

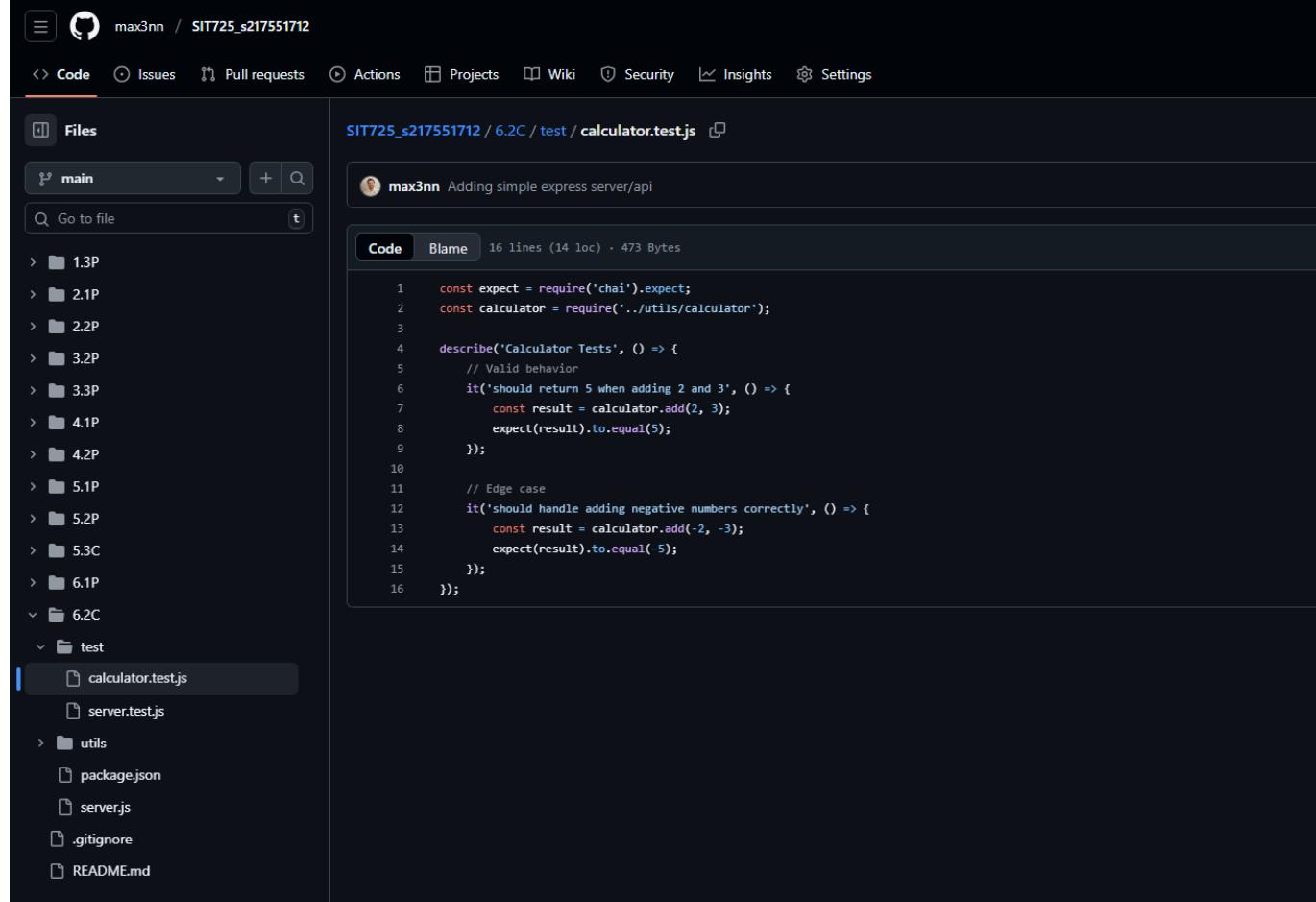
Evidence - Task 6.2C

GitHub Repository

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

Screenshots

Repository



The screenshot shows a GitHub repository page for 'SIT725_s217551712 / 6.2C / test / calculator.test.js'. The left sidebar shows a tree view of files, with 'calculator.test.js' selected. The main pane displays the code for 'calculator.test.js'.

```
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
6.2C > test > JS calculator.test.js > ...
1 const expect = require('chai').expect;
2 const calculator = require('../utils/calculator');
3
4 describe('Calculator Tests', () => {
5   // Valid behavior
6   it('should return 5 when adding 2 and 3', () => {
7     const result = calculator.add(2, 3);
8     expect(result).to.equal(5);
9   });
10
11   // Edge case
12   it('should handle adding negative numbers correctly', () => {
13     const result = calculator.add(-2, -3);
14     expect(result).to.equal(-5);
15   });
16 });
17

server.test.js
6.2C > test > JS server.test.js > ...
1 const request = require('supertest');
2 const expect = require('chai').expect;
3 const app = require('../server');
4
5 describe('API Tests', () => {
6   // Valid behavior
7   it('should return 200 and the correct sum for valid numbers', (done) => {
8     request(app)
9       .get('/add?num1=5&num2=10')
10      .end((err, res) => {
11        expect(res.status).to.equal(200);
12        expect(res.body.result).to.equal(15);
13        done();
14      });
15    });
16
17   // Invalid/error behavior
18   it('should return 400 if parameters are missing or invalid', (done) => {
19     request(app)
20       .get('/add?num1=abc&num2=10')
21      .end((err, res) => {
22        expect(res.status).to.equal(400);
23        expect(res.body.error).to.exist;
24        done();
25      });
26    });
27 });


```