Table of Contents

[Overview 2](#_Toc197532684)

[UI Tests 2](#_Toc197532685)

[API Tests 2](#_Toc197532686)

[Test Data 3](#_Toc197532687)

[Files 3](#_Toc197532688)

[Random Variables 4](#_Toc197532689)

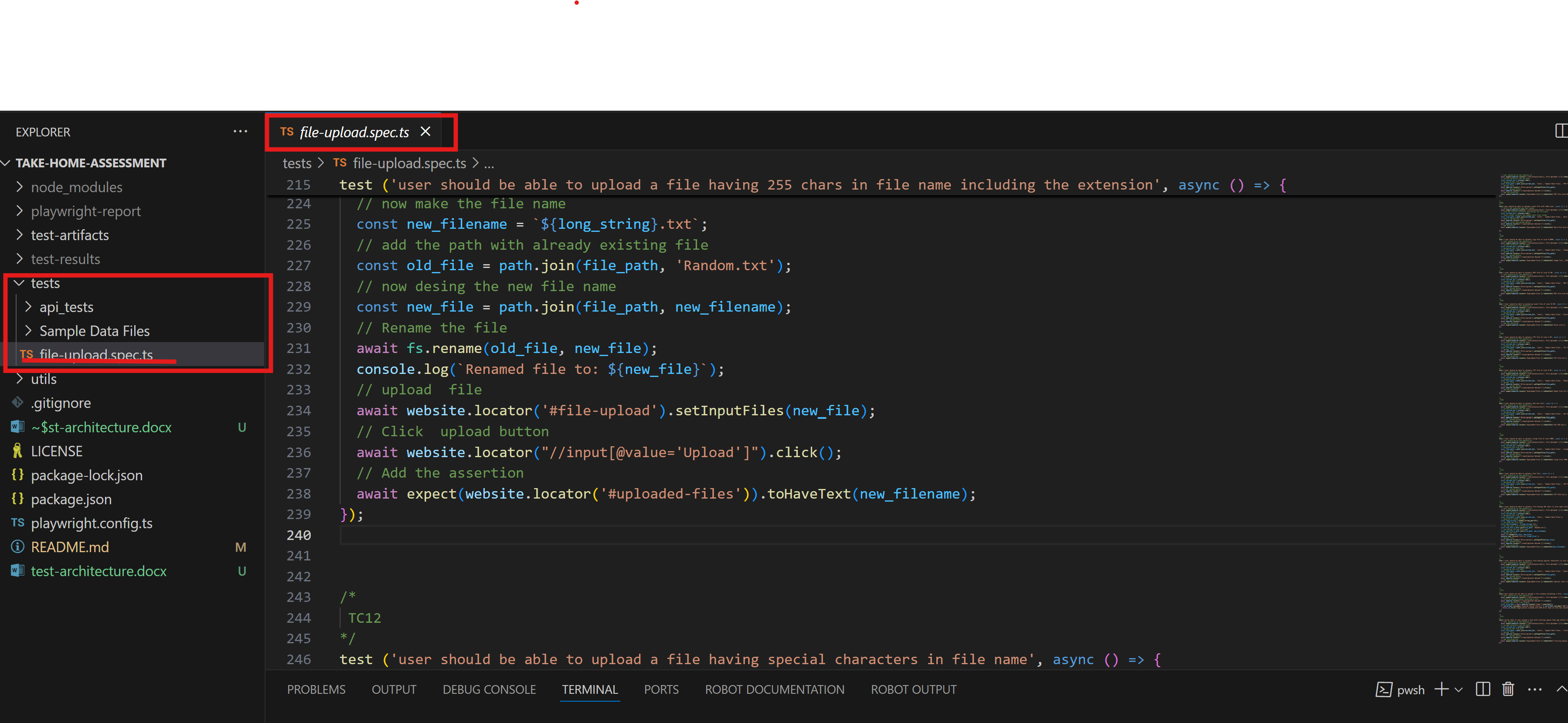
[Test Report 5](#_Toc197532690)

Overview:

The test architecture is simple and consists of two main components: one for API testing and another for UI testing. I have used the built-in reporting mechanism provided by the framework. Additionally, I utilized dynamic data generation using TypeScript functions to create random test data. Test artifacts are also enclosed in the project.

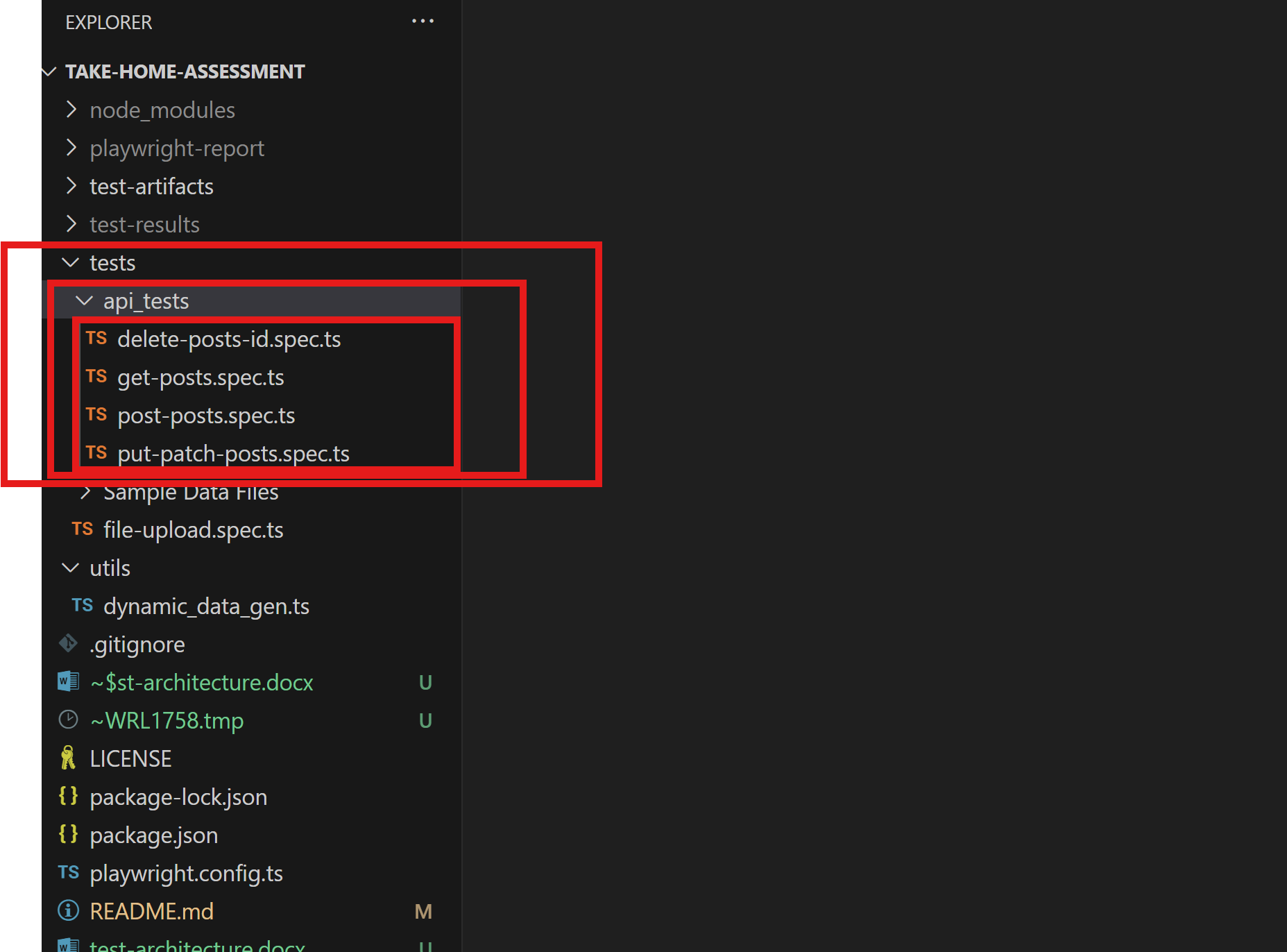
# UI Tests

For UI tests there is the “file-upload.spec.ts” file inside the tests directory which contains all the UI tests for the file upload. You can see the image below:



# API Tests

For API tests there is a separate folder named “api\_tests” inside the tests directory located in root directory. All the API test suites are inside that specific directory. You can see the image below:



# Test Data

The test data consists of two sections which are as follows:

## Files

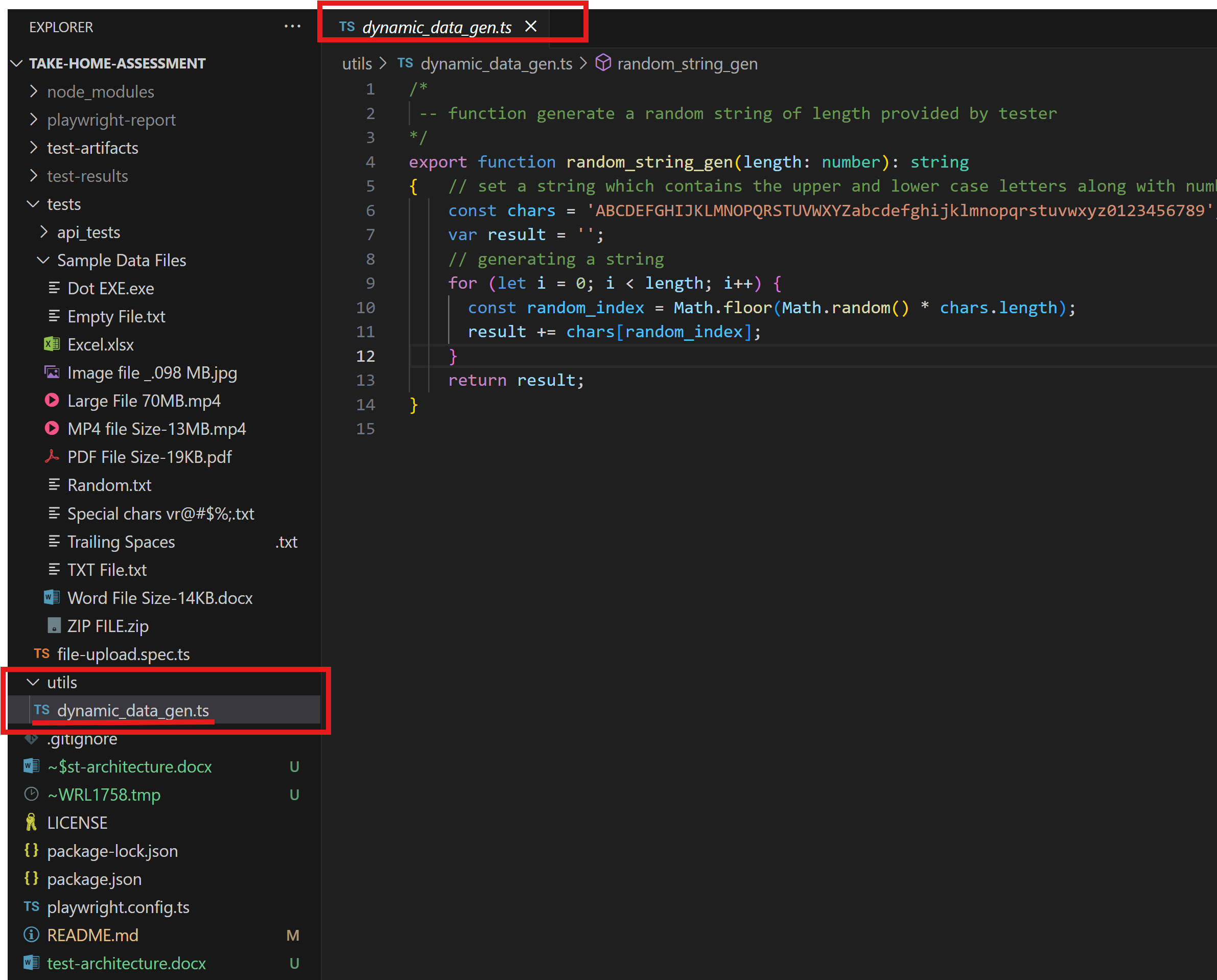
All the files are inside the folder named Sample Data Files, you can see below:

A screenshot of a computer

AI-generated content may be incorrect.

## Random Variables

For random variables generation there is a file named “dynamic\_data\_gen.ts” enclosed in the utils folder located in the root directory. You can see the image below:



# Test Report

The report is overwritten each time the tests are executed, and it is saved in the 'playwright-report' folder located in the root directory. You can see the image below:

