1. What is the most influential book or blog post you've read regarding web development?

   Cats who code http://www.catswhocode.com/blog/ David Walsh https://davidwalsh.name I follow these blogs regularly, which keeps me up-to-date with the changing trends in web development industry.

2. Tell me about a web application you have built. Why did you choose to build it? What did you learn? What challenges did you face and how did you overcome them?

   I developed a Book Catalog Application as part of my Full Stack Web Developer Nanodegree course. It displays genres of books to choose from and displays information about a selected book or displays available books in a selected genre. A user can login to the app using Google SignIn. While building this app I got a chance to learn and work on new frameworks and technologies. I learned to develop a RESTful web application using the Python framework Flask along with implementing third-party OAuth authentication. I learned to deploy a application on cloud infrastructure using Heroku and AWS. During this app development, I utilized multiple technologies, which are new to me. While working on this project I did face many roadblocks when experimenting with new technologies. With the help from some mentors and reading through multiple articles I easily overcame all the hurdles. With the skills learned through developing variety of projects, I am confident that I would be a great asset to your organization.

3. Write a function in Python that takes a list of strings and returns a single string that is an HTML unordered list (< ul>...</ul>) of those strings. You should include a brief explanation of your code. Then, what would you have to consider if the original list was provided by user input?

```
import cgi


input_list = ['Red','Orange','Yellow','Green','Blue']   # List of strings

#input_list = input().split(',')

def unescape(s):

        s = s.replace("&lt;", "<")

        s = s.replace("&gt;", ">")

        s = s.replace("&amp;", "&")

        return s


def htmlString(input_list):
        final_string = '<ul>'+'\n'

        for x in range(0,len(input_list)):

                final_string += '<li>'+input_list[x]+'</li>'+'\n'

        final_string +='</ul>'

        return cgi.escape(final_string)


print(unescape(htmlString(input_list)))


# Initially, I took the list of strings in a input_list variable

# Final String should contain the unordered list of given strings.

# So, first I initiated the final_string variable with opening tag of ul

# Then for each string in the input_list, final_string is concatenated with

#<li>, string value and </li>

# Finally, final_string is concatenated with closing tag </ul>


# If the input_list is provided by user when running the program:

# Assuming input is given as a list of values separated by commas

# I would read the input entered by the user and then split the input

# whenever there is a comma and store it as a list.


# input_list = input().split(',')


# To prevent certaian types of XSS attacks, I used html encoding using python's

# cgi.escape
```

4. List 3 attacks that web applications are vulnerable to. How do these attacks work? How can we prevent those attacks?

   SQL Injection Attackers gain control of databases and all of the information contained within them. This could expose any information that a web application stores in its database, which might contain users sensitive data. An attacker could submit malicious input in such a way that would alter the SQL statements being executed by the database server. SQL Injections can be prevented by multiple techniques such as sanitizing input data, using escape routines to handle special input characters. Cookie Poisoning

   Many web applications use cookies to save user information such as logins, passwords, and account emails. Cookie poisoning allows the attacker to modify the valid cookie and gain false authorization to information about another user. Information stored in the cookies should be encrypted using an industry-standard, well-tested algorithm. By encrypting the data, if a malicious user attempts to modify the value, the decryption process will fail. Cross-Site Scripting(XSS) Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a web application. XSS occurs when a web application makes use of invalidated or unencoded user input within the output it generates. The most common and useful XSS prevention mechanisms are Encoding and Validation. Encoding escapes the user input so that the browser interprets it only as data, not as code. Validation filters the user input so that the browser interprets it as code without malicious commands.

5. Here is some starter code for a Flask Web Application. Expand on that and include a route that simulates rolling two dice and returns the result in JSON. You should include a brief explanation of your code.

```
from flask import Flask

import json

import random


app = Flask(__name__)  # Initializes the application


# Result of the app (i.e., json data) can be accessed through

# following urls from the root url

@app.route('/')

@app.route('/json')

def Dice():
    # Following code generates two random number between 1 to 6
    dice1 = random.randint(1, 6)
    dice2 = random.randint(1, 6)
    result = dice1 + dice2
    # Data is provided in json format through json.dumps
    return json.dumps({
                'Dice1': dice1,
                 'Dice2': dice2,
                 'result': result
              }, sort_keys=True, indent=4, separators=(',', ': '))


# Runs the application
if __name__ == '__main__':
    app.debug = True
    app.run()
```

6. If you were to start your full-stack developer position today, what would be your goals a year from now?

I am a curious learner and would like to learn more frameworks and stay up to date with current trends in the web development, which could help developing applications more efficiently. My goals for a year from now would be mastering web development using Microsoft .NET framework stack. After one year, I would like to focus more on front end technologies such as ReactJS/Redux, Web optimization and building scalable applications. I would like to contribute to at least one open source project.

I am excited to apply for Full Stack Software Engineer job posted by Capital One. My core values, such as constant learning for achieving excellence, doing the right thing and innovative thinking, matches key values of Capital one.

In the long term, I would like to lead web development teams. By working on variety of web frameworks, front end technologies, I can prepare myself for achieving my long term goal.

7. Mentions a specific full stack technology and/or trend. Explains how it works. Discusses advantages and disadvantages of it. Compares technology/trend to alternatives or discusses technology/trend in context

   Containers are replacing VMs. Containers are isolated, share OS with host and will hold all libraries/dependencies needed to run the application. This will provide portability among servers, can be run on developer's laptops and on servers. Using Containers will speed up web development by reducing time taken to deploy and to test applications. Containers will share host OS and can be run on Linux or Windows. But, Virtual Machines run guest OS that needs to be compatible with host OS.

   Size of container in bytes in much smaller than VMs. So, containers start much faster.