**Lab 2**

**Title: Introduction to Test-Driven Development (TDD) and Behavior-Driven Development (BDD)**

**Objective:**

To understand and apply the principles of Test-Driven Development (TDD) and Behavior-Driven Development (BDD) in a Node.js environment using Jest.

**Theory:**

Test-Driven Development (TDD) is a software development technique where tests are written before the actual implementation. The process follows the Red-Green-Refactor cycle:

1. Red – Write a failing test.
2. Green – Write minimal code to pass the test.
3. Refactor – Clean up code while keeping the test passing.

Behavior-Driven Development (BDD) is an evolution of TDD that encourages collaboration and describes tests in natural, human-readable language. BDD frameworks include Jest, Cucumber.

**Code**

### Step 1: Initialize a Node.js project

```
npm init -y
```

### Step 2: Install Jest testing library

```
npm install --save-dev jest
```

### Step 3: Configure test script in package.json

```
"scripts": { "test": "jest" }
```

### Step 4: Create a function file sum.js

```
function sum(a, b) {
  return a + b;
}
module.exports = sum;
```

### Step 5: Create a test file sum.test.js

```
 const sum = require('./sum');
 test('adds 2 + 3 to equal 5', () => {
 expect(sum(2, 3)).toBe(5);
});
```

### Step 6: Run the test

```
npm test
```

**Conclusion:**

In this lab, we practiced writing tests before implementation (TDD) and describing behaviors in test cases (BDD). Using Jest, we created unit tests that verify our function works correctly. This process ensures more reliable code and helps catch bugs early in development