Department of Accounting & Information Systems

ACCT/INFO GROUP PROJECT SUBMISSION

CASE STUDY/PROJECT TITLE

Please complete all sections of this sheet, sign the declaration and attach the sheet to your project.

The next panel must be completed by all team members, **including** the agreed proportion of work done on the project. (For example. if all members of a team of four made equal contributions then enter 25% for each team member.)

	Student ID No.	**User ID** e.g. *afg21*	**Student Names:** (**Surname first** & alphabetical order please)	**Proportion %** (Agreed by group)
1)	47885965	dal113	ALTALIM Dianame	25%
2)	49113406	sfr67	FRASER, Shannon	25%
3)	15995114	ckh35	HAMDAJANI, Christopher Kevin	25%
4)	32321562	hma112	Martin, Harrison	25%
5)				

Honesty Declaration

- I declare that this is an original assignment and is entirely my own work.
- Where I have made use of the ideas, words or work of others, I have acknowledged the source in every instance.
- Where I have used any diagrams (including modifications) prepared by others, I have acknowledged the source in every instance.
- I have read and understood the Dishonest or Improper Practices Statement overleaf.
- I am aware of what constitutes cheating, and the penalties for plagiarism and cheating as described in University publications.
- I am aware that the content of this written work may be checked against an electronic database.

- I have supplied the correct word count and have taken no steps to cause disclosure of an incorrect word count for the assessment.

I have read and fully understand the Honesty Declaration above, and hereby certify that this item of work submitted for assessment is entirely the work of the members of the group, in the proportions stated.

Signed . . .

1) 2)	3) 4)	5)

Under the University Regulations, evidence of any of these or other forms of dishonest practice by any student(s) represents grounds for disciplinary action and may result in penalties ranging from denial of credit for the item or work in question, to exclusion from the University.

Dishonest or Improper Practices

It is recognised that students will discuss course work and assignments with others, and such discussion is an important part of the learning process. However, any work presented by a student for credit in a course must be that student's own original work. If students are directed to complete work submitted for credit in groups, the work submitted must be the original work of the group. Work submitted in breach of these requirements or which fails to comply with other instructions contravenes the University's Dishonest Practice and Breach of Instruction Regulations. Such work will either not be marked, and all credit for the work in question forfeited, or the matter will be referred to the University's proctor for investigation and possible referral to the University's Disciplinary Committee.

Penalties which may be imposed in the event of a finding of dishonest or improper practice include loss of credit for a course or an item of assessment and, in serious cases, suspension or expulsion from the University. A record is kept of all instances of dishonest conduct.

Instances of dishonest or improper practice in coursework and assignments include but are not limited to:

- Plagiarism. Plagiarism means the dishonest presentation of work that has been produced by someone else as if it is one's own. Please note that the presentation of someone else's work as one's own, even without dishonest intent, may still constitute poor academic practice, and this may be reflected in the mark awarded. There are academic conventions governing appropriate ways to acknowledge the work or part of the work of another person, including the APA and Harvard citation styles. For further information see the UC Library website, under "Citations and Referencing".
- Submitting for credit in a course without the prior consent of the Course Coordinator for an essay, research paper or any other written work which, although it is the student's own work, is substantially the same as work which has already been (or will be) submitted for credit in another course, whether in the Department of Accounting and Information Systems (ACIS Department) or some other department or academic institution.
- Copying the work of another student. This includes copying the work submitted by another student for credit for a course in the ACIS Department or some other department or academic institution.
- Knowingly allowing another student to copy work which that other student then submits for credit

for a course in the ACIS Department.

- Arranging for another person to complete work which is then submitted for credit for a course in the ACIS Department. An example falling in this category is work submitted for credit which has been obtained from a commercial assignment completion service. Care must be taken when using editing services as it is **only** assistance with grammar, punctuation and expression that is permissible and does **not** include the addition or amendment of content.
- Completing work for another student which is then submitted by that other student for credit for a course in the ACIS Department.
- Including made up or fabricated material in work submitted for credit for a course in the ACIS Department.
- Collaborating in the preparation of answers for take home or online tests unless advised otherwise in the take home test instructions.

If you are in doubt about any of the above with respect to a particular course, you should discuss the matter with the lecturer or course co-ordinator concerned.

See also the University Discipline Regulations, Dishonest Practice and Breach of Instructions Regulation, and Academic Integrity Policy – refer to UC Calendar and UC web.

Acct/Info 263 Group Project

Note: Username: admin Password: 1234

Interface

The interface we chose for displaying the events was created using a table as our main method of presenting the data, we used a JQuery plugin called JSGrid for this as it was lightweight and allowed for such things like sorting the table and because it was lightweight in terms of resource use and implementation.

The table displays all the events that a user wishes to see, with the ability to search by date or search by event name.

Creation

We began the creation of this solution by first discussing and outlining the project brief, what codebases we were going to use, and how we were going to approach the implementation of our solution. We initially decided that we would begin by creating a simple HTML outline of our solution, so that we had a basic structure into which we could build both the functionality and presentation aspects. Reflecting on our initial project decisions we potentially could have spent more time outlining our solution and discussing our decision-making process to have created a more robust solution in a more efficient way. However, all the decisions we made during this project have led to learning experiences or successful outcomes, and therefore were a valuable process to go through.

Once we had a basic HTML layout, we then moved onto developing the interface more. We started by building all the server-side code in server.php, originally, we were making SQL queries straight from the server.php file to the database, once we had more of an idea of what procedures we wanted on the database side we implemented those.

When the server-side code was up to a stage we were happy with we moved onto the client-side code, working on the JavaScript so that we could make AJAX calls from the client to the server, and present that data.

After that we wrapped up the project by working on styling, and we have created a interface that we think not only functions well, but looks good too.

Login

As detailed in the project brief our solution firstly requires the user to login. We ensure the user is logged in for every page within the solution as none of the pages should be accessible if the user is not logged in. If a user tries to access a page whilst not logged in, they are redirected to the login page.

Once a user has successfully logged in, a session is created. This is what is checked for on each page using PHP.

We took some creative liberty for the login page, making use of JQuery animations and stylistic choices.

Database exchange.

For the information exchange between the database and client we decided to have the client send and receive data by having jQuery make post requests to the server where server.php would use a switch statement to direct requests to specific functions. These functions would make calls to the database using predefined procedures, resulting in the data on the database being changed in some way - added to, edited, or deleted – or having information being fetched. Information that is fetched is then packaged up into a JSON (a lightweight data-interchange format) and sent back to the client. JavaScript would unpack and then present, or modify, the information depending on its use.

We decided to go with the passing of information between different modules/classes approach as it allowed for easier debugging and testing as each module or file the process does specific things e.g., server.php only receives information, makes calls to the database, and returns that information. This simplified our files and made them more readable.

Event Creation.

Event creation is accessed through the navbar. The user is presented an input box where they can input the event name. This input information is then sent to the server, which in turn makes a call to the database which creates the event and returns the generated event ID. Once this ID has been returned to the client-side it moves onto the next page, where the user can add all the relevant information for the event. All of this is then sent to the server when the user submits the form. The server then breaks down the $_POST data into its various bits and sends these to the various functions that are used to create the event.

Searching.

There are two separate search features in out implementation.

The search bar works as soon as the user starts typing into the search bar. Each time there is a new value inputted it sends this input data off to the server which in turn searches the database and returns any data in the database that matches the input data. This is packaged and sent back to the client where it is presented to the user in the form of a dropdown list. Once the user clicks on one of the list items it populates the table with all the events that match that list item.

The second search feature is the ability to find events by date. Below the navigation bar there is a section where the user can input a start date and/or an end date of interest. Once the date has been inputted and the user clicks the search button which is located beside the end date input field, it will send the data to the server and run a function which returns any data within the range of start date and end date which was inputted by the user. The returned data will be used to populate the table to be represented to the user.

Displaying Event Numbers

On the index page you can see four buttons that each have numbers beside them, representing the count of how many events there are in total, how many events that have passed, events that are coming up, and the events that are currently happening. This is done by the client-side code making a call to the server which in turn gathers all the events and sends it back in an array. Once the client gets this all back, it goes through each item in the array and compares it to the current date, from there it puts the results into their respective arrays and counts each array and presents this. The user can also click on any of the buttons to have the events displayed in the table sorted by the event status.

Logout

Lastly, for users to be able to log out, we provide a log out button which is placed beside the search bar. This log out button will call logout.php will remove/destroy the session, which was created when the user logged in, and redirect the users to the login page, which then asks the user to log in again.

Version-Control

During the process of creating this solution we ensured consistency throughout our separate code bases by using version control software and keeping the repository on GitHub while also keeping up communication between members. This was a foundational part of our implementation process as it allowed all members of the group to contribute to the code without all needing to be in the same location, whilst also ensuring functionality and consistency of the code. During the coding process we also made sure to have good documentation in the code so that each person could quickly and easily understand someone else's code and make effective changes to it.

All these project tasks were being worked on and implemented simultaneously by the group members, with initial testing being done by the members who created the code and more in-depth testing being done by the members not familiar with the code. This allowed us to discover and fix both usability issues (UX testing) and bugs simultaneously, with members providing feedback and suggesting solutions to the whole group.

Reflection

Throughout the project the group met on multiple occasions to discuss the project, what our timeline were for individual tasks, who was doing each of the tasks, and other assorted project management tasks. We worked very well as a group and the overall experience was both enjoyable and educational, allowing us to become more confident with creating and implementing a solution. Diagnosing and solving code-based issues within the group allowed opportunities for knowledge to be shared and led to a more efficient and well thought out solution. On some occasions sharing solutions allowed us to diagnose other issues and solve them simultaneously. Not only this but it also allowed group members with more knowledge and experience with specific libraries and languages to support.

Whilst overall we believe this was a sufficient way to create and implement the solution on reflection, we have found some changes regarding the decision making and implementation process for future projects. This includes, but is not limited to, a more thorough planning stage, regarding both the finished product and the steps to take to get to the finished product. This would have made our time usage more efficient during the project as we would have had a more concrete plan regarding the final product and therefore how to go about creating it.

Another reflection would have been maybe to have divided up the different coding elements a bit more to allow for members to get a better understanding of all the parts of this project, while we were commenting all our code, there were times where it was hard to get your head wrapped around what other members had written, this was mostly due to this being the first proper web project most of us had done and with time and practice this won't be an issue in the future.