**Note** - The below steps are created assuming Java and Maven are already installed in the local machine. If they are not already installed please follow the instruction on How to Install Java for Java installation and How to Install Maven for Maven installation

**Execution Of Project Through Command Line**

1. Clone the project to a folder in the local machine
2. Open the command prompt on Windows or terminal on Mac
3. Navigate to the folder to which the project is cloned
4. Execute the following command “mvn clean install”
5. When building first time it will download the necessary dependencies and then build the  
   project
6. You will see all the tests are executed once the dependencies are downloaded
7. If you need to execute a single test only use the command “mvn clean test

-Dcucumber.options="--tags @tagName" where tagName is the tag used for the relevant test (Tag to the test scenario mapping could be found below)

**Execution Of Project Through IDE Using Runner Class**

1. Clone the project to a folder in the local machine
2. Open the IDE and import the project to the IDE
3. Open the test folder under the project structure
4. Open the TestRunner.java file
5. Change the tag name given within double quotes for tags to the desired tag name (Tag to the test scenario mapping could be found below)
6. Right click on TestRunner.java and Run As JUnit
7. The test will be executed and the results will be displayed in the IDE

**Execution Of Project Through IDE Using Maven**

1. Clone the project to a folder in the local machine
2. Open the IDE and import the project to the IDE
3. Right click on the project and Run As and select Maven Build...
4. In the screen that is opened enter the command “clean install” under goals and click run
5. The necessary dependencies will be downloaded and the project will be built
6. All the tests are executed when this command is used
7. If individual tests needs to be executed enter the command “mvn clean test

-Dcucumber.options="--tags @tagName” under goals and run (Tag to the test scenario mapping could be found below)

**Tag Name to Test Scenario mapping**

|  |  |  |
| --- | --- | --- |
| 1  page2image1607600 | @verifyLandingPage | Open the todomvc URL and verify the correct page is displayed |
| 2  page2image1632560 | @AddNewTask | Add single task in Todo List |
| 3  page2image1644000 | @AddMultipleTasks | Add multiple tasks in Todo List and verify the total task count added in the list |
| 4  page2image1645872 | @UpdateTask | Verify user is able to update the existing task name |
| 5  page2image1612176 | @DeleteTask | Verify user is able to delete the tasks from the list |
| 6  page2image1645248 | @TaskCompleted | Verify on checking the task, the task gets completed |
| 7  page2image1647536 | @ClearCompletedTask | Verify on clicking the Clear Completed button, the completed tasks gets removed from the list |