# Web Application Framework (WAF) - Fall 2025 Lab Task - 1 (17th September 2025)

## Part-1: File System Operations

1. Create a folder named "yourName_wafLab_task1_F25".
2. Navigate into the folder and create a NodeJS file named "file_ops.js".
3. In "file_ops.js", write code to programmatically create a subfolder named "storage" in the same directory as the .js file.
4. Write a code statement to create a file named "info.txt" inside the "storage" folder.
5. Write some script to append your full name in the "info.txt" file.
6. Write code to read the data from "info.txt" (your full name) and display it on the console.
7. Write code to overwrite the content in "info.txt" with only the first letter of your first name and the first letter of your last name (e.g., "John Doe" becomes "JD").
8. Repeat step 6 to display the updated content.
9. Clear all data from the "info.txt" file.
10. Repeat step 5 to write your full name again.

## Part-2: Domain Name Classification

Use Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel to generate 1000 instances of JSON data for first and last name, gender, email, IP addresses. Your task is to create four files, named as “com.txt”, “org.gov”, “edu.txt”, and “uk.txt”. Find instances from the email address field of the generated data having .com, .edu, .org, and .uk. Append the complete instance of record to the relevant corresponding created files.

## Part-3: Asynchronous Function Execution

Your program should define and call six functions named `register`, `sendWelcomeMessage`, `login`, `fetchProfile`, `updateStatus`, and `logout`. Use the asynchronous `setTimeout` function to introduce delays in each. Each function should output a console message indicating task completion. Call these functions in the following order with the specified delays, and the last line of code should be `console.log("All operations finished!")`.

- `register` = delay of 2500 milliseconds (2.5 seconds)

- `sendWelcomeMessage` = delay of 3000 milliseconds (3 seconds)
- `login` = delay of 2000 milliseconds (2 seconds)
- `fetchProfile` = delay of 4000 milliseconds (4 seconds)
- `updateStatus` = delay of 1500 milliseconds (1.5 seconds)
- `logout` = delay of 3500 milliseconds (3.5 seconds)

i. Call these functions in the above order with the given delays and observe the execution order. Note the sequence in which messages appear.

ii. Use callbacks to ensure the functions execute in the specified order. Call them again with the same delays and verify the order.

iii. Use Promises to achieve the same ordered execution as in part ii. Call the functions with the delays and confirm the sequence.

iv. Use async/await to accomplish the same as part ii or iii. Call the functions with the delays and verify the execution order.

## Part-4: IP Address Class Detection and JSON Data

Use Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel to generate 1000 instances of JSON data for first and last name, gender, email, IP addresses. Read the generated file containing JSON data. Identify the objects of class A, B, C, D and E IP addresses based on the subnet mask and store the data in five different files (IP_Class_A.txt, IP_Class_B.txt, IP_Class_C.txt, IP_Class_D.txt, IP_Class_E.txt) one each for a class of IP addresses.

## Notes

i. Refer to NodeJs official documentation as needed.
ii. Place each part code in separate files and place it in a folder with the name such as yourRollNo_yourName_Task1_F25. Compress the folder.
iii. Submit the above-compressed folder on fs to the folder WAF-F25-T1