|  |  |
| --- | --- |
| **C:\Users\sram\OneDrive - Unitec NZ\ShiuRam\Unitec\Unitec Logos\UnitecHorizontalLogo.jpg** | **ISCG7420**  **Web Application Development**  **Assignment 2**  **Semester 2, 2022** |
| School of Computing, Electrical and Applied Technology | **Due Date: November 18th, 2022, 23:00PM** |
|  | **Total Marks: 100**  **Course Weighting: 40%** |

**Learning outcomes covered in this assignment**

1. Discuss the philosophy of client-server computing and its impact to the computing industry.
2. Design and develop a database client-server solution that meets specified organisational requirements using database and modern data access technology.

# Assignment information:

* This is an individual assignment.
* Read the given scenario and complete the two tasks.
* For Task 1 you will research the uses of the client-server computing in industry and present your findings within a count of 800 words [+/- 10%], excluding reference list.
* For task 2 you will design and implement a Full Stack web application.

# Assignment Submission:

* Upload your response for task 1 in word format to Moodle “assignment 2-task 1 submission link”.
* For task 2 use the Assignment 2directory of your class GitHub repository to store your task and add your lecturer as a collaborator. Periodically throughout the course your lecturer will clone your repository to check your progress. On the assignment due date (and for up to 72 hours afterwards) your lecturer will download the most recent version of your task 2. You are required to ensure that GitHub always has the latest version of your code.
* For task 2 if you forget to push your most recent changes by the due date, then your lecturer will use the latest version available on GitHub when the deadline is reached for marking. **Make sure you regularly update files. If you use and submit code that you didn’t write, you will fail this assignment, and fail the course. Ask your lecturer for help long before the due date instead of plagiarising.**

**Scenario**

Maungawhau college is upgrading their attendance system. This system allows lecturers to enter student attendance. Students can also view their attendance on the system.

Business Rules:

1. One semester runs one to many courses
2. One course is run in zero to many semesters
3. One course can be separated into one to many classes
4. One class can only run for one course
5. One class can be taught by only one lecturer
6. One lecturer teaches zero to many classes
7. One class holds one to many students’ enrollments
8. One enrollment fit in one class
9. One student enrolls to one to many courses.
10. One student enrollment belongs to one student.
11. One college day runs zero to many classes
12. One class runs on zero to many college days

**Client requirements:**

1. Administrator can login into in the attendance system.
2. Administrator can create/update/delete/show semesters
3. Administrator can create/update/delete/show courses
4. Administrator can create/update/delete/show classes
5. Administrator can create/update/delete/show lecturers
6. Administrator can assign/remove/change/show a lecturer to a class
7. Administrator can create/update/delete/show student
8. Administrator can enrol/remove/show student to classes
9. Administrator can upload students from excel files to the attendance system
10. Administrator/Lecturer can email students when they have poor attendance
11. Lecturer can login into the attendance system
12. Lectures can enter students’ attendances into the attendance system
13. Students can login into the attendance system
14. Students can view their attendance in the attendance system.

Note: there is no need to change password and there is no need for registration.

**Task 1: Research [16 Marks]**

Before you start web development (i.e., in this case gradebook system) you will do research to identify the process and uses of the Full Stack programming in industry. Present your findings within a count of 800 words [+/- 10%], excluding reference list.

**Task 2: Web Development [84 Marks]**

To develop the attendance system, you need to design and implement a Full Stack web application using Django, Rest Framework, React JS and/or other available web development tools. You will also host your web application on AWS EC2 with available add-ons. You will upload your code to GitHub. The requirements for the application are given below:

**Part A – Serializers [12 marks]**

1. Course
2. Semester
3. Lecturer
4. Student
5. Class
6. College Day

**Part B: ViewSets and permissions: [28 marks]**

1. View for administrator to login into in the attendance system.
2. View for administrator to create/update/delete/show semesters
3. View for administrator to create/update/delete/show courses
4. View for administrator to create/update/delete/show classes
5. View for administrator to create/update/delete/show lecturers
6. View for administrator to assign/remove/change/show a lecturer to a class
7. View for administrator to create/update/delete/show student
8. View for administrator to enrol/remove/show student to classes
9. View for administrator to upload students from excel files to the attendance system
10. View for administrator/Lecturer to email students when they have poor attendance
11. View for lecturer to login into the attendance system
12. View for lectures to enter students’ attendances into the attendance system
13. View for students to login into the attendance system
14. View for students to view their attendance in the attendance system.

**Part C: React JS templates: [10 marks]**

1. Template(s) for all administrator’s functions
2. Template(s) for lecturers’ functions
3. Template(s) for student’s functions
4. React JS Navigator on all pages
5. Templates can be reused

**Part D: Axios [28 marks]**

1. administrator to login to the attendance system
2. administrator to create/update/delete/show semesters
3. administrator to create/update/delete/show courses
4. administrator to create/update/delete/show classes
5. administrator to create/update/delete/show lecturers
6. administrator to assign/remove/change/show a lecturer to a class
7. administrator to create/update/delete/show students
8. administrator to enrol/remove/show students to classes
9. administrator to upload students from excel files to the website
10. lecturer to login to the attendance system
11. lectures to assign students to the attendance system
12. student’s to login to the attendance system
13. students to view their attendance in the attendance system
14. auth token is used in login

**Part E: Deployment of Rest Framework [3 marks]**

**Requirements:**

1. Setup Github
2. Setup Heroku/AWS/other host apps that corresponds to GitHub repositories.
3. Deploy your website to Heroku/AWS/another host.

**Part F: Deployment of React JS [3 marks]**

**Requirements:**

1. Setup Github for React JS
2. Setup Heroku/AWS/another host apps that corresponds to GitHub repositories
3. Deploy your website to Heroku/AWS/another host

**ISCG7420**

**Web Application Development**

**Assessment 2 Mark Sheet**

**Student Names:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Task | Maximum  marks | Your mark | Comment |
| 1. Research | 16 |  |  |
| 1. Web Development |  |  |  |
| 1. Serializers | 12 |  |  |
| 1. ViewsSets and permissions | 28 |  |  |
| 1. React JS templates | 10 |  |  |
| 1. Axios | 28 |  |  |
| 1. Deployment of Rest Framework | 3 |  |  |
| 1. Deployment of React JS | 3 |  |  |
| **Total** | **100** |  |  |

**Late Submission of Assignments:**

Assignments submitted after the due date and time without having received an extension through Affected Performance Consideration (APC) will be penalised according to the following:

* 10% of marks deducted if submitted within 24hrs of the deadline,
* 20% of marks deducted if submitted after 24hrs and up to 48hrs of the deadline,
* 30% of marks deducted if submitted after 48hrs and up to 72hrs of the deadline,
* No grade will be given for an assignment that is submitted more than 72hrs after the deadline.

**Assistance to other Students:**

Students themselves can be an excellent resource to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognise what types of assistance are beneficial to another’s learning and also what types of assistance are unacceptable in an assessment.

**Beneficial Assistance:**

* Study Groups
* Discussion
* Sharing Reading Material
* Reading the available online and library resources

**Unacceptable Assistance:**

* Working together on one copy of the assessment and submitting it as own work
* Giving another student your work
* Copying someone else’s work, this includes work done by someone not on the course
* Changing or correcting another student’s work
* Copying from books, the Internet etc. and submitting it as own work; anything taken directly from another source must be acknowledged correctly; show the source alongside the quotation
* Don’t copy code from a website or video tutorial and pretend you made it or slightly change it. This will be an instant fail (0%).