Department of Accounting & Information Systems

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

ACCT/INFO GROUP PROJECT SUBMISSION

CASE STUDY/PROJECT TITLE INFO263 Group Project

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)	68282457	sga146	Simon Garratt	33%
2)	83839332	qho15	Quinn Houghton	33%
3)	71802950	dla82	Dave Larter	33%
4)				
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) SGarratt	**3)**	**5)**
2) qhoughton	**4)**	

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.

INFO263 – Group 3

Web Server UI Solution

Setup

Tserver dump file.

Database dump file for this solution has been edited. It is in the database folder and must be used for proper functionality of project.

Description

Index

Designed to pull data from index.html then check it against the user table in the database. Returns session variables for checking in further pages. Still requires proper integration with index.js to have functionality.

New Events

Standard form built with a combination of basic html, jQuery Ui and Bootstrap.

Bootstrap was used mainly for its layout methods and buttons on this page. jQuery Ui was used for the date-picker input for better user functionality, instead of picking a day, week and year, the user instead just picks a single day which is changed by the php scripting.

New events currently doesn't have the capability to create events with multiple clusters or machine groups. This would require looping sets within the php file, that called stored functions multiple times.

New_event.js contains four main functions:

1. Onload functions, calls two different functions, and tells the form to use jQuery Ui for the datepicker element
2. getGroup and getClusters both function in the same way, an AJAX get request to grab the cluster and group lists from the server then using simple iteration loops over the returned data and fills the drop downs.
3. The last function covers the Event submission using a AJAX post request. The function first grabs the values from the form then compiles them into a single data piece for the request.

New_event.php contains the necessary functions to submit and entire event in several steps.

1. Escaping the inputs from the post request, removing any special characters.
2. Creating a new DateTime variable with the received date, and separating it out to day, week, and year.
3. Setting the event id through calling new_event() – this took the connection and the submitted name for a stored procedure and returned the event id assigned from the database.
4. Calling three stored procedures from the original skeleton file: add_event_week, add_daily and add_action. These built the event within the database given the correct values.

A problem arose with multiple queries on the database, this was solved by utilizing close and next result functions.

```
$stmt->close();
$conn->next_result();
```

View Events

View events uses three tables to function efficiently:

- Search table
- Event table (Main)
- Detail table

Search table includes the search bar and produces results as you type. Instead of pushing an AJAX request every time a key was pushed, I instead looked online for a local example and implemented a search that creates a new array of results from a given array.

The event table iterates over the data from the original Ajax request, pushing it into a table with a button linked to the wider event in the array.

When the button is pushed, it shows the detail table and fills it with every iteration of the event, with further detail than the event table.

View_events.js contains multiple functions regarding the filling and clearing of tables. It also contains several different jQuery listeners focusing on the buttons and how the tables react when the buttons are pushed.

View_events.php only contains the original event view query but echo's a json_encode of the results array instead of a string.

Edit/Delete events

Designed to pull events from database into text inputs then resubmit to the views or delete them entirely.

Additional Libraries

All the project files are linked to external sources.

jQuery

jQuery – v3.6.0-min

```
<script type="text/javascript" src="https://code.jquery.com/jquery-
3.6.0.min.js"></script>
```

jQueryUi – v1.12.1

```
<script type="text/javascript"
src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
```

Bootstrap

Bootstrap – v5.0.1

```
<link type="text/css"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.c
ss" rel="stylesheet" integrity="sha384-
+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x"
crossorigin="anonymous">
```

Bootstrap Icons – v1.5.0

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
```

Tserver Dump file changes

Views

Cluster List – listing all available clusters

Machine List – listing all available clusters

User List – lists all users

Stored Procedures

Add_event() – takes an event name, inserts into table, and retrieves event number

Get_event_number() – gets the highest event number – was removed from usage

Insert_user – inserts new users into database