IT314 Software Engineering
Lab 8
Project Name: **AthleteTalk**
Group 18

Project members :

Name	ID
PATEL KHUSHKUMAR GIRISHBHAI	202001127
ACHARYA HARSH MAHAVIRBHAI	202001131
UPADHYAY DHVANI VIPULKUMAR	202001136
PORIYA NEEL DHARMESHBHAI	202001139
PATEL PARTHKUMAR KANTIBHAI	202001147
VEGAD RAJ MAHESHBHAI	202001160
CHIRAG CHAVDA	202001164
SUVAGIYA ABHAY JAGDISHBHAI	202001171
VIRADIYA ABHAY	202001174

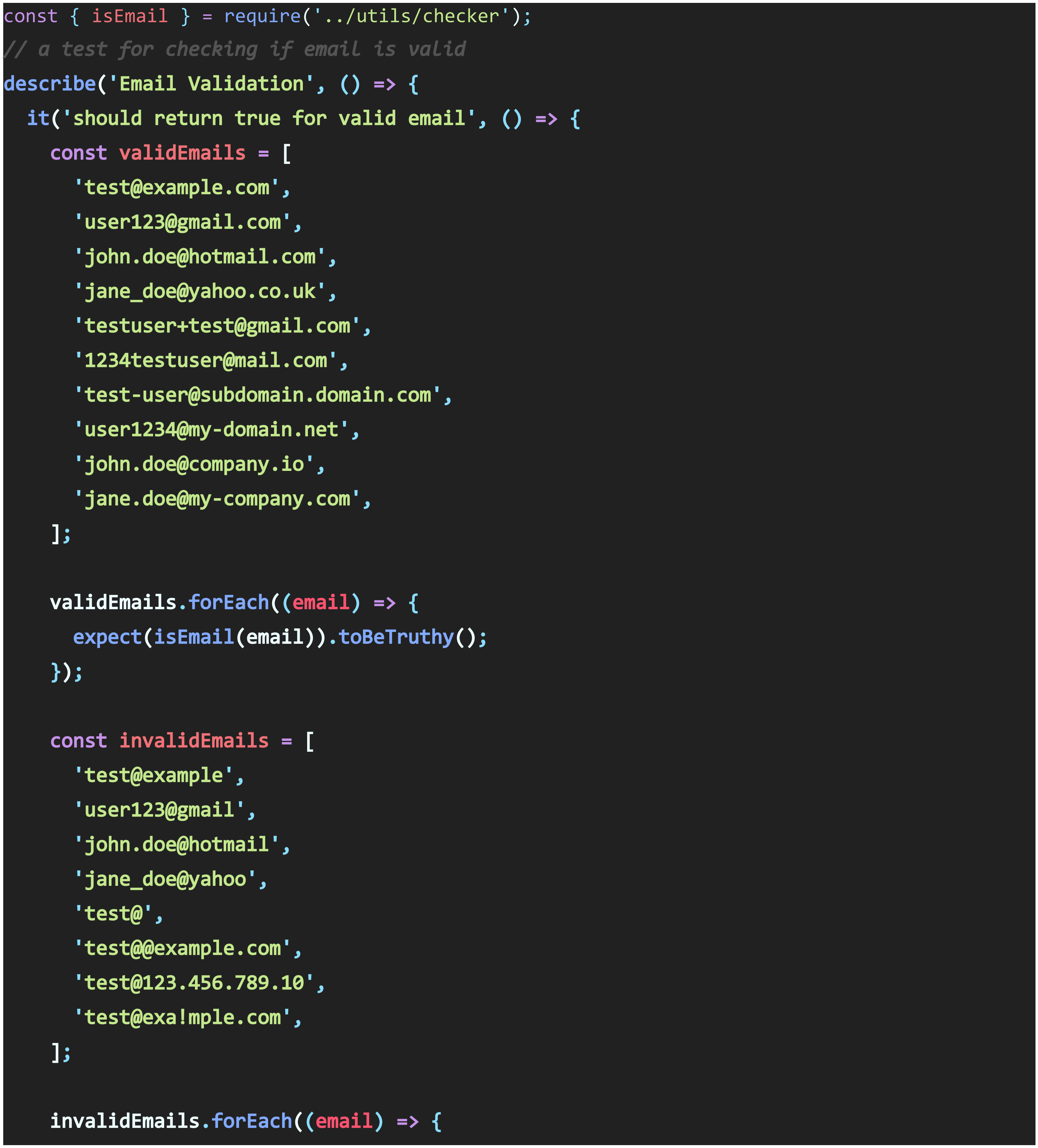
Testing Front-end code:

For testing front-end code, we have used Jest. Jest is a popular JavaScript testing framework used for unit testing and integration testing of JavaScript applications, including React, Node.js, and other JavaScript libraries.

Following are the test cases generated for our application:

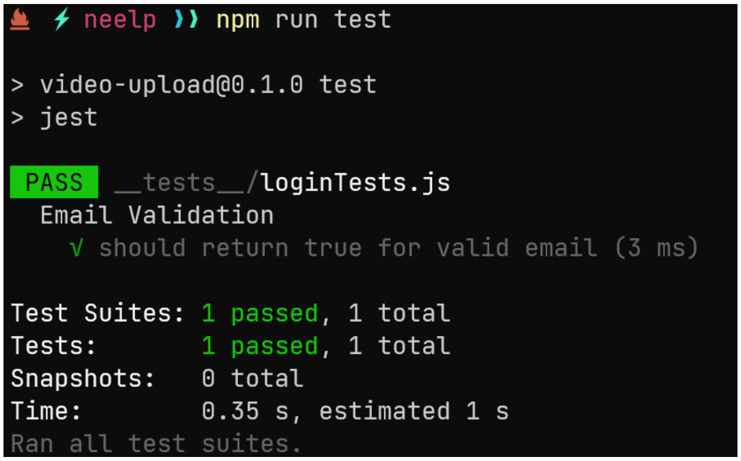
1. Checking for valid email addresses in the signup form.

Jest Code:



```
const { isEmail } = require('../utils/checker');
// a test for checking if email is valid
describe('Email Validation', () => {
  it('should return true for valid email', () => {
    const validEmails = [
      'test@example.com',
      'user123@gmail.com',
      'john.doe@hotmail.com',
      'jane_doe@yahoo.co.uk',
      'testuser+test@gmail.com',
      '1234testuser@mail.com',
      'test-user@subdomain.domain.com',
      'user1234@my-domain.net',
      'john.doe@company.io',
      'jane.doe@my-company.com',
    ];

    validEmails.forEach((email) => {
      expect(isEmail(email)).toBeTruthy();
    });

    const invalidEmails = [
      'test@example',
      'user123@gmail',
      'john.doe@hotmail',
      'jane_doe@yahoo',
      'test@',
      'test@@example.com',
      'test@123.456.789.10',
      'test@exa!mple.com',
    ];

    invalidEmails.forEach((email) => {
      expect(isEmail(email)).toBeFalsy();
    });
  });
});
```

Running the test in terminal,

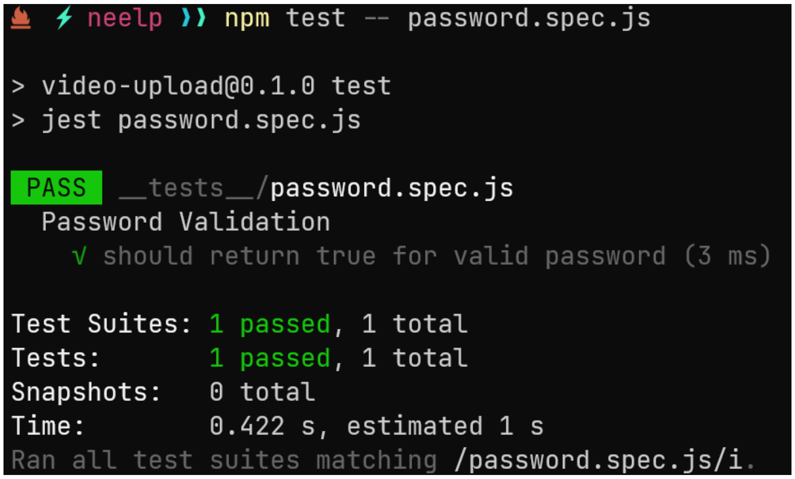


```
⚡ neelp ❯❯ npm run test

> video-upload@0.1.0 test
> jest

 PASS  __tests__/loginTests.js
  Email Validation
    √ should return true for valid email (3 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.35 s, estimated 1 s
Ran all test suites.
```

2. Checking for valid password in specific format.

```
const { isPassword } = require('../utils/checker');

// a test for checking if password is in proper format
describe('Password Validation', () => {
  it('should return true for valid password', () => {
    const validPasswords = [
      'Abcdefg1!',
      'HjkloP5@',
      'Qrstuv7#',
      'WxyZ123$',
      'Abcdefg1!',
      'MnbvCxz3$',
      'PoiuytR2!',
    ];

    validPasswords.forEach((password) => {
      expect(isPassword(password)).toBeTruthy();
    });

    const invalidPasswords = [
      'abcdefg1',
      'ABCDEFG1',
      'abcdefg',
      'ABCDEFGH',
      '12345678',
      '123456789',
    ];

    invalidPasswords.forEach((password) => {
      expect(isPassword(password)).toBeFalsy();
    });
  });
});
```

Running the test for password,



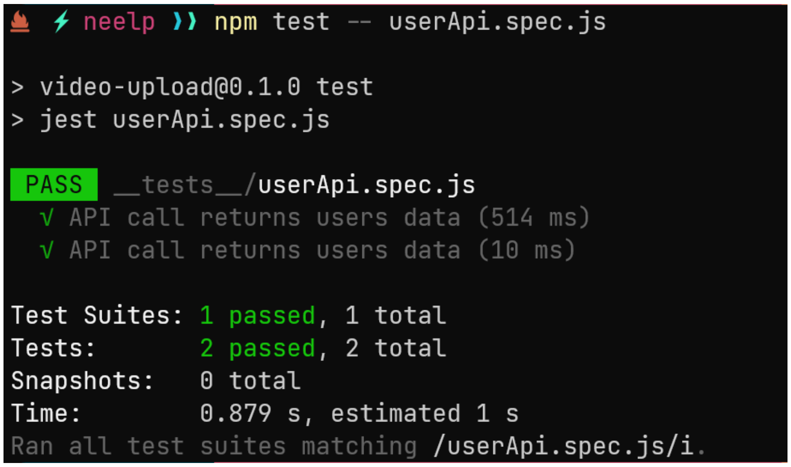
```
⚡ neelp ❯❯ npm test -- password.spec.js

> video-upload@0.1.0 test
> jest password.spec.js

 PASS  __tests__/password.spec.js
  Password Validation
    √ should return true for valid password (3 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.422 s, estimated 1 s
Ran all test suites matching /password.spec.js/i.
```

Tests for Backend API:

1. Tests for getting all the users using JWT token,

```
// Test for checking if an API call returns data where token is valid

test('API call returns users data', async () => {
  const res = await fetch(`http://localhost:3000/api/users`, {
    method: 'GET',
    headers: {
      authorization: `Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYzZmY3MjkwY2MzZmYwZTM0ZTgyYWQ5YyIsImlhdCI6MTY4MDcxM
DE4NCwiZXhwIjoxNjg4NDg2MTg0fQ.Svs6jJyMfgoRqLSwIHDJzOiqLKodfwJGc5xJDK1R_vE`,
    },
  });
  const responseData = await res.json();
  expect(responseData).toBeDefined();
  expect(responseData.users.length).toBeGreaterThan(0);
});

// Test for checking if an API call returns data with invalid token

test('API call returns users data', async () => {
  const res = await fetch(`http://localhost:3000/api/users`, {
    method: 'GET',
    headers: {
      authorization: `Bearer bogusToken`,
    },
  });
  const responseData = await res.json();
  expect(responseData.status).toBe('error');
});
```

Running the test,

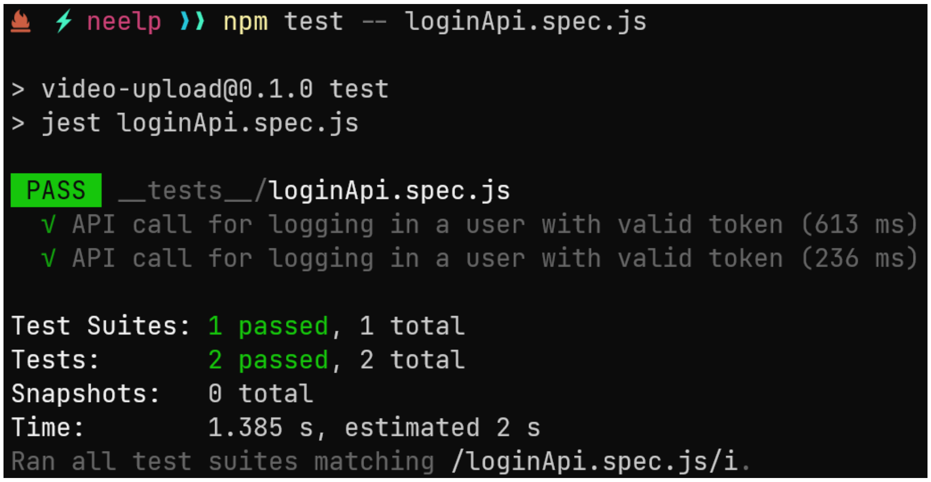


```
 ⚡ neelp ❱❱ npm test -- userApi.spec.js

> video-upload@0.1.0 test
> jest userApi.spec.js

 PASS  __tests__/userApi.spec.js
  √ API call returns users data (514 ms)
  √ API call returns users data (10 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.879 s, estimated 1 s
Ran all test suites matching /userApi.spec.js/i.
```

2. Test for api used for logging the user in,

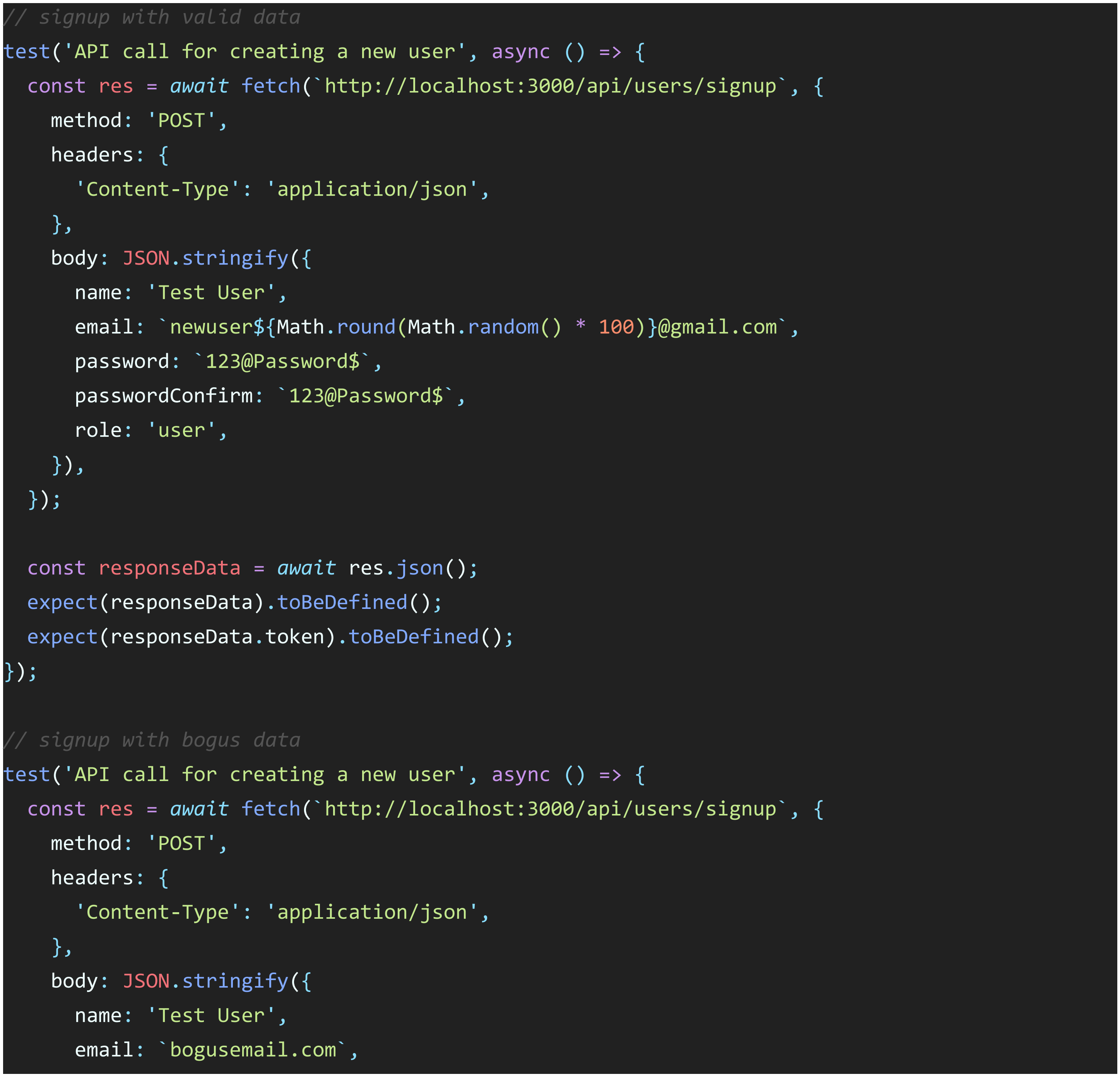
```
// logging in with valid credentials
test('API call for logging in a user with valid token', async () => {
  const res = await fetch(`http://localhost:3000/api/users/login`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      email: 'newuser27@gmail.com',
      password: '123@Password$',
    }),
  });

  const responseData = await res.json();
  expect(responseData).toBeDefined();
  expect(responseData.token).toBeDefined();
});

// logging in with bogus credentials
test('API call for logging in a user with valid token', async () => {
  const res = await fetch(`http://localhost:3000/api/users/login`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      email: 'randomguy@nodomain.com',
      password: '12345688',
    }),
  });

  const responseData = await res.json();
  expect(responseData).toBeDefined();
  expect(responseData.status).toBe('fail');
});
```

Running the test,



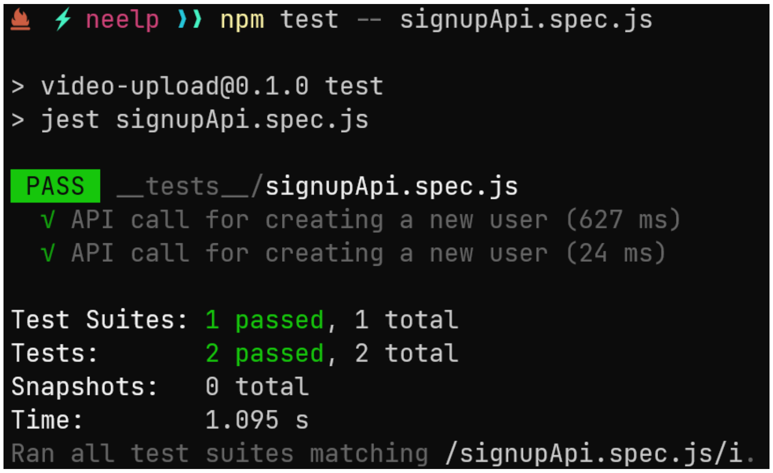
```
⚡ neelp ❯❯ npm test -- loginApi.spec.js

> video-upload@0.1.0 test
> jest loginApi.spec.js

 PASS  __tests__/loginApi.spec.js
  √ API call for logging in a user with valid token (613 ms)
  √ API call for logging in a user with valid token (236 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.385 s, estimated 2 s
Ran all test suites matching /loginApi.spec.js/i.
```

3. Tests for signing up a new user,



```
// signup with valid data
test('API call for creating a new user', async () => {
  const res = await fetch(`http://localhost:3000/api/users/signup`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      name: 'Test User',
      email: `newuser${Math.round(Math.random() * 100)}@gmail.com`,
      password: `123@Password$`,
      passwordConfirm: `123@Password$`,
      role: 'user',
    }),
  });

  const responseData = await res.json();
  expect(responseData).toBeDefined();
  expect(responseData.token).toBeDefined();
});

// signup with bogus data
test('API call for creating a new user', async () => {
  const res = await fetch(`http://localhost:3000/api/users/signup`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      name: 'Test User',
      email: `bogusemail.com`,
      password: `123`,
      passwordConfirm: `123`,
      role: 'not a role',
    }),
  });

  const responseData = await res.json();
  expect(responseData).toBeDefined();
  expect(responseData.status).toBe('error');
});
```

Running the tests,

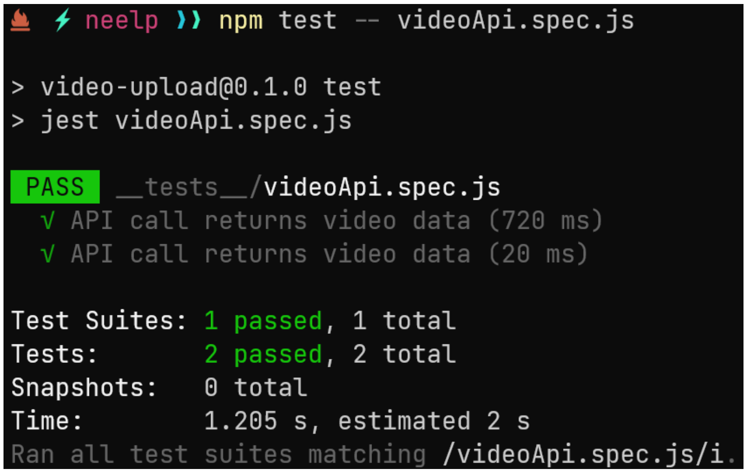


```
⚡ neelp ❯❯ npm test -- signupApi.spec.js

> video-upload@0.1.0 test
> jest signupApi.spec.js

 PASS  __tests__/signupApi.spec.js
  √ API call for creating a new user (627 ms)
  √ API call for creating a new user (24 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.095 s
Ran all test suites matching /signupApi.spec.js/i.
```

4. Test for fetching videos from backend,

```
// calling the right api endpoint
test('API call returns video data', async () => {
  const res = await fetch(`http://localhost:3000/api/videos`, {
    method: 'GET',
    headers: {
      authorization: `Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYzZmY3MjkwY2MzZmYwZTM0ZTgyYWQ5YyIsImlhdCI6
MTY4MDcxMDE4NCwiZXhwIjoxNjg4NDg2MTg0fQ.Svs6jJyMfgoRqLSwIHDJzOiqLKodfwJGc5xJDK1R_vE`,
    },
  });
  const responseData = await res.json();
  expect(responseData).toBeDefined();
  expect(responseData.results).toBeGreaterThan(0);
});

// calling the some other api endpoint
test('API call returns video data', async () => {
  const res = await fetch(`http://localhost:3000/api/myvideos/videos`, {
    method: 'GET',
    headers: {
      authorization: `Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYzZmY3MjkwY2MzZmYwZTM0ZTgyYWQ5YyIsImlhdCI6
MTY4MDcxMDE4NCwiZXhwIjoxNjg4NDg2MTg0fQ.Svs6jJyMfgoRqLSwIHDJzOiqLKodfwJGc5xJDK1R_vE`,
    },
  });
  expect(res.ok).toBeFalsy();
});
```

Running the tests,

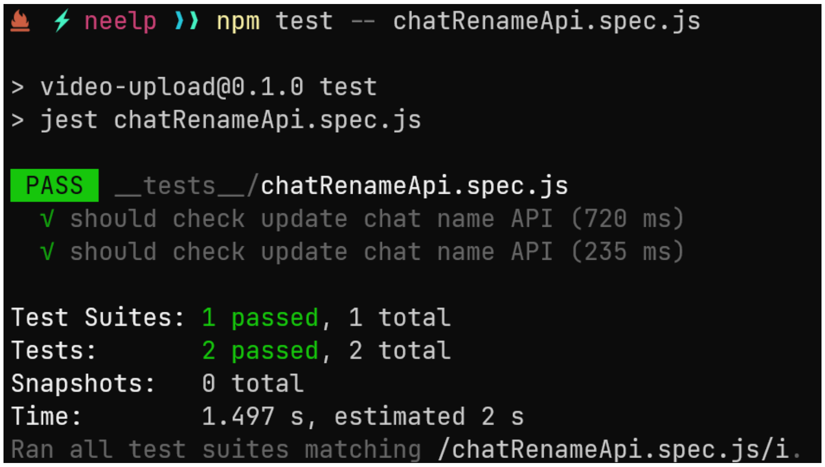


```
⚡ neelp ❯❯ npm test -- videoApi.spec.js

> video-upload@0.1.0 test
> jest videoApi.spec.js

 PASS  __tests__/videoApi.spec.js
  √ API call returns video data (720 ms)
  √ API call returns video data (20 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.205 s, estimated 2 s
Ran all test suites matching /videoApi.spec.js/i.
```

5. Test for API for renaming the group chat name,

```
// updating chatName with valid chatId and valid token
test('should check update chat name API', async () => {
  const res = await fetch(`http://localhost:3000/api/chat/rename`, {
    method: 'PUT',
    headers: {
      'Content-Type': 'application/json',
      authorization: `Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYzZmY3MjkwY2MzZmYwZTM0ZTgyYWQ5YyIsImlhdCI6
MTY4MDcxMDE4NCwiZXhwIjoxNjg4NDg2MTg0fQ.Svs6jJyMfgoRqLSwIHDJzOiqLKodfwJGc5xJDK1R_vE`,
    },
    body: JSON.stringify({
      chatId: '64393739d88cc41d92b96b56',
      chatName: 'myNewChatName',
    }),
  });
  const responseData = await res.json();
  expect(responseData).toBeDefined();
  expect(responseData.status).toBe('success');
  expect(responseData.data.chat.chatName).toBe('myNewChatName');
});

// updating chatName with bogus chatId
test('should check update chat name API', async () => {
  const res = await fetch(`http://localhost:3000/api/chat/rename`, {
    method: 'PUT',
    headers: {
      'Content-Type': 'application/json',
      authorization: `Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYzZmY3MjkwY2MzZmYwZTM0ZTgyYWQ5YyIsImlhdCI6
MTY4MDcxMDE4NCwiZXhwIjoxNjg4NDg2MTg0fQ.Svs6jJyMfgoRqLSwIHDJzOiqLKodfwJGc5xJDK1R_vE`,
    },
    body: JSON.stringify({
      chatId: 'noChatWithThisId',
      chatName: 'myNewChatName',
    }),
  });
  const responseData = await res.json();
  expect(responseData).toBeDefined();
  expect(responseData.status).toBe('error');
});
```

Running the tests,



```
neelp ⟫ npm test -- chatRenameApi.spec.js

> video-upload@0.1.0 test
> jest chatRenameApi.spec.js

 PASS  __tests__/chatRenameApi.spec.js
  √ should check update chat name API (720 ms)
  √ should check update chat name API (235 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.497 s, estimated 2 s
Ran all test suites matching /chatRenameApi.spec.js/i.
```

So these are some of the tests we created for our application using Jest library.

END OF ASSIGNMENT