const request = require('supertest');

const app = require('./server');

describe('Auth Routes', () => {

it('should signup a user successfully', async () => {

const response = await request(app).post('/signup').send({

name: 'John Doe',

email: 'john@example.com',

password: '123456',

});

expect(response.statusCode).toBe(201);

expect(response.body.message).toBe('User signed up successfully!');

});

it('should not signup a user with an existing email', async () => {

await request(app).post('/signup').send({

name: 'John Doe',

email: 'john@example.com',

password: '123456',

});

const response = await request(app).post('/signup').send({

name: 'Jane Doe',

email: 'john@example.com',

password: '654321',

});

expect(response.statusCode).toBe(400);

expect(response.body.message).toBe('User already exists!');

});

it('should login a user successfully', async () => {

await request(app).post('/signup').send({

name: 'John Doe',

email: 'john@example.com',

password: '123456',

});

const response = await request(app).post('/login').send({

email: 'john@example.com',

password: '123456',

});

expect(response.statusCode).toBe(200);

expect(response.body.message).toBe('Login successful!');

});

it('should not login with invalid credentials', async () => {

const response = await request(app).post('/login').send({

email: 'john@example.com',

password: 'wrongpassword',

});

expect(response.statusCode).toBe(401);

expect(response.body.message).toBe('Invalid email or password!');

  });

});