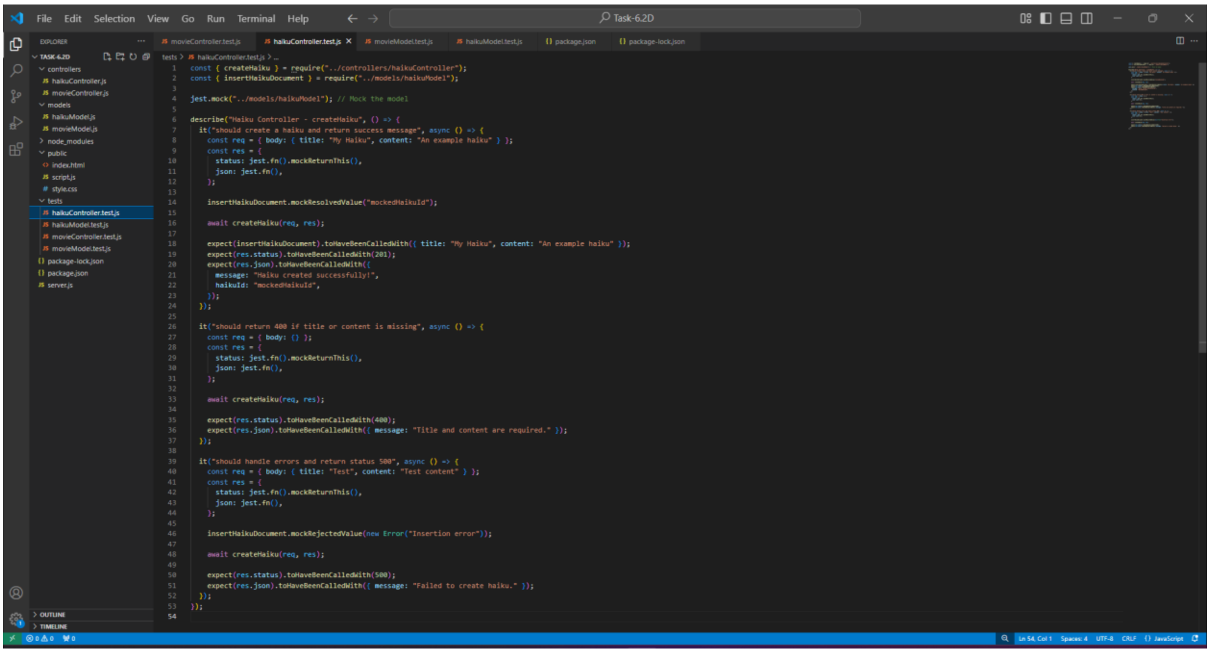
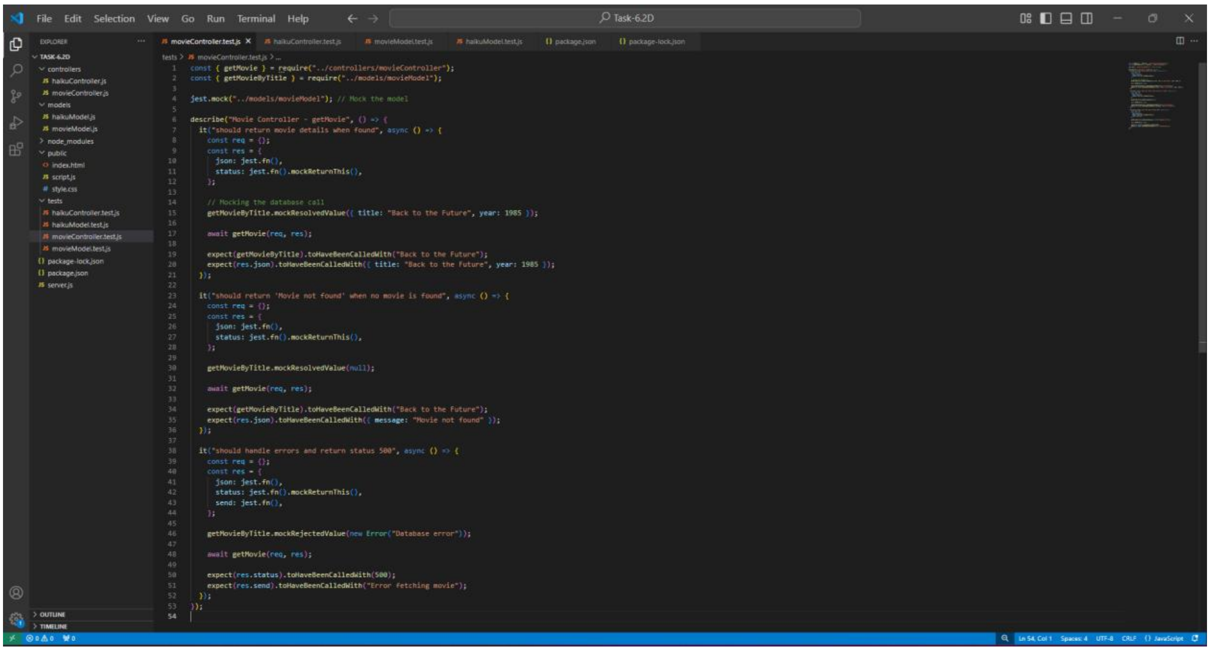
Task 6.2D

I created a *tests* folder and tested code in the *models* and *controllers* folder there.

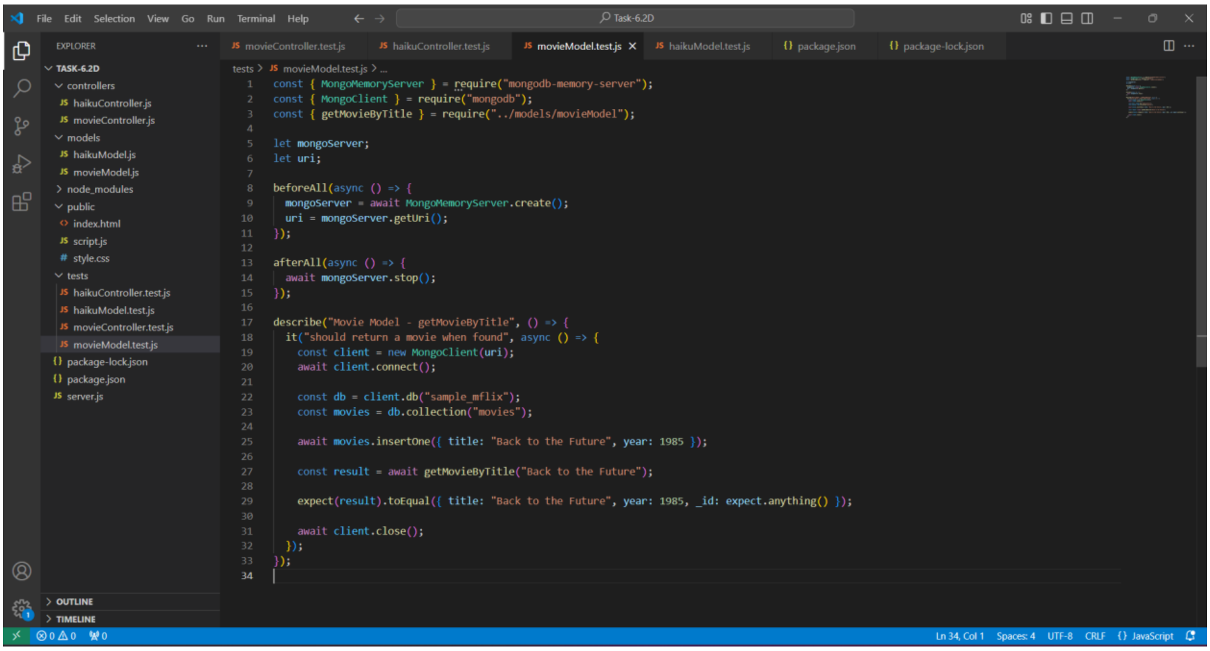
Below given are the screenshots:



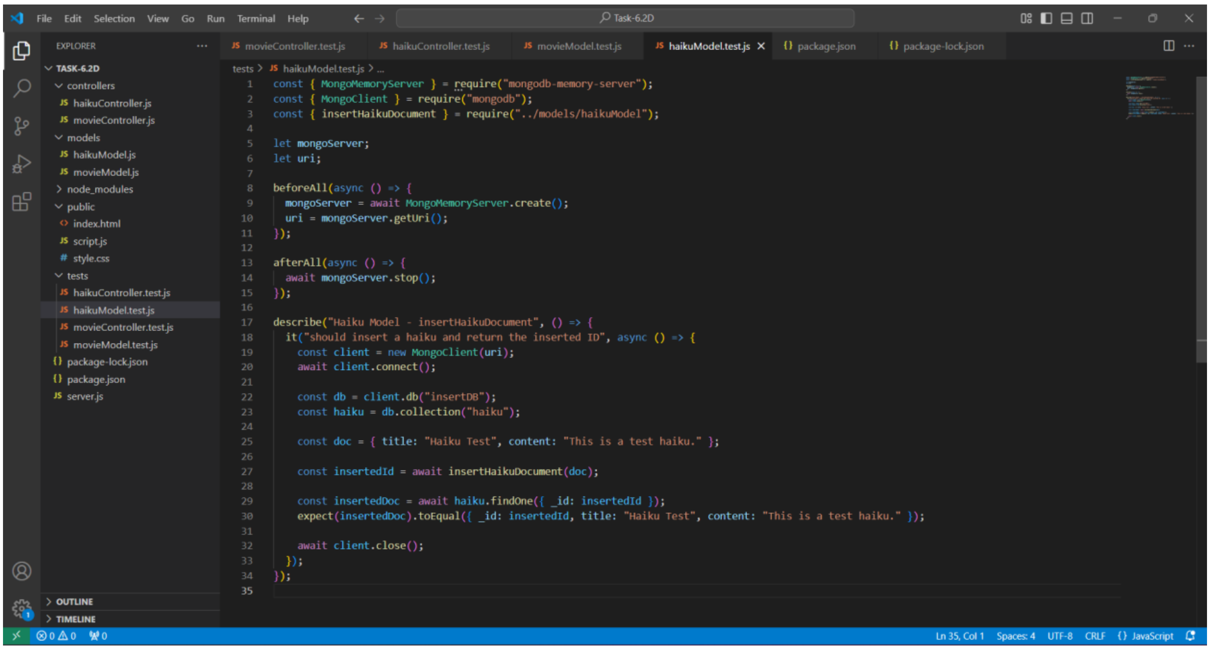
```javascript
const { createHaiku } = require("../controllers/haikuController");
const { insertHaikuDocument } = require("../models/haikuModel");

jest.mock("../models/haikuModel"); // Mock the model

describe("Haiku Controller - createHaiku", () => {
  it("should create a haiku and return success message", async () => {
    const req = { body: { title: "My Haiku", content: "An example haiku" } };
    const res = {
      status: jest.fn().mockReturnThis(),
      json: jest.fn(),
    };

    insertHaikuDocument.mockResolvedValue("mockedHaikuId");

    await createHaiku(req, res);

    expect(insertHaikuDocument).toHaveBeenCalledWith({ title: "My Haiku", content: "An example haiku" });
    expect(res.status).toHaveBeenCalledWith(201);
    expect(res.json).toHaveBeenCalledWith({
      message: "Haiku created successfully!",
      haikuId: "mockedHaikuId",
    });
  });

  it("should return 400 if title or content is missing", async () => {
    const req = { body: {} };
    const res = {
      status: jest.fn().mockReturnThis(),
      json: jest.fn(),
    };

    await createHaiku(req, res);

    expect(res.status).toHaveBeenCalledWith(400);
    expect(res.json).toHaveBeenCalledWith({ message: "Title and content are required." });
  });

  it("should handle errors and return status 500", async () => {
    const req = { body: { title: "Test", content: "Test content" } };
    const res = {
      status: jest.fn().mockReturnThis(),
      json: jest.fn(),
    };

    insertHaikuDocument.mockRejectedValue(new Error("Insertion error"));

    await createHaiku(req, res);

    expect(res.status).toHaveBeenCalledWith(500);
    expect(res.json).toHaveBeenCalledWith({ message: "Failed to create haiku." });
  });
});
```



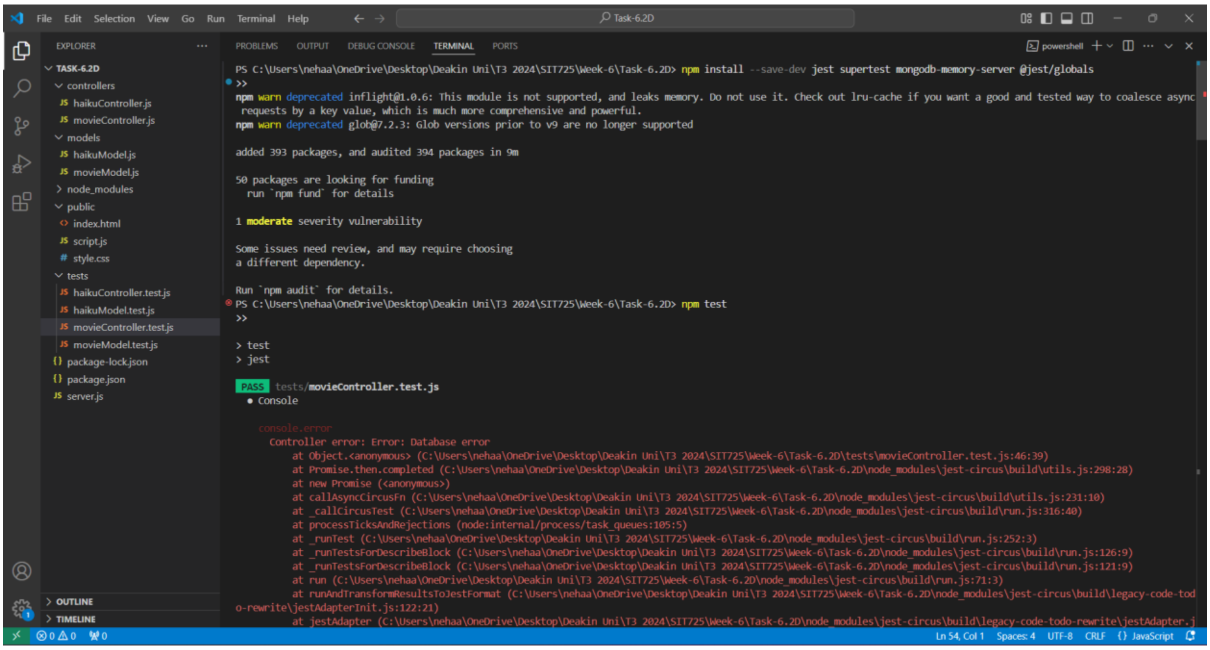
```javascript
const { getMovie } = require("../controllers/movieController");
const { getMovieByTitle } = require("../models/movieModel");

jest.mock("../models/movieModel"); // Mock the model

describe("Movie Controller - getMovie", () => {
  it("should return movie details when found", async () => {
    const req = {};
    const res = {
      json: jest.fn(),
      status: jest.fn().mockReturnThis(),
    };

    // Mocking the database call
    getMovieByTitle.mockResolvedValue({ title: "Back to the Future", year: 1985 });

    await getMovie(req, res);

    expect(getMovieByTitle).toHaveBeenCalledWith("Back to the Future");
    expect(res.json).toHaveBeenCalledWith({ title: "Back to the Future", year: 1985 });
  });

  it("should return 'Movie not found' when no movie is found", async () => {
    const req = {};
    const res = {
      json: jest.fn(),
      status: jest.fn().mockReturnThis(),
    };

    getMovieByTitle.mockResolvedValue(null);

    await getMovie(req, res);

    expect(getMovieByTitle).toHaveBeenCalledWith("Back to the Future");
    expect(res.json).toHaveBeenCalledWith({ message: "Movie not found" });
  });

  it("should handle errors and return status 500", async () => {
    const req = {};
    const res = {
      json: jest.fn(),
      status: jest.fn().mockReturnThis(),
      send: jest.fn(),
    };

    getMovieByTitle.mockRejectedValue(new Error("Database error"));

    await getMovie(req, res);

    expect(res.status).toHaveBeenCalledWith(500);
    expect(res.send).toHaveBeenCalledWith("Error fetching movie");
  });
});
```



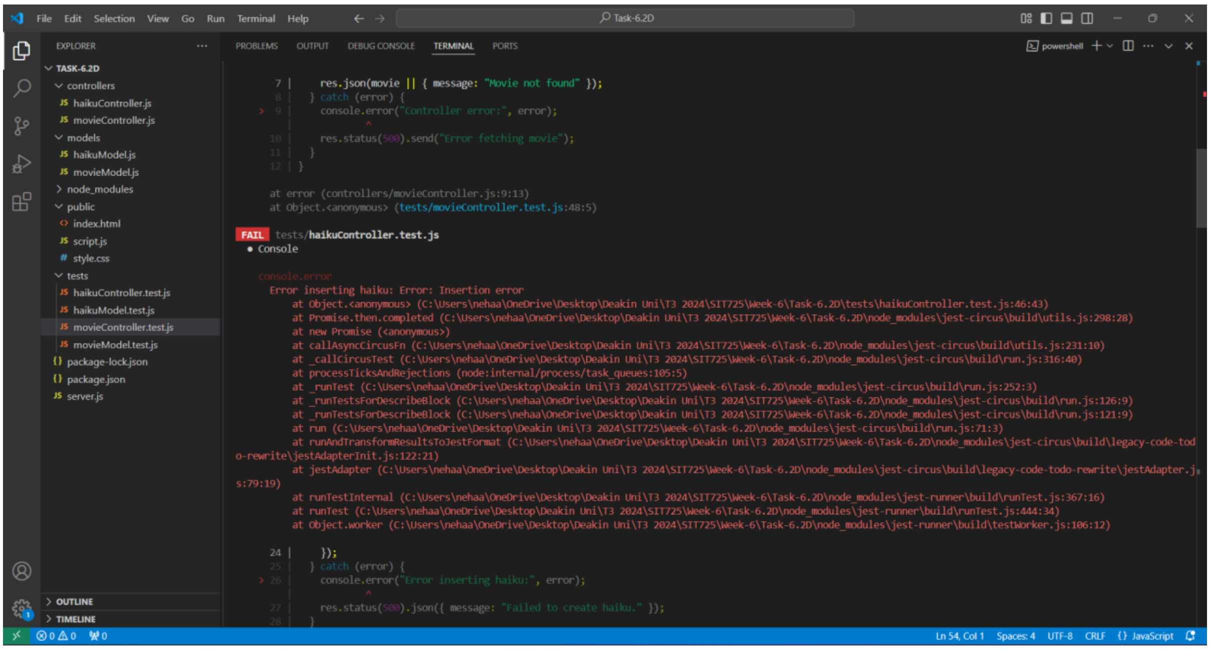
```javascript
const { MongoMemoryServer } = require("mongodb-memory-server");
const { MongoClient } = require("mongodb");
const { getMovieByTitle } = require("../models/movieModel");

let mongoServer;
let uri;

beforeAll(async () => {
  mongoServer = await MongoMemoryServer.create();
  uri = mongoServer.getUri();
});

afterAll(async () => {
  await mongoServer.stop();
});

describe("Movie Model - getMovieByTitle", () => {
  it("should return a movie when found", async () => {
    const client = new MongoClient(uri);
    await client.connect();

    const db = client.db("sample_mflix");
    const movies = db.collection("movies");

    await movies.insertOne({ title: "Back to the Future", year: 1985 });

    const result = await getMovieByTitle("Back to the Future");

    expect(result).toEqual({ title: "Back to the Future", year: 1985, _id: expect.anything() });

    await client.close();
  });
});
```



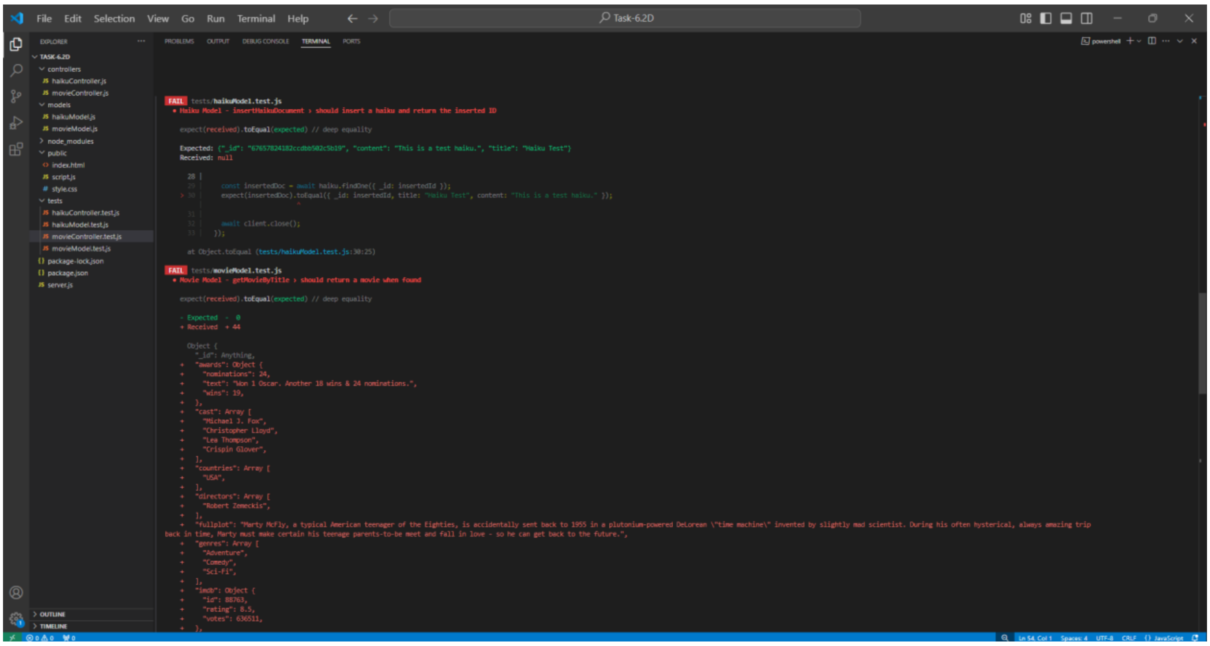
```javascript
const { MongoMemoryServer } = require("mongodb-memory-server");
const { MongoClient } = require("mongodb");
const { insertHaikuDocument } = require("../models/haikuModel");

let mongoServer;
let uri;

beforeAll(async () => {
  mongoServer = await MongoMemoryServer.create();
  uri = mongoServer.getUri();
});

afterAll(async () => {
  await mongoServer.stop();
});

describe("Haiku Model - insertHaikuDocument", () => {
  it("should insert a haiku and return the inserted ID", async () => {
    const client = new MongoClient(uri);
    await client.connect();

    const db = client.db("insertDB");
    const haiku = db.collection("haiku");

    const doc = { title: "Haiku Test", content: "This is a test haiku." };

    const insertedId = await insertHaikuDocument(doc);

    const insertedDoc = await haiku.findOne({ _id: insertedId });
    expect(insertedDoc).toEqual({ _id: insertedId, title: "Haiku Test", content: "This is a test haiku." });

    await client.close();
  });
});
```



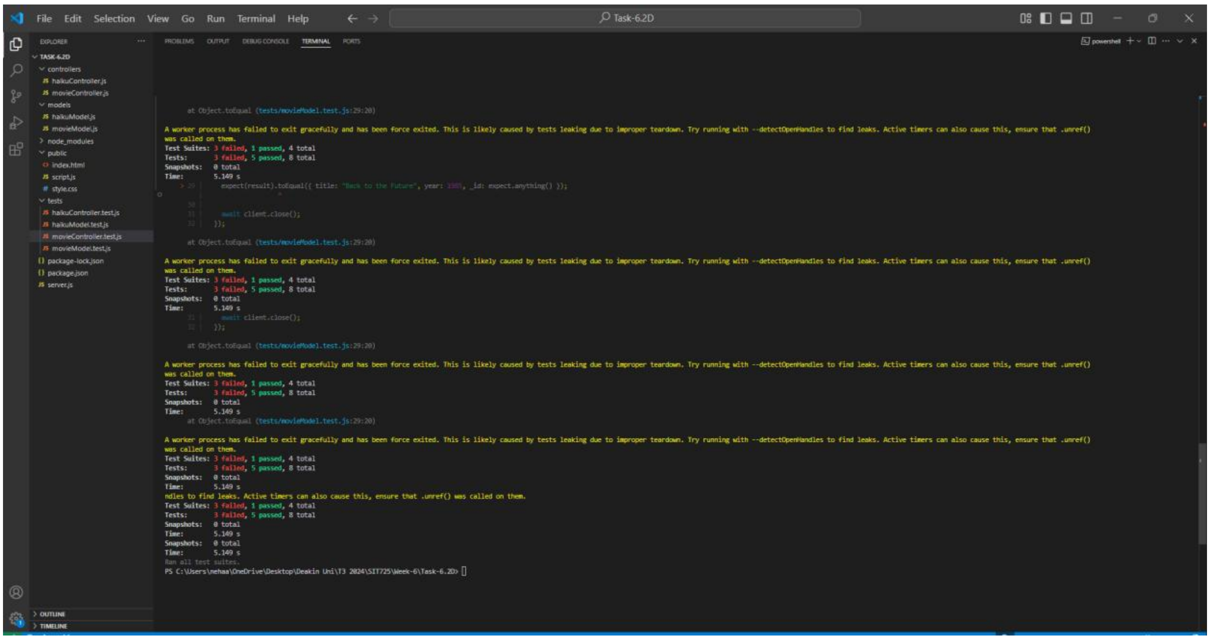
```
PS C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D> npm install --save-dev jest supertest mongodb-memory-server @jest/globals
>>
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async
 requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported

added 393 packages, and audited 394 packages in 9m

50 packages are looking for funding
  run `npm fund` for details

1 moderate severity vulnerability

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
PS C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D> npm test
>>

> test
> jest

 PASS  tests/movieController.test.js
  ● Console

    console.error
      Controller error: Error: Database error
          at Object.<anonymous> (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\tests\movieController.test.js:46:39)
          at Promise.then.completed (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\utils.js:298:28)
          at new Promise (<anonymous>)
          at callAsyncCircusFn (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\utils.js:231:10)
          at _callCircusTest (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:316:40)
          at processTicksAndRejections (node:internal/process/task_queues:105:5)
          at _runTest (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:252:3)
          at _runTestsForDescribeBlock (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:126:9)
          at _runTestsForDescribeBlock (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:121:9)
          at run (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:71:3)
          at runAndTransformResultsToJestFormat (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\legacy-code-tod
o-rewrite\jestAdapterInit.js:122:21)
          at jestAdapter (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\legacy-code-todo-rewrite\jestAdapter.j
```



```
       7 |       res.json(movie || { message: "Movie not found" });
       8 |     } catch (error) {
    >  9 |       console.error("Controller error:", error);
         |               ^
      10 |       res.status(500).send("Error fetching movie");
      11 |     }
      12 |   }

      at error (controllers/movieController.js:9:13)
      at Object.<anonymous> (tests/movieController.test.js:48:5)

 FAIL  tests/haikuController.test.js
  ● Console

    console.error
      Error inserting haiku: Error: Insertion error
          at Object.<anonymous> (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\tests\haikuController.test.js:46:43)
          at Promise.then.completed (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\utils.js:298:28)
          at new Promise (<anonymous>)
          at callAsyncCircusFn (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\utils.js:231:10)
          at _callCircusTest (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:316:40)
          at processTicksAndRejections (node:internal/process/task_queues:105:5)
          at _runTest (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:252:3)
          at _runTestsForDescribeBlock (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:126:9)
          at _runTestsForDescribeBlock (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:121:9)
          at run (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\run.js:71:3)
          at runAndTransformResultsToJestFormat (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\legacy-code-tod
o-rewrite\jestAdapterInit.js:122:21)
          at jestAdapter (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-circus\build\legacy-code-todo-rewrite\jestAdapter.j
s:79:19)
          at runTestInternal (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-runner\build\runTest.js:367:16)
          at runTest (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-runner\build\runTest.js:444:34)
          at Object.worker (C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D\node_modules\jest-runner\build\testWorker.js:106:12)

      24 |       });
      25 |     } catch (error) {
    > 26 |       console.error("Error inserting haiku:", error);
         |               ^
      27 |       res.status(500).json({ message: "Failed to create haiku." });
      28 |     }
```



```
FAIL tests/haikuModel.test.js
  ● Haiku Model - insertHaikuDocument › should insert a haiku and return the inserted ID

    expect(received).toEqual(expected) // deep equality

    Expected: {"_id": "67657824182ccdbb502c5b19", "content": "This is a test haiku.", "title": "Haiku Test"}
    Received: null

      28 |
      29 |         const insertedDoc = await haiku.findOne({ _id: insertedId });
    > 30 |         expect(insertedDoc).toEqual({ _id: insertedId, title: "Haiku Test", content: "This is a test haiku." });
         |                             ^
      31 |
      32 |         await client.close();
      33 |     });

      at Object.toEqual (tests/haikuModel.test.js:30:25)

FAIL tests/movieModel.test.js
  ● Movie Model - getMovieByTitle › should return a movie when found

    expect(received).toEqual(expected) // deep equality

    - Expected  -  0
    + Received  + 44

      Object {
        "_id": Anything,
    +   "awards": Object {
    +     "nominations": 24,
    +     "text": "Won 1 Oscar. Another 18 wins & 24 nominations.",
    +     "wins": 19,
    +   },
    +   "cast": Array [
    +     "Michael J. Fox",
    +     "Christopher Lloyd",
    +     "Lea Thompson",
    +     "Crispin Glover",
    +   ],
    +   "countries": Array [
    +     "USA",
    +   ],
    +   "directors": Array [
    +     "Robert Zemeckis",
    +   ],
    +   "fullplot": "Marty McFly, a typical American teenager of the Eighties, is accidentally sent back to 1955 in a plutonium-powered DeLorean \"time machine\" invented by slightly mad scientist. During his often hysterical, always amazing trip
back in time, Marty must make certain his teenage parents-to-be meet and fall in love - so he can get back to the future.",
    +   "genres": Array [
    +     "Adventure",
    +     "Comedy",
    +     "Sci-Fi",
    +   ],
    +   "imdb": Object {
    +     "id": 88763,
    +     "rating": 8.5,
    +     "votes": 636511,
    +   },
```



```
      at Object.toEqual (tests/movieModel.test.js:29:20)

A worker process has failed to exit gracefully and has been force exited. This is likely caused by tests leaking due to improper teardown. Try running with --detectOpenHandles to find leaks. Active timers can also cause this, ensure that .unref()
was called on them.
Test Suites: 3 failed, 1 passed, 4 total
Tests:       3 failed, 5 passed, 8 total
Snapshots:   0 total
Time:        5.149 s
    > 29 |         expect(result).toEqual({ title: "Back to the Future", year: 1985, _id: expect.anything() });
         |                        ^
      30 |
      31 |         await client.close();
      32 |     });

      at Object.toEqual (tests/movieModel.test.js:29:20)

A worker process has failed to exit gracefully and has been force exited. This is likely caused by tests leaking due to improper teardown. Try running with --detectOpenHandles to find leaks. Active timers can also cause this, ensure that .unref()
was called on them.
Test Suites: 3 failed, 1 passed, 4 total
Tests:       3 failed, 5 passed, 8 total
Snapshots:   0 total
Time:        5.149 s
      31 |         await client.close();
      32 |     });

      at Object.toEqual (tests/movieModel.test.js:29:20)

A worker process has failed to exit gracefully and has been force exited. This is likely caused by tests leaking due to improper teardown. Try running with --detectOpenHandles to find leaks. Active timers can also cause this, ensure that .unref()
was called on them.
Test Suites: 3 failed, 1 passed, 4 total
Tests:       3 failed, 5 passed, 8 total
Snapshots:   0 total
Time:        5.149 s
      at Object.toEqual (tests/movieModel.test.js:29:20)

A worker process has failed to exit gracefully and has been force exited. This is likely caused by tests leaking due to improper teardown. Try running with --detectOpenHandles to find leaks. Active timers can also cause this, ensure that .unref()
was called on them.
Test Suites: 3 failed, 1 passed, 4 total
Tests:       3 failed, 5 passed, 8 total
Snapshots:   0 total
Time:        5.149 s
ndles to find leaks. Active timers can also cause this, ensure that .unref() was called on them.
Test Suites: 3 failed, 1 passed, 4 total
Tests:       3 failed, 5 passed, 8 total
Snapshots:   0 total
Time:        5.149 s
Snapshots:   0 total
Time:        5.149 s
Ran all test suites.
PS C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-6\Task-6.2D> 
```