also expected to provide a project presentation and project demo recorded video (i.e. screen recording with audio). In this video you'll talk through your Powerpoint slides to explain various aspects of your project and demonstrate functionality/features of your website by running the website in a web browser. The video will help us to see it is your work. Also, with the help of the video you can get your point(s) across and this can help you to gain good marks. Please be concise and provide examples in your presentation and project demo. Please be mindful of selecting appropriate working scenarios for the demonstration.

You may use any editor of your choice to write the code but you are not allowed to use web development tools such as WYSIWYG editors, without express permission of the module leader. You are allowed to use HTML, CSS, JavaScript, MySQL and Python Flask. You are allowed to use a front-end framework at your own risk, but for back-end Python Flask must be used. All the Python programs must use version Python35 or above. **Please discuss with your tutors if you would like to use any other programming/scripting language or web framework.** If needed, your tutors may schedule a meeting for demo and ask questions about your project. Please see deliverables section for more details.

The software requirements for the assignment are described in more detail in **section 2**.
The assessment is worth **100%** of the overall mark for the module. **Each student must upload deliverables including code, DB dump, slides, test data, demo and presentation video by the submission deadline to claim any marks.**

Working on this assignment will help you to develop effective and systematic web development, web and data security, data management and programming skills. **Your tutor will organise one online session to answer your questions about the project and this will be your opportunity to ask questions and clarify requirements**. **In addition, as part of**

your weekly practical sessions one exercise will be to discuss how the topic covered in the practical can be applied on your website project. If you have more questions about this assessment then post them to the discussion board on Blackboard. You can find the discussion board from the front page of the module Blackboard pages and use the forum under the title "**Website project**".

Section 2: Task Specification

Case Study: Horizon Hotels (HH)

The HH is a successful chain of hotels across UK cities and is very popular among tourists. They have specific requirements and constraints as detailed below. Table 1 shows cities where customers can book hotel rooms, hotel capacity and room prices for peak and off-peak seasons. Peak season rates are applicable only for months April to September (inclusive).

Table 1: Cities, capacity and rates

Cities	**Capacity i.e. # of Rooms**	**Rate (GBP) for Standard Room per night**	
		Peak Season (Apr - Sept)	**Off-peak Season**
Aberdeen	80	140	60
Belfast	80	130	60
Birmingham	90	150	70
Bristol	90	140	70
Cardiff	80	120	60
Edinburgh	90	160	70
Glasgow	100	150	70
London	120	200	80
Manchester	110	180	80
New Castle	80	100	60
Norwich	80	100	60
Nottingham	100	120	70
Oxford	80	180	70
Plymouth	80	180	50
Swansea	80	120	50

HH is looking for an IT solution for their room booking with the objective to make it convenient for customers to book rooms online as well as generate different reports. They have identified the following requirements.

Requirements:

1. Booking process is easy to follow and generates a booking receipt after payment. Booking can be made up to 3 months in advance. Booking a room in advance will provide discount on the total booking price as per the following table:

Number of days in advance booking	**Discount % on total booking price**
Between 80 and 90 days	20%

Between 60 and 79 days	10%
Between 45 and 59 days	5%
Under 45 days	No discount

2. Booking cancellation before 60 days of booking date does not incur cancellation charges. Booking cancellation between 30 and 60 days of booking date will incur charges up to 50% of booking price. Within 30 days of booking date 100% of booking price will be charged.

3. Each hotel has 3 types of rooms: 1. Standard room; 2. Double room; 3. Family room. Each hotel has 30% standard room; 50% double rooms and 20% Family rooms. A standard room can have 1 guest only. A double room is 20% more price of a Standard room and can have 2 guests max. For second guest extra 10% of a Standard room price will also be charged. Family room is 50% more price of a Standard room and can host a family of maximum 6 persons.

4. All hotels accept payment in the following currencies: Euros, USD; Currency conversions should be at the following rates: 1 GBP = 1.2 Euros; 1 GBP = 1.6 USD;

5. Consider other suitable non-functional requirements necessary for the HH online booking system. Be creative.

More specifically there are following tasks and %age of marks allocation:

Task 1: 20% - Normalised database: You should design and implement database of the selected case study in 3rd Normal Form.

Task 2: 15% - Responsive web design: Your website front-end should have nice look and feel and must be responsive for different screen-sizes.

Task 3: 15% User system and security: Your website should have a user system which means your website should allow users to sign up and login/logout. All users signed up via website should be 'Standard' users and have same privileges e.g., making bookings, retrieve records, cancel bookings, update user data, etc. Website should also have an admin user to perform admin tasks e.g., generate reports such as monthly bookings in a hotel or showing comparison of bookings in all or selected hotels, adding new entries e.g., new hotel, delete or retrieve or update hotel or booking data. Also, explain security design necessary for the website and its data and implement suitable security measures in your website.

Task 4: 25% Website business logic: Your website should fulfil case study requirements and generate required outputs. This should include both static and dynamic webpages as appropriate.

Task 5: 10% LESP aspects: You should explain what LESP aspects are covered in your website and what benefits it provides to end users and HH.

Task 6: 10% Website Demo: Your presentation and demo must be well planned and clearly highlight strengths and limitations of your website. You should provide evaluation of your website project i.e. whether or not the objectives of the project achieved and how well are they achieved? You may also use Powerpoint slides to explain aspects which cannot be covered in the demo e.g., database design diagram, choice of technologies used, test cases/evaluation etc.

Progress reviews signing-off (5%): You should look at tentative list of weekly milestones in Appendix-B. You're expected to discuss your progress with your tutors during practical sessions. There will be up to four progress sign-off opportunities for each student – two in semester 1 and two in semester 2. In semester 1, the main focus will be on Front-end i.e. **HTML/CSS** and **database design and implementation** i.e. Relational model and MySQL (until week 12 of teaching). In semester 2, the focus will be on **Python Flask for business logic** and **any other topic not covered in the first three sign-offs** (until week 22).
Note: Work sent via email will not be entertained.
Whilst you'll be working on your practical exercises, your tutor will allocate some time during practical sessions and sign-off will be on first-come-first-served basis. Each progress sign-off will be awarded maximum 2.5 marks. Do not worry if you missed one sign-off as **marks for two best sign-offs will be selected.** These sign-offs are to motivate you to apply your weekly learning and not leave the assignment work until very last day of submission. In addition, **this is your opportunity to get verbal feedback on your project progress from your tutors**. Please remember your tutors will not do your assignment for you; their feedback will be on whether or not you are making reasonable progress and signpost you to the relevant learning material or resources. Please make sure you have at least completed draft of the work to get it signed-off.

NOTE:
1. You are NOT required to provide a payment mechanism for your website, though you may simulate this. Hint: Think about PayPal payment buttons.
2. Your tutors will act as end users and you can clarify requirements by posting queries on the Discussion Forum.

Section 3: Deliverables

You must use the Blackboard electronic submission system to submit your work. Each student will have to upload complete package individually. Electronically submitted deliverables include:

1. A web system based on the specifications in section 2, with source code in ZIP file (using e.g., 7Zip). This should contain all the files and folders for the full working website. All programme files must have student ID and student name who has written the code.
2. Relevant Database (e.g. MySQL or MongoDB) dump.
3. A presentation in MS PowerPoint (if you are working in Linux these slides can be in PDF format). You should explain through your slides specific tasks which cannot be covered in demo.
4. Create and submit a project presentation and demonstration recording (audio and video) i.e. screen-recording (**up to 8 minutes maximum**) showing/explaining working features of your web site and underlying code and explaining other tasks through your slides. Also, very briefly list what is not working or not completed. You

may choose any software for creating the recording e.g., OBS, Kaltura, Screencast-o-matic, etc. Acceptable recording file types are: .avi, .mp4. Use your recording time wisely. The video will help us to see it is your work. Also, with the help of the video you can get your point(s) across and this can help you to gain good marks.

5. Provide a text file with instructions on how to run your website e.g., index script/webpage and user passwords which you used for your demo video.

Instructions for submission

You should compress all above into one zip file and will submit via Blackboard submission system. The naming standard of the zip file is WP1234567.zip where 1234567 is the student Id. You must submit your work *before* the stated deadline by electronic submission through Blackboard.

- Multiple submissions can be made to the portal, but only the final one will be accepted. Please save your work frequently.
- **It is your responsibility to submit deliverables in a format stipulated above.** Your marks may be affected if your tutor cannot open or properly view your submission.
- **Do not leave submission to the very last minute**. Always allow time in case of technical issues.
- The date and time of your submission is taken from the Blackboard server and is recorded when your submission is complete, not when you click Submit.
- **It is essential that you check that you have submitted the correct file(s),** and that each complete file was received. Submission receipts are accessed from the Coursework tab.
- UWE academic regulations for late submissions will be applied. Please check module handbook.

Note: All submitted work will be checked for Plagiarism using SafeAssign. Every student undertaking an assignment or other piece of assessed work is required to take, and will be deemed to have taken, full responsibility for all the work submitted by the student. In particular, this includes responsibility for any assessment offence (e.g., Plagiarism, collusion) committed.

Assessment offence such as plagiarism or collusion will result in 0 marks.

Section 4: Marks Distribution and Marking Criteria

In common with all UWE standard undergraduate assignments, the pass mark for this assignment is 40%. The detailed marking criteria for this portfolio is detailed in Appendix-A.

Section 5: Feedback mechanisms

Detailed summative written feedback will be provided via blackboard.
There will be opportunity for you to get formative interim verbal feedbacks during practical sessions. Please use Blackboard discussion forum to ask questions about this project.

Appendix A. Marking Criteria

Tasks	0-3	4-7	8-10	11-15	16-20
Task 1 (20): Normalised Database	Little or no DB design; None or very limited explanation; DB is not normalised.	Entities defined and DB design (e.g., ERD/UML diagram) covers some scope of the case study but no explanation of normalisation process or design is in at least 1NF; implementation is partially mapped on to the design	Entities defined with attributes in DB design (e.g., ERD/UML diagram) and cover the partial scope of the case study but no explanation of normalisation process provided and design is at least in 2NF; implementation is partially mapped on to the design	DB design (e.g., ERD/UML diagram) with attributes and relationships are defined and covers most of the scope of case study. Normalisation process explained with examples and design is in 3NF; implementation is partially mapped on to design	DB design (e.g., ERD/UML diagram) with attributes and relationships are correct and cover full scope of the case study. Normalisation process explained with examples and the design is in 3NF; implementation is fully mapped on to design. There is good explanation of design decisions that lead to a good DB design.
Task 2 (15): Responsive Design and Look & Feel	No or little responsive design; No or very little use of CSS for look and feel or web pages;	Partially responsive design; Viewable on two different screen sizes and look & feel of the website is reasonably fine	Fully responsive design; Viewable on at two different screen sizes and nice look and feel (intuitive) of the website.	There is insightful thinking for designing the mobile-first design. Fully responsive design; Viewable on at three different screen sizes and nice look and feel (intuitive) of the website.	
Task 3 (15): User System (10) and Security (5)	Little or no user features; errors on the pages; None or very little security explained or implemented e.g., hard coded login	Partially functional with Standard user i.e. at least one user feature (fulfilling requirements) is implemented e.g., user login implemented and works for user specific functionality e.g., to allow making bookings and show user transactions;	Partially functional with Standard user. At least two user features (fulfilling requirements) implemented with reasonable design and these are error free; e.g., login, cancelling booking with cancellation fee, viewing booking, etc. but suitable examples from the project were missing. Basic security at DB level is provided and explained (2).	Both standard and admin users are implemented. Fully functional and error free; insightful thinking for user features (using both standard and admin user types) demonstrating client-server scripting and correct/complete data storage; good number of user features (i.e. > 4) implemented e.g., changing user account password; adding new hotels, etc. and suitable examples provided. Website and data security is well designed/implemented and explained e.g., SQL injections, hashing, XSS, sessions and authorization token etc (5).	
Task 4 (25): Website Business Logic (Booking process - 15)	Little or no implementation of booking process; Case study data is partially stored and displayed on webpages;	Partially functional booking process; Some of case study data (lookup) is loaded from the DB and displayed; A user is able to enter booking parameters (i.e. dates,	Fully functional process and error free; booking process is reasonable and does not miss necessary data; demonstrates good integration of client and server-side scripting and data storage; booking/room records	Fully functional process and error free; logical and insightful thinking for booking process demonstrating client-server scripting and correct data storage; records are updated correctly; end user is kept well-informed with booking details i.e.	

	Errors on the page; Misses most of the information on the webpage	number of rooms, number of guests, room type, etc.) and system calculates and displays correct prices.	are correctly updated in the database e.g., rooms booked/available, booking id, etc.	screen messages and booking confirmation; demonstration was well planned with suitable examples	
Task 4 (25): Website Business Logic (Other Requirements – 10)	At least one requirement implemented e.g., 3 months advance booking and discounts correctly applied based on booking dates; Website has static (e.g., business info, privacy notices, etc.) and dynamic pages well presented	In addition to implementation of a advance booking requirement, booking cancellation is correctly implemented; Also, Amounts in said currencies is allowed and correctly converted/stored;	Most requirements are well designed and reasonably implemented. Also, careful consideration is given to suitable non-functional requirements (at least 2).		
Task 5 (10): LESP	No LESP or basic aspects of LESP aspects are partially covered in the project	Reasonable LESP issues covered in the project and well explained with at least one example of each LESP aspect covered in the project.	Good understanding of LESP issues and well explained with at least two examples for each LESP aspect covered in the project		
Task 6 (10): Website Presentation and Demo video	Lacks coherence of arguments and demo is not well planned; Some English spelling and grammatical mistakes on slides; Text is less legible; quality of diagrams is often poor; lacks good quality of examples in the demo. Demo/presentation time could be better managed.	Reasonable presentation and demo that covers detailed scope of the project tasks and requirements. Though examples are provided but could be well planned. Test/evaluation could be detailed. Allocated time could be better managed.	Well planned presentation, with reasonable contents, and project demo with reasonable test data/scenario and coverage of project; Good use of allocated time and talked through all necessary aspects of the project; Good quality examples used. Good evaluation of the project is provided.		

Progress Sign-offs have total (5) marks. Each sign-off will reward up to 2.5 marks (criteria: No sign-off or no progress: 0 mark; Did some work but very limited progress indicated: 1 mark; Reasonable progress made and mostly on track: 2 marks; Very good progress made indicating that student is on track to complete and submit the assignment: 2.5 marks).

Appendix B. Tentative Milestones

Week 1 and 2: Coursework is released and you have read and understood your project brief
Week 3 and 4: You have clarified any questions with your tutors and have started planning your coursework.
Week 5 and 6: You have started to design front end e.g., sitemap, static pages, static contents, logos, privacy statements, CSS style rules, media queries
Week 7 and 8: You have started to design database model.
Week 9 and 10: You have completed data model and SQL queries written which can then be used in python script
Week 11 and 12: You have started designing server-side end points i.e. forms, data transfer, flow of information
Week 13, 14, 15, 16: You have started to implement your server side script using Flask.
Week 16, 17, 18 : You have completed the core business logic/functionality and have started to work on client-side scripting for dynamic interaction and client-side modification
Week 18-19, 20: You have started to put together your slides and also have started to implement security aspects
Remaining weeks: Your project is complete and you have started to review and refine your website. Make sure upload the website and required deliverables