HENRY HEFFERNAN

Introduction to ITWS

Quiz 2: March 21, 2022

Place your name on the top of this document in the header

Enter your answers directly into this document (unless instructed otherwise)

All answers should be in be in Your Own Words, and use proper grammar

There are 4 questions on this test. Make sure you complete them all.

Make sure your answers use an alternative font and/or color – (not black or red)

Create a branch for this quiz called quiz2

Create a folder, somewhere under the root of your website (iit) for this quiz called quiz2

Save this document into that folder as *yourName*-*yourRCSID-S22*Quiz2.docx

Create a readme file in the same folder and discuss any relevant information about the lab;

(Include at least; your GitHub id, Repo name, and Discord handle.)

Place all quiz other specific documents (if any) in the same folder

Commit your changes as instructed below and push to GitHub – do NOT merge the branch to you will get a 0 for this quiz

Submit a link to your repo to LMS

NOTE: You are not to discuss this quiz with anyone. You are not to reference old (previous semester) submissions for ‘help’ or guidance. You may not solicit or receive help online or in-person. You may reference online resources, and you may use the notes from this class, but all work must be your own and you must figure out the solutions on your own.

1. Technology (coding): (50 points, 40min)
   1. You will be making changes to your website
   2. Create an issue indicating that Quiz 2 is taking place.
   3. Check out the quiz2 branch from your iit repo

You should make stages/commits to this branch – DO NOT merge it back or you will receive no credit for this quiz.

* 1. Forms
     1. Add a login form to the main page of your website containing a UserID, Password and Button that says ‘login’ – ex.



* + 1. (The form may be on the main page, in a modal, or on a separate page – your choice.)
    2. It should not break any of the ‘rules’ covered in lab 5
    3. When the ‘Login’ button is clicked, you must validate the form using JavaScript
    4. To validate successfully, the userID must match your RCSID and the password must match the word ‘password’.
    5. You only need to validate for non-blank fields
    6. When the form is submitted, the form should issue a GET request back to the index.html page.
    7. After validation, post a message to the screen in the header or footer that says whether the user is logged in or not logged in
    8. Using jQuery, make the message green if validation passes, and red if not.
    9. The JavaScript for the form must be in a separate document in your quiz2 folder
  1. Using external fonts from Google – make sure your Header and Menu and Footer use one unique font, and your text uses another.
  2. Close the issue on GitHub with a detailed comment
  3. Make sure your readme is descriptive and styled (Headings, etc… Nothing fancy)

1. Technology (description) (10 points, 10 min): Web Development
   1. I want to use jQuery for my website, but I don’t want to use the jQuery call and a company API already uses the ‘$’ to call its functions. What can I do? Show an example.

You can import jquery using the following syntax: import \* as jquery from ‘/path/to/jquery’. The important thing to note here is that the as keyword lets you import jquery as any valid variable name.

* 1. I have written a website with 200,000 lines of HTML, CSS and JavaScript. How can I go about making my site more efficient for the end user? Describe in detail and give at least two example covered in class.

200,000 lines is a lot of code, and there are plenty of ways to speed up the delivery of html/css/javascript. First of all, a very simple technique and basically global technique is to minimize file sizes when making a build for our website. A lot of stuff we write in js and css files are in there to make it more readable for humans. A minifier will remove newlines, comments, and in some cases even shorted variable names across the board to shrink the file sizes down. This is one quick way we can speed up the site and make it more efficient for the end user. The second way is a bit more involved, but it has to do with what we serve the user when they request to see our site. If we are statically serving all 200,000 lines of our code across multiple files, we will obviously have a long initial load time to download all that content. We can circumvent this by only serving specific files at specific endpoints and consolidate our code down that way. This would require a server to know where to serve what, but if used correctly would speed up the experience of an end user.

1. HCI - Website mockups (20 points, 15 min)
   1. Create/choose a persona for your website and create a user story – including your new ‘login’ function. Describe it below. If you like, but not required, include any mockups that might help tell your story.

Persona:

Name: Geoffrey S. Smith

Age: 26

Location: NYC

Occupation: Recruiter for Netflix

User Story: Geoffrey is a recruiter that is going through job applications sent into a role for a frontend engineer at Netflix. He is looking for someone with a very specific set of skills and after limiting down the initial pool of applicants based on resume, is now looking at Cover letters, personal websites, etc for each applicant. Geoffrey has just gotten back to my application and notices that I have linked my iit website, which was submitted by me in the initial job application. Geoffrey clicks the link and starts using the website. Liking what he is seeing, he notices a login form at the top of the page. He goes to login and realizes he doesn’t have an account. Confused, he searches for a place on the site to create an account but cannot find it anywhere. Geoffrey likes the website design and is impressed by the content, but is ultimately dissuaded by the limited login functionality, and decides to no longer consider me for the position he is recruiting for. Geoffrey continues to go through more applications and forgets he ever came across my website.

1. Web Science (20 points, 15 min)
   1. Social Problem
      1. Using references from the in-class lecture, identify a social problem and propose an engineered solution. What could you do to offer a solution?

Social Problem: Not knowing where your friends are when out on weekends.

Engineered solution: FriendTracker. Have you ever been out in the town on a night and end up separated from your friends, well FriendTracker has you covered! FriendTracker is a opt in tracking service that will allow you and your friends to know where each other are on your nights out. It’s very simple to set up: Simply download the app, add friends, and select timeslots when you want to be findable by your friends. Friends added on the app will be able to see each other during the timeslots provided and will save the headache of not being able to reach someone because they can’t hear their phone while out and about. FriendTracker can also double as a safety measure, notifying friends when they leave somewhere abruptly or are taken somewhere unknown. I would be able to create the app for iOS or android and it would simply request access for a user’s location even if the app is closed, but only ever send location data during timeslots the user gives permission for. I think the potential of such an idea is very high and could be a great thing for anyone to use to feel safer on nights out especially in areas they are unfamiliar with.