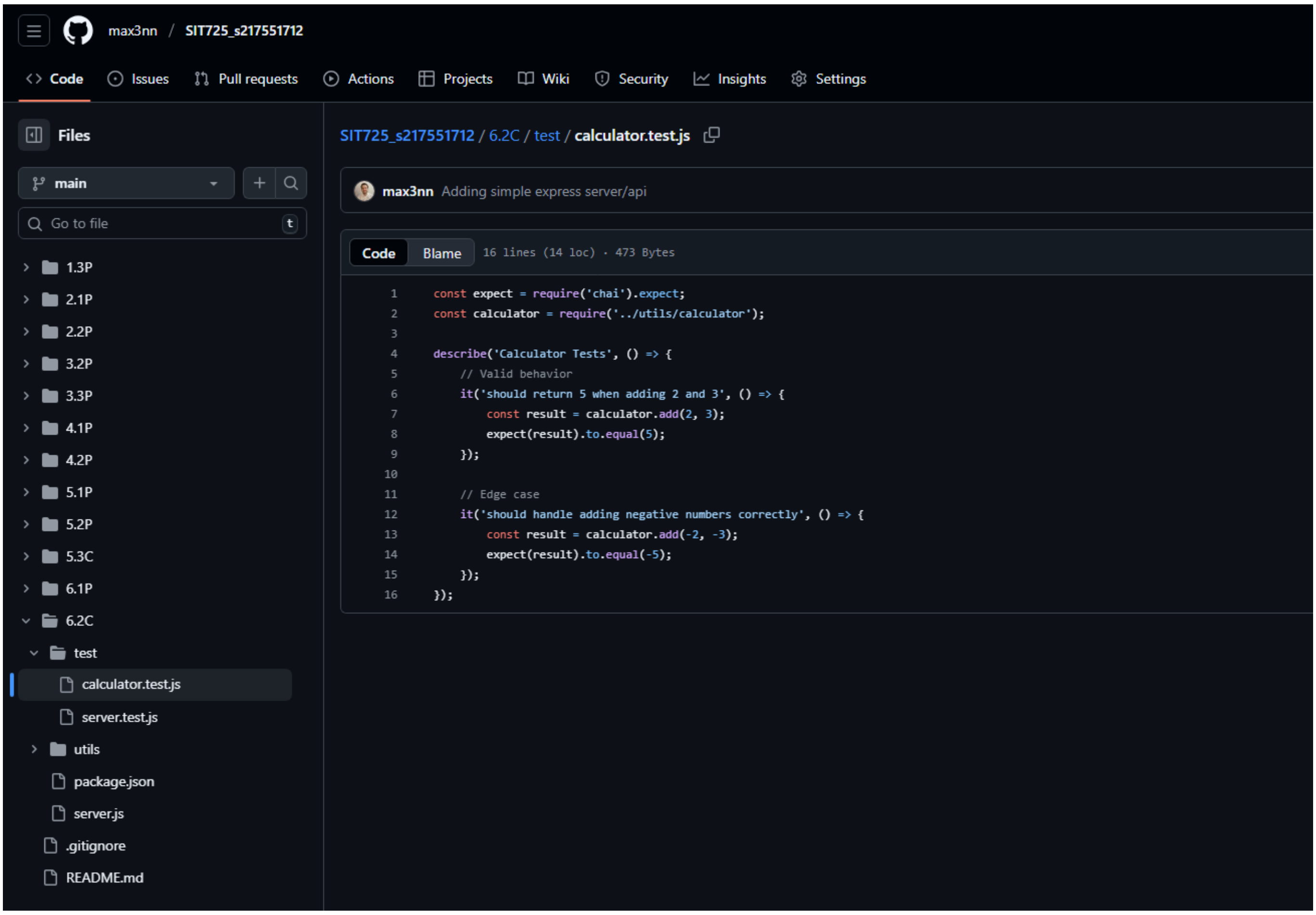
Evidence - Task 6.2C

GitHub Repository

https://github.com/max3nn/SIT725_s217551712/blob/main/6.2C/test/calculator.test.js

Screenshots

Repository

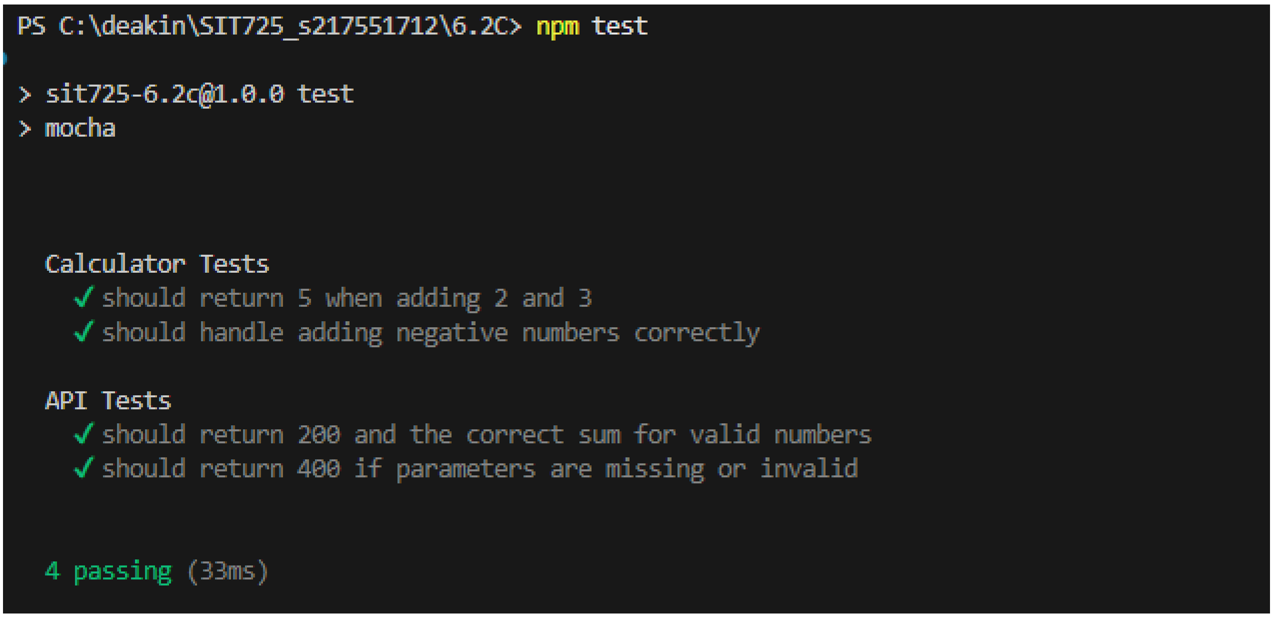


max3nn Adding simple express server/api

16 lines (14 loc) · 473 Bytes

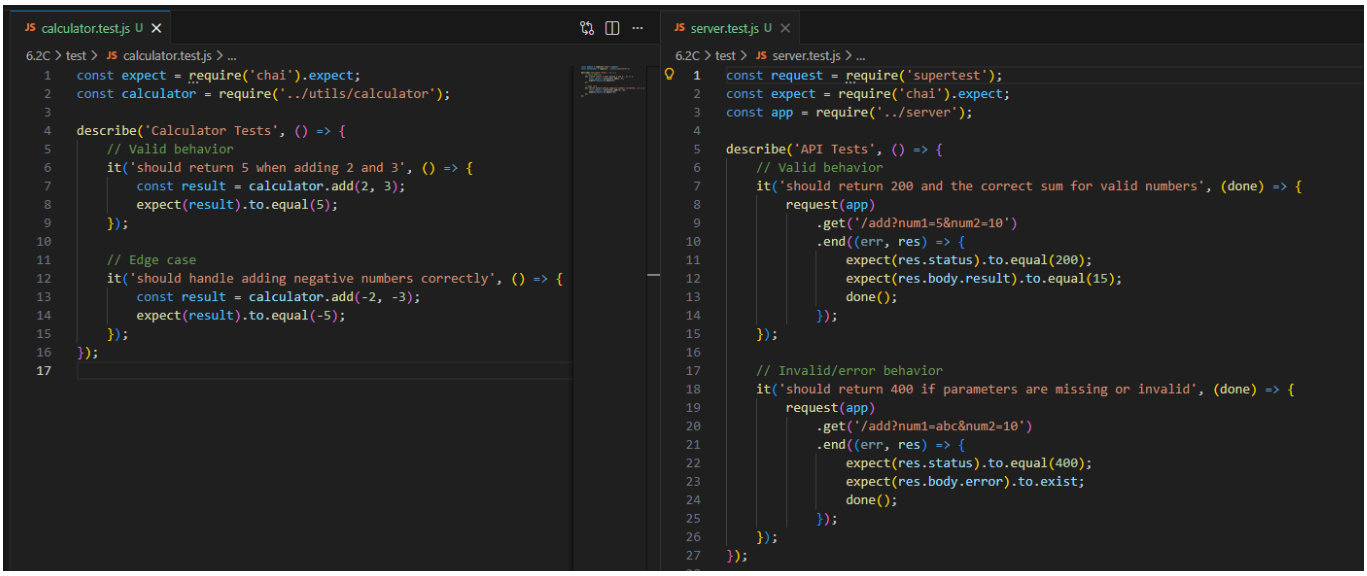
```
const expect = require('chai').expect;
const calculator = require('../utils/calculator');

describe('Calculator Tests', () => {
    // Valid behavior
    it('should return 5 when adding 2 and 3', () => {
        const result = calculator.add(2, 3);
        expect(result).to.equal(5);
    });

    // Edge case
    it('should handle adding negative numbers correctly', () => {
        const result = calculator.add(-2, -3);
        expect(result).to.equal(-5);
    });
});
```

Tests Running



```
PS C:\deakin\SIT725_s217551712\6.2C> npm test

> sit725-6.2c@1.0.0 test
> mocha



  Calculator Tests
    ✓ should return 5 when adding 2 and 3
    ✓ should handle adding negative numbers correctly

  API Tests
    ✓ should return 200 and the correct sum for valid numbers
    ✓ should return 400 if parameters are missing or invalid


  4 passing (33ms)
```

Tests



calculator.test.js U
6.2C > test > calculator.test.js > ...

```javascript
const expect = require('chai').expect;
const calculator = require('../utils/calculator');

describe('Calculator Tests', () => {
    // Valid behavior
    it('should return 5 when adding 2 and 3', () => {
        const result = calculator.add(2, 3);
        expect(result).to.equal(5);
    });

    // Edge case
    it('should handle adding negative numbers correctly', () => {
        const result = calculator.add(-2, -3);
        expect(result).to.equal(-5);
    });
});
```

server.test.js U
6.2C > test > server.test.js > ...

```javascript
const request = require('supertest');
const expect = require('chai').expect;
const app = require('../server');

describe('API Tests', () => {
    // Valid behavior
    it('should return 200 and the correct sum for valid numbers', (done) => {
        request(app)
            .get('/add?num1=5&num2=10')
            .end((err, res) => {
                expect(res.status).to.equal(200);
                expect(res.body.result).to.equal(15);
                done();
            });
    });

    // Invalid/error behavior
    it('should return 400 if parameters are missing or invalid', (done) => {
        request(app)
            .get('/add?num1=abc&num2=10')
            .end((err, res) => {
                expect(res.status).to.equal(400);
                expect(res.body.error).to.exist;
                done();
            });
    });
});
```