**IST105 - Introduction to Programming  
Assignment #7**

*Assignment #7 - Python Logic, Bit Operations, and Deployment with Git, GitHub, and AWS Load Balancer*

**Important Note:**

Assignments that do not fulfill all the requirements or contain errors preventing successful execution on the AWS EC2 instance with a load balancer will receive a grade of zero. Ensure your repository is complete, functional, and publicly accessible before submission. Thoroughly test your application to catch and fix issues before finalizing the assignment.

### **Prerequisites:**

* Basic knowledge of Python loops, lists, and bit operations.
* Familiarity with Git and GitHub commands for branching and merging.
* Access to AWS Academy for deploying applications.

**Objective:**

In this assignment, you will develop a Python application that uses logic and bit operations to solve a practical problem.

The application will:

* Process a list of integers using loops.
* Perform specific bitwise operations (e.g., AND, OR, XOR).
* Output results based on user input.  
  You will also utilize Git and GitHub for version control, create multiple branches, and deploy the main branch to an AWS EC2 instance configured with a load balancer for scalability.

**Assignment Tasks:**

**1. Python Application Development:**

**Create the User Input Form:**

* Write a PHP script named **form.php** that includes a form where users can input a list of integers.
* The form should have:
* 1 Input fields where users can input a series of integers separated by commas.
* A submit button labeled "Submit".
* Alternatively, users can send values directly via URL parameters.

**Create the Python Script:**

Create a Python script **bitwise\_operations.py** that processes a list of integers and performs the following operations:

* **Input Validation:** Retrieves the values from user input (either from the form or URL).
* Validate the input to ensure it contains only integers.
* **Bitwise Operations:**
  + Calculate the AND, OR, and XOR of all integers in the list.
  + Output the results of each operation.
* **Filtering with Loops:**
  + Use a loop to filter and display all numbers greater than a user-specified threshold.

**Create the PHP Script to Process Input:**

* Write a PHP script named **process.php** that:
* Receives the user input from **form.php** or URL parameters.
* Calls **bitwise\_operations.py** to process the input and capture the output.

**Example Input/Output:**

**Integers separated by commas: 3, 5, 7, 9**

**Threshold: 4**

**Bitwise AND: 1**

**Bitwise OR: 15**

**Bitwise XOR: 14**

**Numbers greater than threshold: [5, 7, 9] [5, 7, 9]**

1. **Create two EC2 Instance:**
   * Launch two new EC2 instance using the Amazon Linux AMI.
   * Configure security groups to allow HTTP (port 80) and SSH (port 22) access.
   * Note the public IP address of your instance.

**Install Required Software:**

* 1. Install Apache web server, PHP, and Python.
  2. Start and enable the Apache service.

**3. GitHub Repository and Branching:**

* **Create a GitHub Repository:**

Create a new public GitHub repository and push your local repository to GitHub.

* **Create Branches:**

Set up the following branches in your GitHub repository:

* + **main** - This branch should contain the final, tested version of the application.
  + **development** - For integrating changes and testing before merging into the main branch.
  + **feature1** - For implementing the bitwise operations feature.

**4. Code Modifications on Branches:**

* **Update Code on feature1 Branch:**

Make incremental updates to **bitwise\_operations.py**

Add comments or minor modifications in feature1 to explain your use of Git and branching.

* **Merge Changes to development:**

After testing modifications in feature1, merge the branch into development and confirm that all functionality works as expected.

* **Merge Changes to main:**

After thorough testing in development, merge the code to the main branch to mark it as the final version.

**5. Deploy to EC2 Instance:**

* **Clone the GitHub Repository to EC2:**
  + Connect to your EC2 instance via SSH:

**ssh -i <your-key-pair.pem> ec2-user@<your-public-ip>**

* + Use git clone to retrieve your GitHub repository on the EC2 instance:

**git clone <your-github-repository-url>**

* + Change directory into your project folder (replace <repository-name> with the actual name of your GitHub repository):

**cd <repository-name>**

* **File Permissions:**

Ensure that the data\_management.py script has the correct permissions to be executed:

**sudo chmod +x /var/www/html/bitwise\_operations.py**

Test the script locally on each instance:

**python3 bitwise\_operations.py**

* **Start Apache Server:**

Start or restart the Apache server to ensure it’s running your updated code:

**sudo service httpd start**

**# or**

**sudo service httpd restart**

**6. Set Up a Load Balancer:**

* **Navigate to the EC2 Dashboard:** Access the AWS Management Console and open the EC2 service.
* **Create an Application Load Balancer:** Go to the Load Balancers section, and click Create Load Balancer. Select the Application Load Balancer type.
* **Add EC2 Instances to a Target Group:** During the configuration, create or select a target group. Add both previously created EC2 instances to the target group to ensure they are part of the load balancer's traffic distribution.
* **Configure Listener for HTTP Traffic:** Set up a listener to route HTTP traffic on port 80 to the target instances in the target group.

**7. Testing and Verification:**

* Test the script using the public DNS name of the load balancer.
* Verify that the load balancer distributes traffic evenly across both instances by running tests multiple times.

**Submission Requirements:**

* **GitHub Repository Link:**

Submit the public link to your GitHub repository, which should include the branches main, development, and feature1.

**[Insert your URL here]**

* **Screenshot of EC2 Compute Instance 1 - Web Application:**

Include a screenshot of the deployed web application showing the public IP address in the URL.

**[Insert your screenshot here]**

* **Screenshot of EC2 Compute Instance 2 - Web Application:**
* Include a screenshot of the deployed web application showing the public IP address in the URL.

**[Insert your screenshot here]**

* **Load Balancer Verification:**
* Include a screenshot showing the load balancer’s DNS name and successful traffic routing.

**[Insert your screenshot here]**

**Important Notes:**

* **Functionality:** Make sure your application is fully functional on the EC2 instance and accessible via the public IP address.
* **Testing:** Verify all features thoroughly before submission.

**Tips:**

* **Git Commands Recap:** Use git branch, git checkout, git merge, and git push commands effectively to manage branches.
* **Testing Before Merging:** Always test changes on the feature1 and development branches before merging to main.