Dhirubhai Ambani Institute of Information and Communication Technology

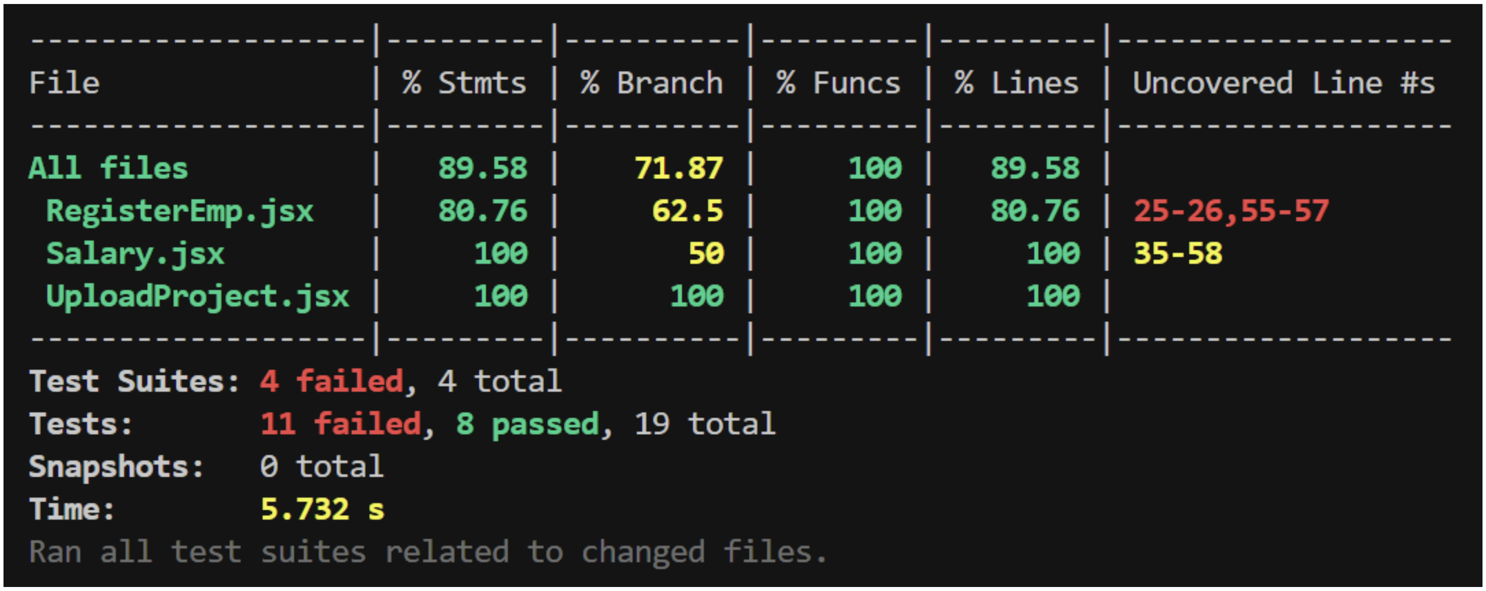
DA-IICT

IT314: Software Engineering

G17: StaffGrid

JEST for frontend

JEST in frontend code coverage:



```
-------------------|---------|----------|---------|---------|-------------------
File               | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-------------------|---------|----------|---------|---------|-------------------
All files          |   89.58 |    71.87 |     100 |   89.58 |
 RegisterEmp.jsx   |   80.76 |     62.5 |     100 |   80.76 | 25-26,55-57
 Salary.jsx        |     100 |       50 |     100 |     100 | 35-58
 UploadProject.jsx |     100 |      100 |     100 |     100 |
-------------------|---------|----------|---------|---------|-------------------
Test Suites: 4 failed, 4 total
Tests:       11 failed, 8 passed, 19 total
Snapshots:   0 total
Time:        5.732 s
Ran all test suites related to changed files.
```

As shown in the snapshot files that do not contain “axios” imports, it can have the best testing strategies. As for files that do contain “axios” imports have low testing strategies because, when testing React components that utilize “axios” to fetch or send data, the core challenge lies in reliably testing asynchronous operations and external dependencies. These can introduce complexities and uncertainties into your test suite.

Approach for not prioritizing “axios” containing files:

1. **Mocking External Dependencies:**

   - **Complexity:** Mocking ‘`axios`’ calls can be complex, especially when dealing with complex data structures and error handling.
   - **Maintenance Overhead:** As your application evolves, maintaining these mocks can become burdensome.

2. **Test Coverage Limitations:**

   - **Asynchronous Nature:** Asynchronous operations, especially those involving network requests, can be difficult to test comprehensively.
   - **External Factors:** Factors like network latency, server response times, and API changes can impact test reliability.

3. **Dependency on External Services:**

   - **Unreliable Environments:** Relying on external services can make tests flaky and unpredictable.
   - **Mock Data Limitations:** Mocked data might not always accurately represent real-world scenarios.