

UNIVERSITY OF TORONTO  
Faculty of Arts and Science

APRIL 2016 EXAMINATIONS

CSC309H1 S - Programming on the Web  
Instructor: Ahmed Shah Mashiyat

Duration – 2 hours

Aid Sheet: Both side of one 8.5 x 11" sheet allowed (printed or handwritten)

Pass Mark: 14 out of 35

This exam is worth 35% of your final mark. You must get 14 out of 35 in this final exam to pass this course. Please answer all questions in the space provided. If you require additional space, please use the back of the page and indicate with an arrow that you have done so. For those questions that involve writing code, demonstrating a clear understanding of the concepts is important, but you should also make an effort to write clean and valid code.

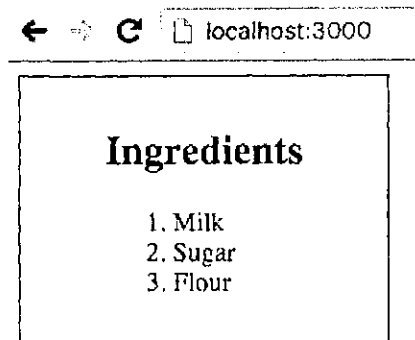
**Good luck!**

Last Name	
First Name	
Student Number	

**Marks**

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
3	4	3	7	2	2	4	2	3	5	35

Q1.- [3 points] Fill in the code in *index.html* and *style.css* to match the design below.



```
index.html
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <div class="container">
      <!-- Your HTML code here -->
```

```
    </div>
  </body>
</html>
```

```
style.css
.container {
  <!-- Your CSS here -->
```

```
}
```

Q2.- [4 points] Draw how the below HTML code will appear on a contemporary browser.

**style.css**

```
.container {
  display: inline-block;
  width: 45%;
  height: 100%;
  padding: 10px;
}

.p1 {
  text-align: right;
  border-right: 1px solid black;
}

.p2 {
  text-align: left;
  border-left: 1px solid black;
}
```

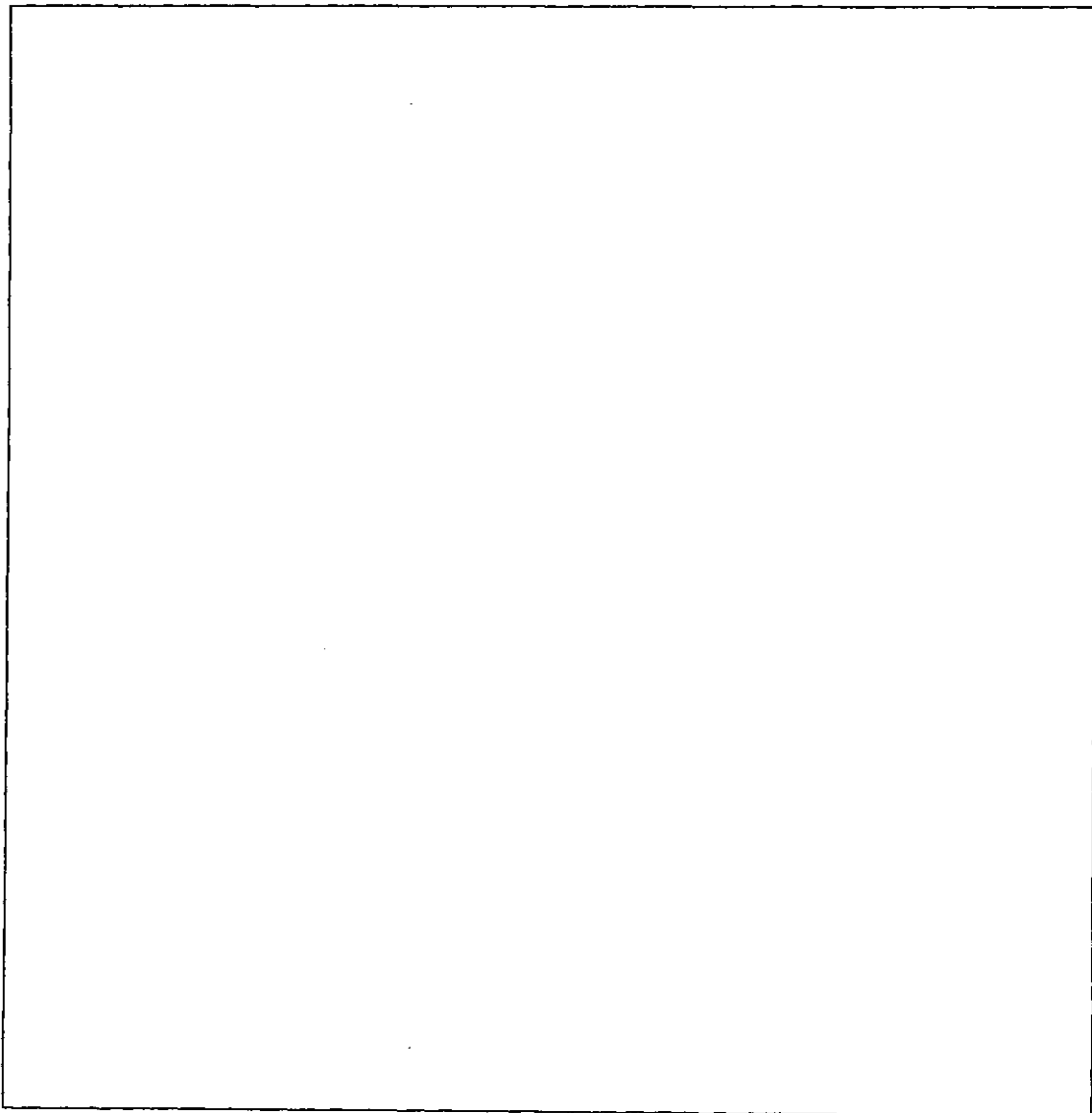
**index.html**

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body align="center">
  <h1>Duel Arena</h1>
  <hr>

  <div class="container p1">
    <h2>Player 1</h2>
    <form name="p1-form" action="submit.php">
      Nickname:<br>
      <input type="text" placeholder="Nickname"/><br><br>
      Class:<br>
      <input type="radio" name="class" value="gnome" checked> Gnome<br>
      <input type="radio" name="class" value="goblin"> Goblin<br><br>
      <input type="submit" value="Submit">
    </form>
  </div>

  <div class="container p2">
    <h2>Player 2</h2>
```

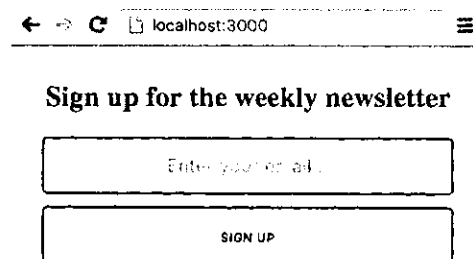
```
<form name="p2-form" action="submit.php">
  Nickname:<br>
  <input type="text" placeholder="Nickname"/><br><br>
  Class:<br>
  <input type="radio" name="class" value="gnome"> Gnome<br>
  <input type="radio" name="class" value="goblin" checked> Goblin<br><br>
  <input type="submit" value="Submit">
</form>
</div>
</body>
</html>
```



Q3.- [3 points] Consider the following webpage:



We want to re-structure the page when the screen width is equal or less than 600 pixels (e.g. on mobile devices) and make it look as follows:



You are given the HTML and CSS of the webpage

index.html

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>

<body align="center">
  
  <h2>Sign up for the weekly newsletter</h2>
  <div class="container">
    <input id="input-email" type="email" placeholder="Enter
your email..."/>
    <button id="btn-submit">SIGN UP</button>
  </div>
</body>
</html>
```

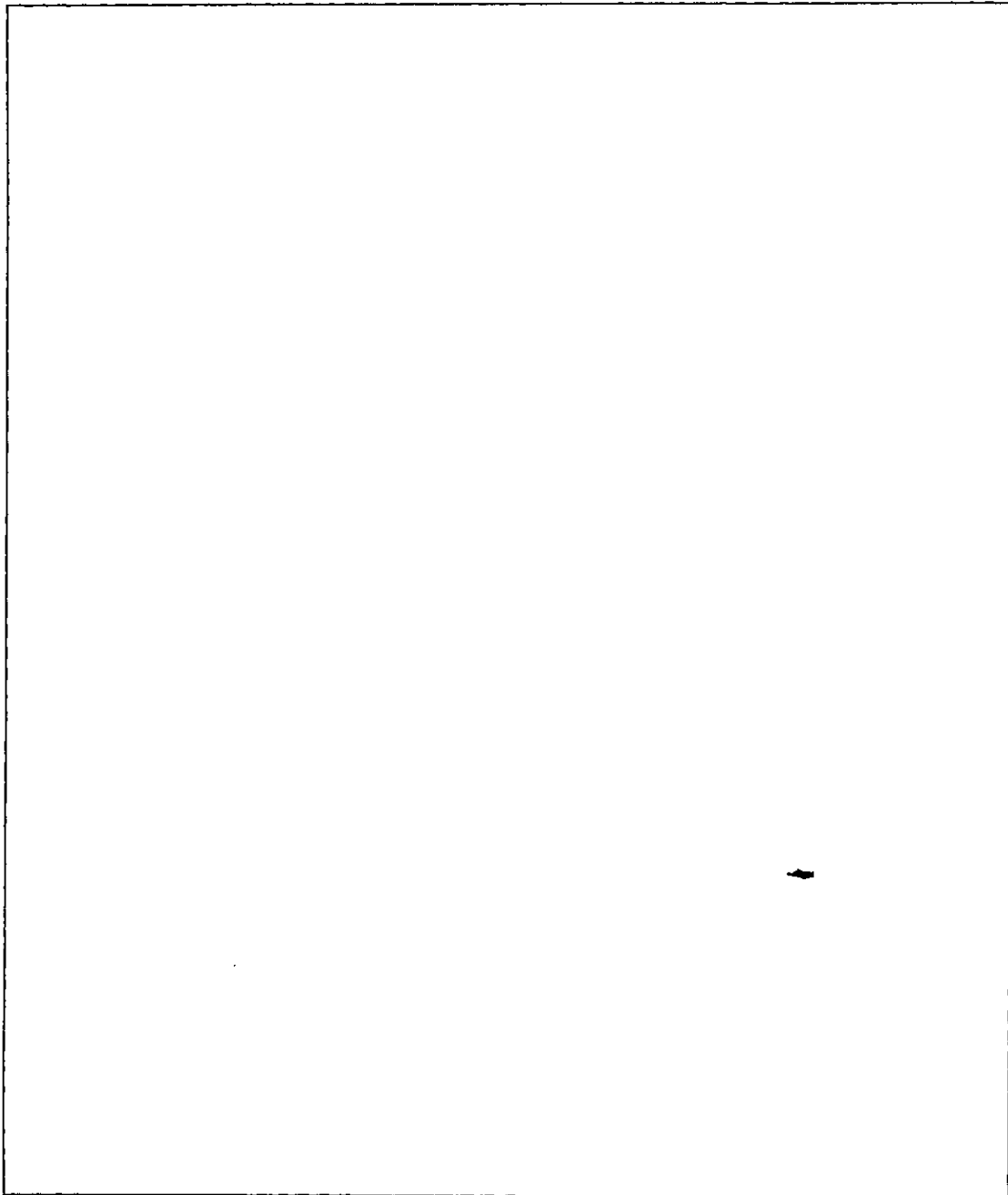
style.css

```
#input-email {
  width: 75%;
  height: 50px;
  font-size: 1em;
  border-radius: 5px;
  padding: 4px 16px;
  border-color: black;
  margin-bottom: 10px;
}

#btn-submit {
  height: 50px;
  padding: 18px 16px;
  border-radius: 5px;
  background: none;
  border-color: black;
  font-weight: bold;
}
```

```
#logo {  
  width: 200px;  
  height: auto;  
}
```

Write the CSS media query that yields the desired result for screens which are equal or less than 600 pixels wide.



Q4 to Q7. You are given the following HTML page (the styling is not of the essence.)

## Welcome

**Username:**

Enter Username...

**Password:**

Enter Password...

LOG IN

1. <html>
2. <head>
3. <script src="csc309.js"></script>
4. <script src="libs/jquery-1.12.0.min.js"></script>
5. <link rel="stylesheet" href="style.min.css">
6. </head>
7. <body>
8. <div id="login-box">
9. <h2 id="welcome-msg" align="center">Welcome</h2>
10. <label for="username">Username: </label>
11. <input class="form-control" type="text" id="username" placeholder="Enter Username...">
12. <label for="password">Password: </label>
13. <input class="form-control" type="password" placeholder="Enter Password...">
14. <button class="btn btn-primary" onclick="login()">LOG IN</button>
15. </div>
16. </body>
17. </html>

You are also given the following API endpoints and documentation by the (fictional) domain **csc309.com**:

**POST** /login

**GET** /users/:id/friends

### API Documentation

**POST** /login

Description: Authorizes a user.

Body parameters example:

```
{
  username: "Gandalf",
  password: "y0ush4lln0tp4ss"
}
```

Response Example (Success – HTTP 200):

```
{
```

```
status: success,  
user_id: 12233445,  
user_name: "Gandalf",  
auth_token: "bad18eba1ff45jk7858b8ae88a77fa30"  
}
```

Response Example (Error - HTTP 401):

```
{  
  status: error,  
  error: "Unauthorized"  
}
```

**GET** /users/:id/friends

Description: Returns a list of all the friends of a specific user.

Parameters:

*id:* User's unique identifier.

Example Request:

GET <http://csc309.com/users/12233445/friends>

Example Response (Success - HTTP 200):

```
{  
  status: success,  
  friends: [  
    "Frodo Baggins",  
    "Bilbo Baggins",  
    "Samwise Gamgee"  
  ]  
}
```

**Part A (AJAX & DOM Manipulation)**

Q4.- [7 points] Write Javascript code that supports the functionalities described below:

- Log-In user.
  - a. On a successful log-in attempt:
    - i. Hide the Login box (#login-box)
    - ii. Update the text in *h2* (#welcome-msg) so it reads "Welcome <username>" (e.g. Welcome Gandalf)
    - iii. Request and display the user's friends list (just the names) as returned from the server (styling is not of the essence.)
  - b. On a failed log-in attempt:
    - i. Display below the "Log In" button the following message: *"Incorrect username or password."*

The changes should be visible on the webpage without the need to refresh the page (use AJAX requests.) You can use jQuery as you see suitable. Assume you are writing the code in **csc309.js** which is included in the webpage.





**Part B (HTTP, Security, Performance)**

Q5.- [2 points] You used GET and POST requests in Part A. Name two other common HTTP methods and describe briefly when each would be used in the context of a web service.

Q6.- [2 points] On a very similar webpage as in Part A, you happened to enter the following in the Username input box: `<script>alert("Hello.")</script>`. An alert box pops up that reads "Hello."

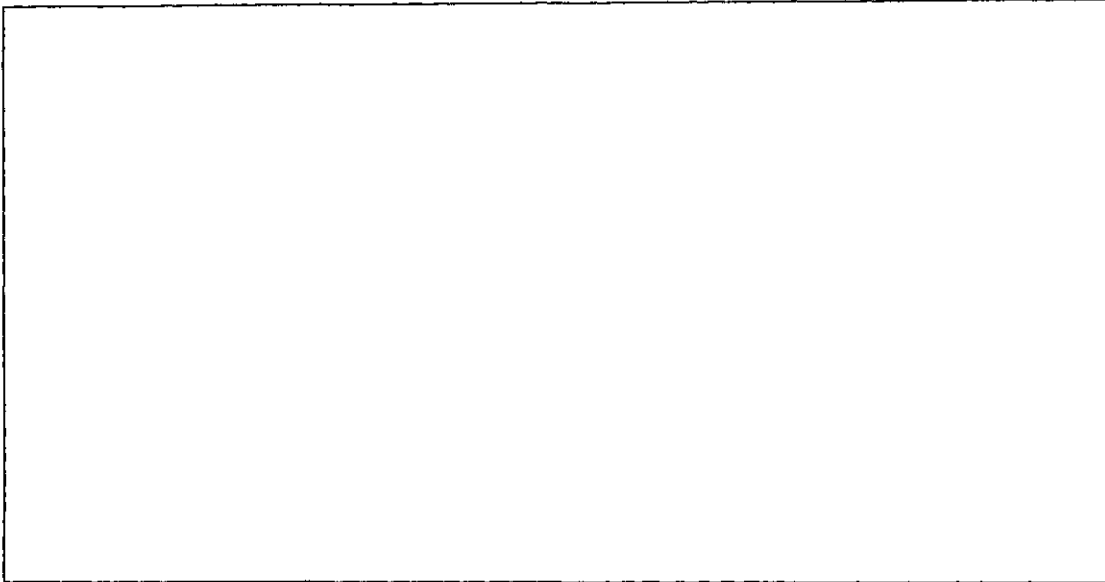
What's the name of that security vulnerability? State one way to fix it.

Q7.- [4 points] "PageSpeed Insights" is a tool that analyzes the content of a web page, then generates suggestions to make that page faster. We used that tool to analyze the web page shown at Part A. We received the following suggestions:

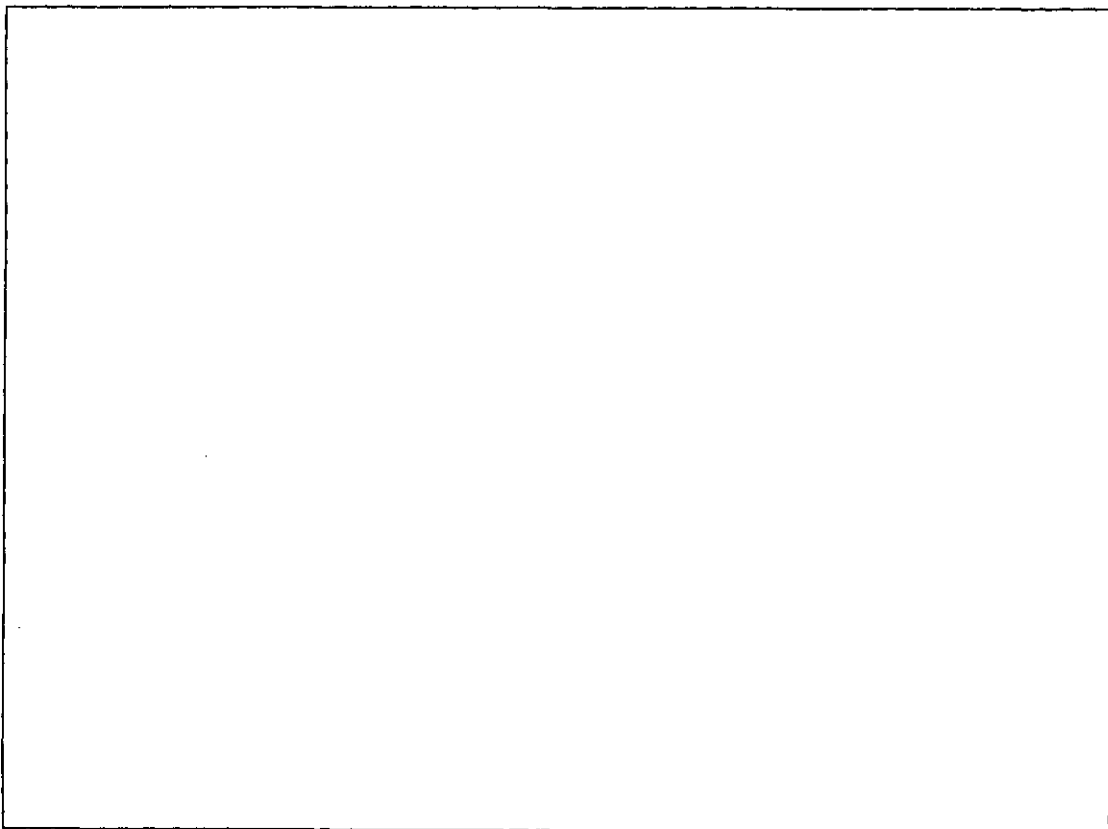
- i. Eliminate render-blocking JavaScript and CSS in above-the-fold content
  - ii. Minify JavaScript
- a) Which are the line numbers that point i. refers to? Why are those lines characterized as "render-blocking"? How would you proceed to fix this?
- b) What does minification do and how does it improve the webpage's performance?



Q8.- [2 points] What is screen Reflow? Why it is important?

A large, empty rectangular box with a thin black border, intended for the student to write their answer to question Q8.

Q9.- [3 points] What is Nodejs Event loop? Explain with a diagram.

A large, empty rectangular box with a thin black border, intended for the student to write their answer to question Q9, including a diagram.

Q10.- [5 points] Please clearly write your answer in the empty cell as TRUE or FALSE.

Marking scheme:

- +1: for correct answer
- -1: for incorrect answer
- 0: for not answering

The minimum mark for this assignment is 0 (never below).

1		XML has lighter weight data structure than JSON.
2		By default, Mocha will run 'UnitTest.js' in the root folder.
3		Ajax calls are faster than synchronous calls.
4		"history.get(-4)" will return a url that we have visited before.
5		We can use secret token to eliminate XSRF attack.

Scratch paper

Scratch paper