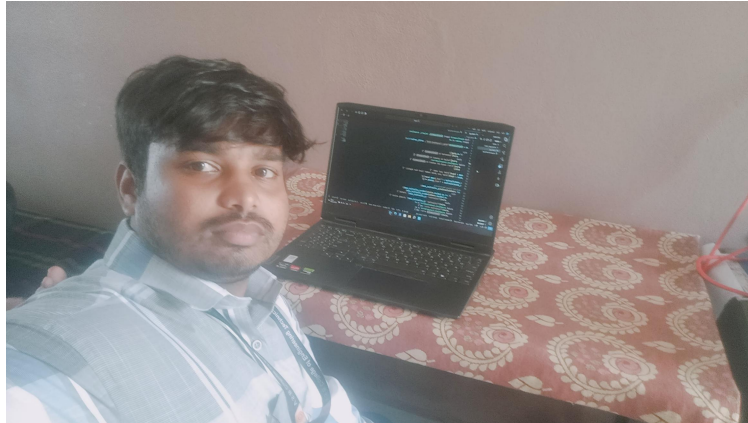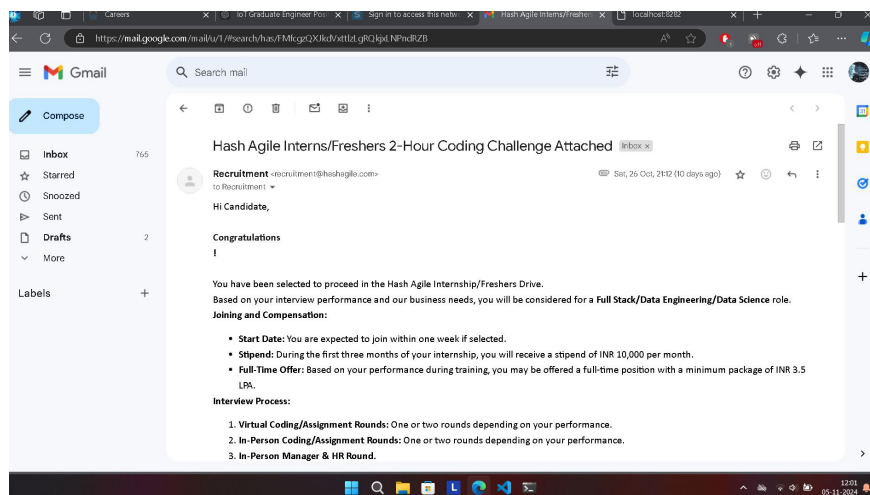# ROUND 2

Name: Sakthibala A
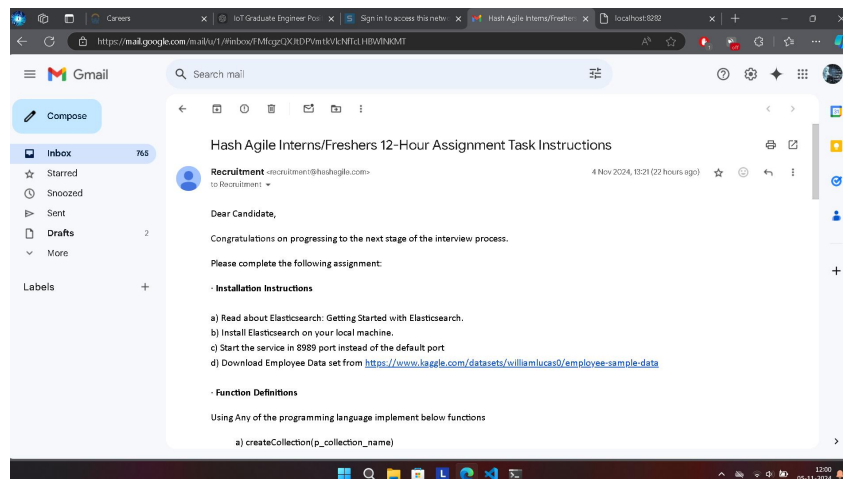**1. Selfie Pic:**

2. 1 st Task Email Screenshot:

3. 2 nd Task Email:

4. Github URL For Round 1: darkbala343/HASH_ROUND-_1

5. Github URL For Assesment: darkbala343/HASH_ROUND-2

6. Function Execution Results:

Service running on port 8989:
http://localhost:8989

Program:

```python
name = input("Enter your name: ")
phone = input("Enter your phone number (last four digits): ")

v_nameCollection = f'hash_{name.lower()}'
v_phoneCollection = f'hash_{phone[-4:]}'

# Qodo Gen: Options | Test this function
def createCollection(p_collection_name):
    try:
        if not es.indices.exists(index=p_collection_name):
            es.indices.create(index=p_collection_name)
            print(f"Collection '{p_collection_name}' created.")
        else:
            print(f"Collection '{p_collection_name}' already exists.")
    except exceptions.ConnectionError as e:
        print(f"Connection error: {e}")
    except exceptions.TransportError as e:
        print(f"Elasticsearch error: {e}")
```

```python
def getDepFacet(p_collection_name):
                        "field": "Department.keyword"
                    }
                }
            }
        }
        res = es.search(index=p_collection_name, body=query)
        print("Employee count grouped by department:")
        return res['aggregations']['departments']['buckets']
    except exceptions.ConnectionError as e:
        print(f"Connection error: {e}")

createCollection(v_nameCollection)
createCollection(v_phoneCollection)
getEmpCount(v_nameCollection)
indexData(v_nameCollection, 'Department')
indexData(v_phoneCollection, 'Gender')
getEmpCount(v_nameCollection)
delEmpById(v_nameCollection, 'E02003')
getEmpCount(v_nameCollection)
print(searchByColumn(v_nameCollection, 'Department', 'IT'))
print(searchByColumn(v_nameCollection, 'Gender', 'Male'))
print(searchByColumn(v_phoneCollection, 'Department', 'IT'))
print(getDepFacet(v_nameCollection))
print(getDepFacet(v_phoneCollection))
```