**Task:**

The update\_task API initially sends notifications synchronously, causing execution delays. To improve response time, the blocking notification functionality can be handled asynchronously. This can be achieved using the Thread class from the threading library, which allows code to run in parallel with the main thread. By executing notifications asynchronously, the main thread can continue running and complete execution without waiting for other threads to finish. Depending on the requirement, the main thread can either wait for all created threads to complete or proceed without waiting, ensuring a more efficient execution flow.

**Changes in the code:**

1. Creating a method/function with the data as argument which uses "smptlib" Library to send the Email notifications.

2. In update\_task method, Thread Object is created with newly created method as target and data as arguments which in turn calls the notification method in a separate thread than the main thread.

3. By using the start() method provided for the Thread Object, it creates a new thread asynchronously and also calls run() method immediately to run the thread parallelly calling the sendnotification target method.

4. time.sleep(15) added in the sendnotification method at the end to simulate the time taken to complete the thread asynchronously.

5. Used the Postman to test the API.

**Code Observation (Error):**

While running the given assignment code, while running the code this line is below the new\_task variable assignment, it throws error as the 'next\_task\_id' is used first and then global keyword used. So, it throws error as it is considered as a local variable and then using the global keyword in the next line throws the error that it is used before the global declaration.