



PROJECT
Web Technologies
SECONDARY

1 INTRODUCTION

This pre-release project is intended to give you an idea of what to expect for the two-day competition in Winnipeg. While there are samples of tasks provided here, the scope of web technologies is wide and so you will be challenged in a variety of areas not represented in these samples. Please see the [Contest Description](#) for more details.

The two days of competition will be broken into two three-hour modules each for a total of 12 hours. Day 1 will include a three-hour web design module followed by a three-hour JavaScript game development module. Day 2 will be comprised of a series of mini-tasks which will touch on a wide variety of web technologies. More details are provided below to help you start practicing for this competition.

2 DESCRIPTION OF PROJECT AND TASKS

The competition will be comprised of four three-hour modules over the course of two days. Some additional detail for each module is provided below.

2.1 Module A: Web Design (3-hours)

The first module will test your web design skills. You will need to develop a multi-page, responsive website from scratch with provided images. Your website will need to have common elements including a functional menu, header and footer areas. You can use a CSS framework such as Bootstrap for your layout design. You will also need to create page content elements that may include drop down menus, HTML forms, and tables.

For practice, you should work on how to manipulate images using an application such as Photoshop or GIMP (<https://www.gimp.org/downloads/>) to be able to crop them or alter their colour (e.g. convert to greyscale). You should also work on developing the HTML and CSS required to render a layout for your site. These sites must be responsive so that they will render properly on a mobile device such as a smart phone as well as a full-size computer. The Bootstrap framework can help with this responsive design.

The web sites you build for practicing should include common elements. For example, all of your pages must have the same look and feel. Any menu elements should remain consistent on all of your pages. You may use PHP to share menu elements in your design. If you wish to use PHP, you may need to install a local server environment such as XAMPP (<https://www.apachefriends.org/>).

To work on the HTML and CSS you may want to use an editor that can help such as VSCode (<https://code.visualstudio.com/download>).

2.2 Module B: JavaScript Game (3-hours)

For this module, you will build a game using JavaScript. This will require you to build interactivity, animation and gameplay using JavaScript. You will be able to use a JavaScript framework such as jQuery to help with this interactivity.

To get ready for this module, you will want to practice how to respond to events including clicks, mouseover, and mouseout. You will want to be able to drag and drop objects and respond to these actions. You may also need to animate elements on the screen to simulate movement.

For practice, try building a game that has 36 randomly distributed puzzle pieces that you show the user. The user is required to move the puzzle pieces into their correct location in an 6x6 grid. You can show a counter of the number of pieces that have been correctly placed and another counter that shows the number of mistakes that the user makes. You can provide the user feedback when they have successfully placed all the pieces. Consider how you may restrict the number of mistakes to 10 and provide feedback when they make that number of mistakes. What happens when they correctly place all pieces? Also, add a 'new game' button that will reset the game at any point they want to restart.

2.3 Module C: Speed Tasks (2 x 3-hours)

This module will be comprised of a series of mini tasks. These tasks will test your skills in several web technologies including JavaScript, HTML, CSS, image manipulation, Databases, and PHP development.

The tasks will be of varying length and complexity. Some may be 20-minutes while others could take a full hour to complete. Read through all the tasks and tackle those you feel most comfortable with first.

Below are three examples of the tasks you can expect. During the actual competition, there will be more of these types of tasks.

TASK 1: Create a blog database (20-min)

For this task you will create a database to be the home of a Blog. Your database will need to contain at least three tables – users, posts and comments.

Your tables need to meet the following requirements:

- Each table should have a unique identifier to be the primary key for the data.

- The user table should have a last name, a first name, a username and a password. There should be a restriction on the table so that the same username cannot be used more than once.
- The post table should have a subject, post body (for the text of the post), status field and posted date. In addition, there should be a field to tie the post back to the user.
- The comment table should have a comment field and date. In addition, there should be two fields to tie a comment to a user and to a post.

The relationship between tables should also be set so that when data is removed (for example a user) that any posts and/or comments they have tied to them are also deleted.

To work on this on your local computer, you may need to install a local server environment such as XAMPP (<https://www.apachefriends.org/>).

TASK 2: Use jQuery to create like/dislike buttons (40-min)

You will be provided with a webpage that has several fake user comments. There is a “thumbs up” and “thumbs down” button associated with each comment, but there is no functionality associated with the buttons.

Your task is to make these buttons functional. There are several requirements for the buttons:

- A user can only have one button selected for each comment (either the “thumbs up” or “thumbs down”, but not both)
- A user can click back and forth between each button.
- The user should be provided feedback for which button is selected.

In addition to the buttons associated with each comment, there is a button at the bottom of the page called “check ratings”. At any time when this button is clicked, the user should be presented with the information for which button is selected for which comment (“thumbs up” or “thumbs down”).

2.3.1 TASK 3: Write a secure login (60-min)

As a web developer, security is a key consideration when developing scripts. In this task, you must write a simple login system for a web site using PHP. Pages in your web site must ensure that there is a valid login or redirect to the login page if there is not. You will need to use sessions for this task. You need to create at least two separate

pages that perform a login check. Your login system must accommodate for the following:

- Passwords must not be stored in plain text
- If you use cookies, they cannot contain information about the user
- You should accommodate for SQL injection attacks
- You should accommodate for brute force attacks
- You should provide feedback to the user on bad logins

You have been provided with a basic user table for your database, but you will need to add fields to implement your updates. The create syntax for your basic table is

```
CREATE TABLE `users` (  
  `id` int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  `username` varchar(100) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  UNIQUE KEY `users_uid_unique` (`username`)  
)
```

To work on this on your local computer, you may need to install a local server environment such as XAMPP (<https://www.apachefriends.org/>).



WRITING



CREATIVITY & INNOVATION



PROBLEM SOLVING