

# **" Jest"**

## **Unit Testing Tool**



Prepared by:  
**Samia Alam ,Abdus Salam**

# Contents

<b>Contents</b>	<b>I</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Installation and Usage</b>	<b>2</b>
<b>3 Conclusion</b>	<b>6</b>
<b>Credits</b>	<b>7</b>

# 1 Introduction

## 1.1) What is Unit Testing Tool ?

Unit testing tools are software applications that help developers automate the testing of individual components or functions of a program. They allow for the isolation of specific sections of code to verify their correctness, ensuring that each unit performs as intended. These tools provide features like test case creation, execution, and reporting. Popular examples include Jest, Mocha, and JUnit.

## 1.2) What is Jest ?

Jest is a JavaScript testing framework developed by Facebook, primarily used for testing React applications. It provides a simple API for writing and running tests and supports features like mocking code coverage, and snapshot testing. Jest's built-in watch mode also allows developers to run tests automatically as they make changes to the code.

## 1.2) Key Features

- a) Zero Configuration: Jest requires minimal setup, allowing for quick integration into projects.
- b) Mocking Capabilities: It provides built-in functions to easily mock modules and functions for isolated testing.
- c) Snapshot Testing: Jest can capture the rendered output of components and track changes over time, helping to identify unexpected UI changes.

For more information check [here](https://jestjs.io/docs/getting-started)

<https://jestjs.io/docs/getting-started>

## 2 Installation and Usage

### 2.1 Installation:

1) Install Node.js if it's not already installed on your system. You can download it from the

<https://nodejs.org/en/download/package-manager>

2) Install jest using this command:

```
npm install --save-dev jest  
yarn add --dev jest  
pnpm add --save-dev jest
```

For better understanding click [here](#)

<https://www.youtube.com/watch?v=ZnIv8u2-XrA>

### 2.2 How jest Works:

Jest works by executing test files that contain specified test cases and assertions, comparing actual outputs against expected results. It uses a test runner to identify and run these tests, generating detailed reports on pass/fail statuses. During execution, Jest can automatically mock functions, modules, and components to isolate tests and ensure reliability.

## 2.3 A quick guide for using jest unit testing tool:

### Step 1: Environment Setup:

Install node.js if it is not installed

### Step 2: Create Your folder and go to the terminal and write this command :

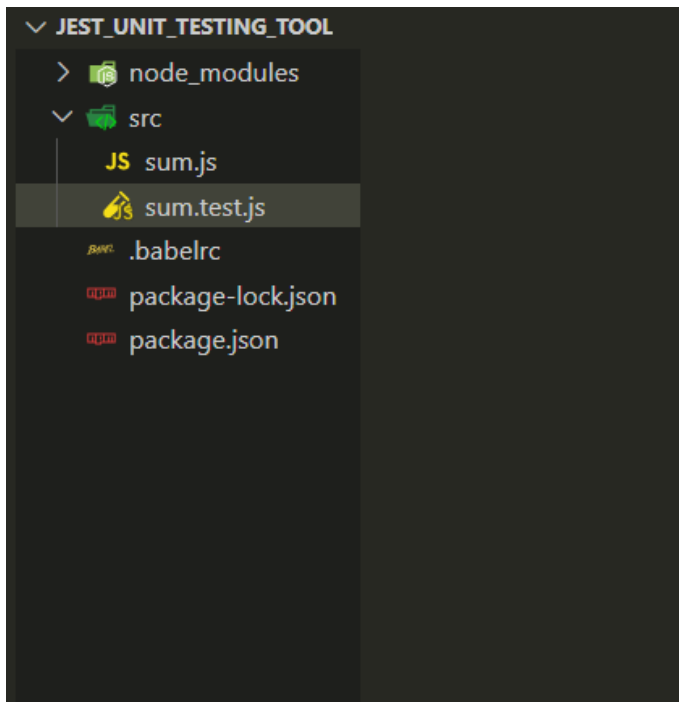
```
npm init
```

### Step 3: Add the following section to your package.json:

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



Step 3: Make a folder name src where your js main file and test file will remain :

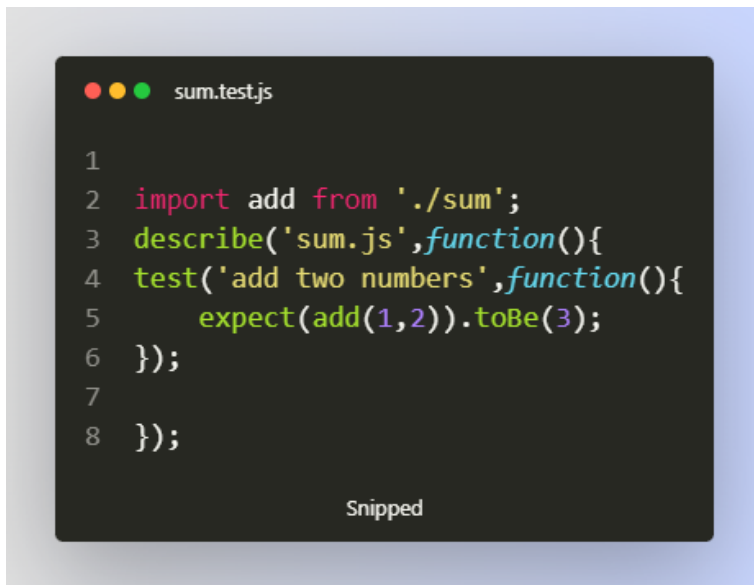


Step 4: Install the jest using this command:

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

Step 5: Make a JS file and a test file like below:





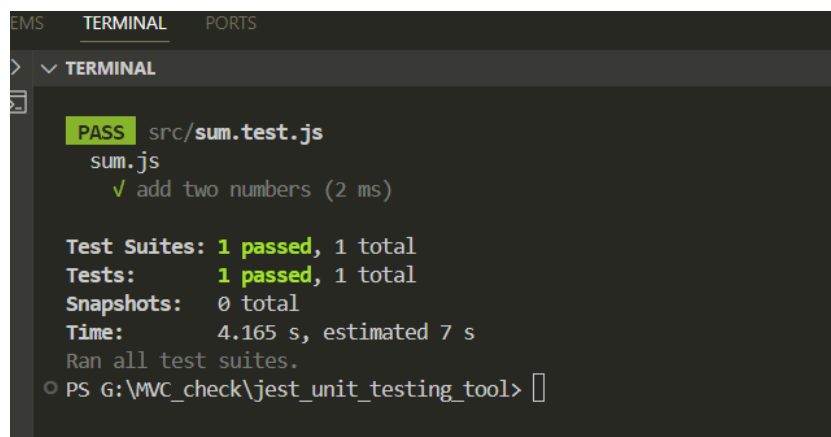
Step 6: Make a .babelrc file to recognize the file and add this code:

```
{
  "presets": ["@babel/preset-env"]
}
```

Step 7: After that execute these command:

```
npm install --save-dev babel-jest
npm install @babel/preset-env --save-dev
```

Step 7: After executing these all command the output like below:



## 3 Conclusion

### 3.2) Advantages

- **Easy Setup:** Jest requires minimal configuration, making it quick to integrate into projects, especially React applications.
- **Snapshot Testing:** It allows developers to capture the rendered output of components, making it easy to track UI changes over time.
- **Built-in Mocking:** Jest provides powerful mocking capabilities to isolate tests, helping to ensure that unit tests remain focused on specific functionality.
- **Rich Reporting:** Jest offers detailed test reports, including coverage statistics, which help identify untested parts of the codebase.
- **Active Community and Ecosystem:** As a popular framework, Jest has extensive documentation, community support, and a range of plugins that enhance its functionality.

### 3.3) Disadvantages

- **Performance with Large Codebases:** In very large projects, Jest can become slower compared to some other testing frameworks due to its extensive feature set.
- **Learning Curve for Advanced Features:** While basic usage is straightforward, mastering advanced features like custom matchers and configuration can be challenging for beginners.
- **Compatibility Issues:** Some legacy code or specific configurations may not work seamlessly with Jest, requiring additional setup or workarounds.



## Credits

<https://jestjs.io/docs/getting-started>

THANK YOU