



SAMPLE PROJECT
Web Technologies
TEAM CANADA

Sample Project : Day 1

INTRODUCTION

Day 1 is going to be comprised of two three-hour modules. The first module will test your skills with WordPress and will require you to know the template system and how plugins work. You will not be required to write a plugin for this module, but you should be comfortable with how to find and install WordPress plugins. This task will also require you to develop a look and feel for a website using HTML and CSS in the form of a WordPress template. Your web design skills will be put to the test here!

The second module will be developing a JavaScript based game. This will require using JavaScript to watch for certain events including clicking and possible dragging elements. You will need to keep score and provide feedback to the user. A sample game is provided below for you to practice on.

MODULE A: WordPress (3 hours)

Module A will be a three-hour project where you will need to develop a WordPress template design. You should familiarize yourself with the BlankSlate theme and how to build a theme based on that. You can use a CSS framework such as Bootstrap for your layout as that will help with your responsive design.

For practice, create a website design for a restaurant. Consider the look and feel of your website along with the menu and page layout. What sort of WordPress plugins might help provide functionality for your web site? Your web site must work on a mobile device such as a smart phone as well as a full-size computer. That means your web site must use a responsive design.

MODULE B: Tile Matching Module (3 hours)

Module B is going to be a JavaScript module where you will build a game. This will require you to build interactivity using JavaScript. You will be able to use a JavaScript Framework such as jQuery to help with this interactivity.

You will want to practice how to respond to click events and how to manipulate elements as a result of these events.

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.

Sample Project : Day 2

INTRODUCTION

Day 2 is going to 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 1h20m 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, comments and posts.

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”).

TASK 3: Write a secure (60-min)

As a web developer, security is a key consideration when writing. 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) NOT NULL,  
  `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/>).