**IST105 - Introduction to Programming**

**Assignment #6**

*Assignment #6 - Building a Data Management Application Using Lists, Logical Operators, and Bitwise Operations*

**Important Note:**

**Assignments with code that contains errors and does not run correctly will receive a grade of zero.**

**Please ensure that your code is fully functional and free of syntax or runtime errors before submitting. It is recommended that you thoroughly test your code in Python IDLE and fix any issues that arise.**

**Include comments where necessary to explain your logic and observations.**

**Prerequisite:**

* **AWS EC2 Setup:** Ensure you have access to AWS and know how to create an EC2 instance.
* **Software Installation:** Familiarize yourself with installing and configuring Python, PHP, and Apache on an EC2 instance.
* **Basic Programming Knowledge:** Understand Python lists, logical operators, and bitwise operations.

### **Objective:**

The goal of this assignment is to build a data management web application that collects user input, processes the data using lists and logical operations, and dynamically displays results using PHP and Python. This will enhance your understanding of web development, user input handling, and backend processing.

**Assignment Tasks:**

1. **Create an EC2 Instance:**
   * Launch a 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.
2. **Install Required Software:**
   * Install Apache web server, PHP, and Python.
   * Start and enable the Apache service.
3. **Create the User Input Form (Optional):**

* Write a PHP script named **form.php** that includes a form where users can input five numerical values.
* The form should have:
* Input fields for five numbers (a, b, c, d, e).
* A submit button labeled "Submit".
* Alternatively, users can send values directly via URL parameters.

1. **Create the Python Script:**

* Write a Python script named **data\_management.py** that:
  + Retrieves the values from user input (either from the form or URL).
  + Checks if all inputs are numeric; if not, return an error message.
  + Checks if any value is negative and displays a message if so.
  + Calculates the average of the numbers and checks if it’s greater than 50.
  + Uses bitwise operations to determine if the count of positive numbers is even or odd.
  + Creates a new list with values greater than 10, sorts it, and returns both the original and sorted lists formatted as HTML.

1. **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 data\_management.py to process the input and capture the output.
* Add labels and comments to the output using HTML for clarity and formatting. For example:

### echo "<h2>Results:</h2>";

### echo "<p>Original Values: " . implode(", ", $original\_values) . "</p>";

### echo "<p>Sorted Values: " . implode(", ", $sorted\_values) . "</p>";

1. **Deploy the Files:**

* Upload **form.php, data\_management.py, and process**.**php** to the /var/www/html/ directory on your EC2 instance.

1. **Configure Permissions:**

* Ensure the Python script has the correct permissions to be executed:

sudo chmod +x /var/www/html/data\_management.py

1. **Test the Setup:**

* Open your web browser and navigate to **http://<your-public-ip>/form.php** to see the user input form.
* Submit values to see the calculations and output displayed on the results page.

Submission Requirements:

1. **GitHub Repository:**
   * Create a GitHub repository and upload all the required files (form.php, data\_management.py, and process.php).
   * Use **three branches**:
     + **main:** The stable, production-ready version of your code.
     + **development:** The development branch where active work is being done.
     + **feature1:** A branch dedicated to implementing a new feature (e.g., validation improvements or new functionality).
2. **Files Required in the Repository:**
   * form.php (optional)
   * data\_management.py
   * process.php
3. **Important Notes:**
   * Ensure your repository contains all necessary files and that the .py file runs correctly.
   * If the repository is incomplete or contains errors in the code, your grade will be significantly reduced.
4. **Submission Process:**
   * Add here link to your GitHub repository via the assignment portal.

**[Here your GitHub repository]**

* + Include a **screenshot** showing the web page with the Python script running, including the public IP address of your AWS EC2 instance.

**[Here your screenshot]**

1. Upload the zip file to Canvas named as **Assignment6\_FirstNameLastName**.

**Tips:**

* Ensure all paths and permissions are set correctly to avoid access issues.
* Test your Python script and PHP script locally before deploying them on the EC2 instance.
* Double-check the public IP and ensure it is correctly displayed in the screenshot.
* Review the concepts covered in previous classes to help structure your program.