

2021 Web Application Group Project Specification

Weighting:	30%
Due date:	Submissions on Friday 14 May (Week 9) & Monday 14 June (SWOT) Please refer to the Milestone 1 and Final Submission sections for details.

▾ Aims

1. Understand principles of designing user friendly websites.
2. Write standards compliant websites in HTML & CSS.
3. Develop dynamic, client-side web content using Javascript.
4. Understand the interactions between the client-side and server-side components of web applications.
5. Design and build AJAX calls to a server and handle responses.
6. Use third party APIs in web applications.
7. Understand the role of databases in web applications.
8. Plan & integrate a database into a web application.
9. Understand and mitigate security issues faced by web applications.

▾ Project Overview

Description

Your group assignment for the Semester is to design and develop a COVID-19 Contact Tracing Web Application that allows users to check-in when they visit businesses/public places, shows hotspots/areas to avoid, and alerts health officials & the user if the user has come into contact with a hotspot.

Detailed information on features and functionality is provided in the Requirements and following sections.

Submission

This is a **big** project and can't be completed at the last minute so to keep you on track we've set two milestones:

1. A Milestone submission at the end of week 9
2. A Final submission at the end week 13.

Be aware that there are no extensions on the Final submission.

Assessment

This project is worth a total of 30% of your final course grade and is NOT a hurdle requirement.

Detailed marking rubrics will be available on the MyUni submission page for each part.

Milestone	Due	Weight
Milestone 1	11:59 pm Friday 14 May Week 9	12%
Final Presentation	11:59 pm Monday 14 June Week 14/SWOT	18%
TOTAL		30%

Individual Performance

Your final mark will be scaled (reduced or increased) around the group's overall mark depending on your individual performance, which is determined based on a combination of peer review by your group and review of your contributions by the teaching staff.

Workshop Sessions

Weekly workshop sessions are a great opportunity to ask questions and get feedback from the lecturers and tutors before the submission deadline.

We recommend using this time to liaise with your group members to get collaborative work done. You are welcome to attend sessions that you are not enrolled in as needed to complete group work.

Core Requirements

! Important

The requirements listed in this section must be followed.
Not adhering to these requirements may result in a significant reduction in marks or even 0 for the assessment.

Software Requirements

- The web application must be **developed in and run in** the CS50 IDE.
 - You should also make regular (at least weekly) backups/downloads of your work.
 - You are encouraged to setup a dedicated GitHub and CS50 account for groupwork with shared access to all group members to avoid accidentally sharing your personal work.
 - You must get explicit permission from the course coordinator if you want to use a different environment.
- The web application must use NodeJS, Express, and AJAX .
- The database must be a SQL database.
- Your markup/code must pass W3C Validation and ESLint linting.
- You may use other libraries/frameworks not addressed in this course, however:
 - Remember that you're working in a group; all group members are expected to have contributed to and understand the work developed.
 - Your implementation must still demonstrate your understanding of the concepts taught in this course.
 - We may be unable to mark work that deviates too much from the course content.

Group Requirements

- This assignment must be worked on in groups of 4.
- Groups must be formed before the mid-semester break (end of Week 6).
- You must [register your group in MyUni](https://myuni.adelaide.edu.au/courses/64700/groups) (<https://myuni.adelaide.edu.au/courses/64700/groups>).
- You may form a group with any other students in this course, but it will be easier to organise your group if they are in the same workshop as you.
- Postgraduate Students (enrolled in COMP SCI 7207) will be assigned groups.
- See "Forming A Group" below for details/advice on forming groups.

Feature Requirements

- You must implement a COVID-19 contact tracing website/system.
- Your system should contain features as described in the "Feature Details" section below.
- Postgraduate Students (enrolled in COMP SCI 7207) are expected to implement an additional feature from the special features section.

Feature Details

Your COVID contact tracing web app should include, and will be assessed on, the following features:

- Users should be able to sign up and log in to the system in order to
 - Manage their user information.


- Check-in to locations by entering a check-in code.
- View their check-in history on a map.
- See current hotspots on a map.
- See if they've been to a hotspot.
- Venue Owners/Managers should be able to sign up and log in to the system in order to
 - Manage their venue information.
 - View the check-in history for their venue.
- Admins/Health Officials should be able to log in in order to:
 - Manage their user information.
 - Create and manage hotspot areas/venues & timeframes.
 - View the check-in history for users and venues.
 - See current hotspots on a map.
 - Manage Users/Venues.
 - Sign-up other Health Officials.
- Users/venue managers should be able to choose to link a social media/email/other account, allowing login via that platform, to make logging in easier.

One (1) of the following three (3) special features must also be present (2 for PG Students):

- Email notifications
 - Users can request that email notifications be sent to them that include a list of current hotspots and/or whether they've been to a hotspot.
 - Users should be able to choose which types of email notifications are sent
 ~ OR ~
- GPS Location
 - Users can check in using their browser's location
 ~ OR ~
- QR Code Scan
 - The system automatically generates a QR Code page for each venue that can be displayed or printed by that venue.
 - When the QR Code is scanned by a logged in user, it checks them in.

▼ Forming a Group

For UG students this project is completed in groups of 4.

- Find 3 other group members enrolled in the same workshop session as you.
 - You can use the "[Search for Teammates](https://piazza.com/class/kirbdk29i9y65q?cid=5)  (<https://piazza.com/class/kirbdk29i9y65q?cid=5>)" feature in Piazza to help find a group.
 - You may form a group with any other students in this course, but it will be easier to organise your group if they are in the same workshop as you.
- Once you've found a group, [register your group in MyUni](https://myuni.adelaide.edu.au/courses/64700/groups) (<https://myuni.adelaide.edu.au/courses/64700/groups>).
- Groups must be formed before the mid-semester break (end of Week 6).

- After Week 6, groups will be locked and anyone not in a group will be randomly assigned one.

PG students will be manually assigned to a group.

▸ Preparing for Milestone 1

Research

Think about and explore web pages & apps. What might a COVID-19 contact tracing system look like? What information might users be required to supply? How will they log in? How will users check in? How will hotspots be shown on the map? How will health officials interact with the check-in data? You are encouraged to think about additional features for this application; we have outlined the minimum.

- Research existing websites/apps to get ideas for your web application.
 - Record any pages/services you find that are good examples of what your site could look like. What parts work well. What parts don't work well?
 - Also record any examples you find of features you might want to include in your site.

Basic Design

Once you decide what a user will be able to do, then you need to think about how they will do each action and how you will make those actions visible on your site – keeping in mind the articles and lectures on design and usability.

- Draw/Sketch (don't code) a basic design for what your site will look like.
 - It should be obvious from the drawing what each part of the site does.
 - Be sure to include any dimensions for the layout of the site and its components:
 - Are components fixed in size or do they stretch?
 - If the window is resized, how do elements change?
 - How might the site look on a phone vs desktop?
- Thinking about the features in the description section, start to plan how each of these features of your site will function, plus any additional features that you might like to have.
 - Where does this feature appear and how does the user interact with it?
 - What happens when a user performs one of these interactions?
 - If a change occurs to the page/site following an interaction, clarify what that will be and draw if necessary.

Describe these for each feature.

Once you have a basic design, you now need to review and tweak it for usability and accessibility.

- Review your design.
 - How does it minimise kinematic and cognitive load?
 - Does it meet the standards & heuristics referenced in lectures and on MyUni?

- Have another group review your website design.
 - Which parts of your site were clear. Which weren't?
 - Record the feedback from the other group and any findings.
- Record any updates to your design made as a result of the above review.

As you learn more about implementing web applications throughout this course, your design will likely change; that's okay, but be sure that as those changes creep in, you continue to review your site for usability and accessibility. We will be assessing you for this in your final submission.

Your week 7 workshop time is a great opportunity to ask questions, get feedback, and compare your design with another group before the deadline.

Building your Web App

Now that you have a solid design, it's time to start coding it! You should now have a firm grasp on these languages and be able to implement ALL of the static elements of your website.

- Use HTML and CSS to implement all of the parts of your site.
- Follow best practices as outlined in lectures and exercises.
 - Ensure CSS is used appropriately to provide a consistent and maintainable style across all your pages.
 - Ensure your implementation passes validation.

As you build your website, begin to add interactive features using JavaScript and Vue.

- Use JavaScript and Vue to further build your design.
- Start to add client-side code for users / venues / health-officials to interact with the system
 - Use test data/variables to simulate client-server interactions as needed and to help you test your code.
 - Start to build your map features with one of the Map APIs shown in lectures.
- Follow best practices as outlined in lectures and exercises.
 - Ensure JavaScript files are used appropriately to provide a consistent and maintainable code across all relevant pages.
 - Ensure your implementation still passes validation after modifications.
 - Ensure your JavaScript and Vue passes linting.

Do not worry about making user interactions persistent (i.e. user changes remain after a refresh) at this stage, but do begin to think and make notes about which interactions will need server resources.

Planning your Routes and Database

Now that you have a solid design and basic client side implementation, you can start thinking about the content that your web application will be dealing with, and how.

- For each of the features in your website, create a data plan that lists the different pieces of content/information that your web application will be dealing with & determine where it should be stored, what format, and where it should be processed.
 - Where does the information come from?
 - What form should it take?
 - If the information is on the server, what will the client need to send to retrieve that data?
 - If the information is on the client, how will it be sent to the server?
 - Does all information need to be stored on the server?
 - What processing needs to be done to make the data useful?

Once you have an idea of the different pieces of data that you'll need to be working with, you can start thinking about how you'll store that data on your server.

- Using the methods outlined in lectures and pracs, create a Database Schema diagram for your web application's data.

▼ Milestone 1

Submit all work done so far to the MyUni Group Project Milestone Submission page before the deadline, **11:59pm Friday 14 May (Week 9)**.

This should include the following deliverables as described in the section above:

- Your research, design, feedback, review and changes
- Your basic client side implementation
- Your data plan and Database Schema

Submit your group's work as **a single ZIP file**;

- Be sure to include all your work in the ZIP file.
- If you have your files organised in folders/directories, be sure to keep this structure when adding them to your ZIP file.
- Do not include your node_modules folder in your submission, but do ensure you include any other server files.
- Test your work, verifying that running install and start commands works as expected and nothing is missing.
- You will also need to provide the markers access to your CS50 IDE
 - Provide access by sharing your group's workspace with the following GitHub usernames
 - `ian-knight-uofa`
 - `wdc-marker-uofa`

- This will allow us to test your work if files are missing from your submission, and review your progress.
- **It is your responsibility to ensure that your code works in the environment it will be marked in before submission.**

This submission is worth 12% of your final course grade

- You will be assessed on:
 - Site Design
 - Feature plan
 - Basic Implementation
 - Data Plan
 - Database Schema
- A detailed mark rubric will be available on the submission page no later than 2 weeks before the deadline.
- Your mark will be available in the MyUni gradebook within 2 weeks of the deadline unless otherwise noted.
- **Late submission penalties apply!** (see course outline)

▾ Preparing for the Final Submission

Client-server integration

Further develop and implement your design.

- Follow best practices as outlined in lectures and exercises.
 - Ensure modifications don't affect validation of your site.

It's time to set up the server so your website can be accessed on other computers!

- Following the lecture notes and documentation on MyUni, set up an Express server and migrate your website to it.
- Test to ensure all parts of your website are working as expected now that they are being served by the Express server.

You're now ready to start handling data on the server.

- Using a combination of GET/POST methods and AJAX, modify your website and server to implement the calls needed to handle the content/information for each of your features as identified in your data plan.

Referring to the appropriate APIs and documentation, implement your chosen special feature:

- Email notifications using the Nodemailer library/API.
~ OR ~

- GPS Location using the Web Geolocation API.
~ OR ~
- QR Code Scan using a QR Code generation library

Improve your login system to allow users to link a social media/email/service with OpenID.

- Integrate OpenID into your web application's login system.
 - Ensure OpenID authentication is verified correctly.
 - You can use any identity provider or any combination of providers.

The main functionality of your web application should now be almost complete.

- Continue to build your web application implementing any remaining features as described above.

Database Integration

Using your database schema, you can set up your database and begin writing queries.

- Convert your Schema to SQL
- Setup your SQL database and tables.
- Write down the queries needed to store and retrieve data in your database.

Once your database has been built, and queries written, you can now integrate it with your web application.

- Integrate your database and web application.
- Follow best practices as outlined in lectures and exercises.
 - Ensure SQL statements follow the recommended style.
 - Ensure your implementation is safe against SQL injection.

Check Security

The main functionality of your web application should now be almost complete.

- Review and test your web application for functionality and bugs.
- Test the security of your web application, ensuring it is not prone to SQL Injection, XSS, XSRF, and that its authentication cannot be easily bypassed.

▾ Video Demonstration

With your group, record a video of at most 5 minutes duration in which you demonstrate the features of your web application.

- You will not be assessed on presentation skills/delivery/preparation; only on the functionality of your web application.
- This will help your markers to ensure they don't miss anything.
- Zoom includes recording capabilities.

Final Submission

Submit all work to the MyUni Group Project Final Submission page before the deadline, **11:59pm Monday 14 June (Week 14/SWOT)**.

This should include:

- Your complete implementation.
- Your database schema & queries should be in the form of a plain-text `.sql` file containing the CREATE TABLE statements of the schema, and your draft queries.
- A backup of your final database.
- A README file containing any relevant information on how to run the work.
- Your video presentation or a link to it.

Submit your group's work as **a single ZIP file**;

- Be sure to include all your work in the ZIP file.
- If you have your files organised in folders/directories, be sure to keep this structure when adding them to your ZIP file.
- Do not include your node_modules folder in your submission, but do ensure you include any other server files.
- Test your work, verifying that running install and start commands works as expected and nothing is missing.
- You will also need to provide the markers access to your CS50 IDE
 - Provide access by sharing your group's workspace with the following GitHub usernames
 - `ian-knight-uofa`
 - `wdc-marker-uofa`
 - This will allow us to test your work if files are missing from your submission, and review your progress.
- **It is your responsibility to ensure that your code works in the environment it will be marked in before submission.**

This submission is worth 18% of your final course grade

- You will be assessed on:
 - All progress continuing from Milestone 1, plus

- Server Implementation.
- Login System.
- Database Implementation.
- The overall functionality and quality of your system as demonstrated.
- A detailed mark rubric will be available on the submission page no later than 2 weeks before the deadline.
- Your mark will be available in the MyUni gradebook within 3 weeks of the deadline unless otherwise noted.
- **Late submissions NOT ACCEPTED.**

▾ Peer Review & Individual Performance

We know that group projects can sometimes be challenging, especially if everyone isn't contributing the same amount.

If you experience issues within your group that cannot be reasonably resolved, such as a group member becoming unresponsive to communications, then please message your course coordinator on MyUni sooner rather than later.

Following your final submission, you will need to complete a peer-review survey.

- The survey will ask you to rate the contribution of each of your group members to the overall project, as well as any notable contributions by group members, and what part(s) they contributed most toward.
- The results of this survey will be averaged out for each group member and if their contribution is significantly higher or lower than expected, that group member's final grade for the project will be scaled accordingly.

The teaching team will also review your contributions to the work submitted and, the course-coordinator may adjust or apply further scaling if there is evidence to show that a group member's contribution was significantly below what is reasonably expected for the mark attained.