

# Accessibility Automation using Axe, Cucumber.JS, Selenium and JavaScript

- Create a new folder, **AccessibilityJSSelenium**, this will be our Project root
- Open this folder via VSCode
- In the VSCode terminal initiate a new NPM project – **npm init**. This will create a package.json file in the AccessibilityJSSelenium folder.
- Install Cucumber -- **npm install --save-dev @cucumber/cucumber**
- Install Selenium -- **npm install selenium-webdriver --save-dev**
- Install WebDriver Manager -- **npm install webdriver-manager --save-dev**
- Install Chai Assertion Library -- **npm install --save-dev chai**
- Install Axe Core -- **npm install @axe-core/webdriverjs --save-dev**
- Create a folder, **features**. This will contain all the project files.
- Create the feature files in the features folder or create multiple folders under **features/** which will then have the feature files.
- Create a subfolder, **features/step\_definitions**. This folder will have all the step definitions
- Create a subfolder, **features/reports**. This folder will store the detected Accessibility violations after a test is run.
- Create a subfolder **features/support** and create the **hooks.js**, **configs.js** and **selectors.js** files

## Running commands:

- **.\node\_modules\.bin\cucumber-js** – This will run all the feature files
- **.\node\_modules\.bin\cucumber-js --format html:.\features\reports\Report.html** – This will run all the feature files and store the report to the reports folder
- **.\node\_modules\.bin\cucumber-js .\features\feature\_addition\add2.feature --format html:.\features\reports\Report.html** – This will run the add2.feature file and store the generated report to the reports folder
- **.\node\_modules\.bin\cucumber.js -tags "@new" --format html:.\features\reports\Report.html** – This will run the scenario with @new tag and store the generated report to the reports folder

## Folder Structure:

