Test Scenarios:

**TS01:**

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.Test;

public class TodoAppTest {

    private TodoApp todoApp;

    @Before

    public void setUp() {

        todoApp = new TodoApp();

    }

    @Test

    public void testAddSingleTodo() {

        todoApp.addTodo("Drink Water");

        assertTrue(todoApp.getTodos().contains("Drink Water"));

        assertEquals(1, todoApp.getTodos().size());

    }

    @Test

    public void testAddMultipleTodos() {

        todoApp.addTodo("Drink Water");

        todoApp.addTodo("Exercise Daily");

        todoApp.addTodo("Meditate");

        assertEquals(3, todoApp.getTodos().size());

    }

@Test

public void testTodoCounter() {

        todoApp.addTodo("Clean the House");

        todoApp.addTodo("Study");

        assertEquals(2, todoApp.getTodoCount());

    }

    @Test

    public void testEmptyTodoShouldNotBeAdded() {

        todoApp.addTodo("");

        assertEquals(0, todoApp.getTodos().size());

    }

}

**TS02:**

import [org.openqa.selenium.By](http://org.openqa.selenium.by/);

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import java.util.List;

public class TodoTest {

    WebDriver driver;

    @BeforeClass

    public void setUp() {

        System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

        driver = new ChromeDriver();

        driver.manage().window().maximize();

        driver.get("https://todomvc.com/examples/angular/dist/browser/#/all");

    }

    @Test

    public void testMarkItemsCompleted() {

        List<WebElement> todoItems = driver.findElements(By.cssSelector(".todo-list li"));

        for (WebElement item : todoItems) {

            String label = item.getText();

            if (label.contains("Drink Water Every Hour") || label.contains("Exercise Daily")) {

                WebElement checkbox = item.findElement(By.cssSelector("input[type='checkbox']"));

                if (!checkbox.isSelected()) {

                    checkbox.click();

                }

            }

        }

        for (WebElement item : todoItems) {

            String label = item.getText();

            if (label.contains("Drink Water Every Hour") || label.contains("Exercise Daily")) {

                String className = item.getAttribute("class");

                Assert.assertTrue(className.contains("completed"),

                        label + " should be marked as completed.");

            }

        }

        WebElement counter = driver.findElement(By.className("todo-count"));

        String countText = counter.getText();

        Assert.assertTrue(countText.contains("2 items left"), "Expected '2 items left' but got: " + countText);

    }

    @AfterClass

    public void tearDown() {

        driver.quit();

    }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**TS03:**

import [org.openqa.selenium.By](http://org.openqa.selenium.by/);  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.interactions.Actions;  
import org.testng.Assert;  
import org.testng.annotations.AfterClass;  
import org.testng.annotations.BeforeClass;  
import org.testng.annotations.Test;

import java.util.List;

public class DeleteTodoTest {  
    WebDriver driver;

    @BeforeClass  
    public void setUp() {  
        System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");   
        driver = new ChromeDriver();  
        driver.manage().window().maximize();  
        driver.get("https://todomvc.com/examples/angular/dist/browser/#/all");   
    }

    @Test  
    public void testDeleteItem() {

        List<WebElement> todoItems = driver.findElements(By.cssSelector(".todo-list li"));

        for (WebElement item : todoItems) {  
            if (item.getText().contains("Clean the house")) {  
                Actions actions = new Actions(driver);  
                actions.moveToElement(item).perform();  
                WebElement deleteBtn = item.findElement(By.cssSelector("button.destroy"));  
                deleteBtn.click();  
                break;  
            }  
        }

        List<WebElement> updatedItems = driver.findElements(By.cssSelector(".todo-list li"));  
        boolean itemDeleted = updatedItems.stream()  
                .noneMatch(el -> el.getText().contains("Clean the house"));

        Assert.assertTrue(itemDeleted, "'Clean the house' should be deleted.");  
        WebElement counter = driver.findElement(By.className("todo-count"));  
        String counterText = counter.getText();  
        Assert.assertTrue(counterText.contains("1 item left"), "Expected '1 item left' but found: " + counterText);  
    }

@AfterClass  
    public void tearDown() {  
        driver.quit();  
    }  
}

**TS04:**

import [org.openqa.selenium.By](http://org.openqa.selenium.by/);  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.testng.Assert;  
import org.testng.annotations.\*;

import java.util.List;  
import java.util.stream.Collectors;

public class FilterTodoTest {  
    WebDriver driver;

    @BeforeClass  
    public void setUp() {  
        System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");   
        driver = new ChromeDriver();  
        driver.manage().window().maximize();  
        driver.get("https://todomvc.com/examples/angular/dist/browser/#/all");   
    }

    @Test  
    public void testFilterTodos() {  
        driver.findElement(By.linkText("Active")).click();  
        List<WebElement> activeTodos = driver.findElements(By.cssSelector(".todo-list li"));  
        List<WebElement> completedInActiveView = activeTodos.stream()  
                .filter(item -> item.getAttribute("class").contains("completed"))  
                .collect(Collectors.toList());

        Assert.assertTrue(completedInActiveView.isEmpty(), "Active view should not show completed items.");   
        driver.findElement(By.linkText("Completed")).click();  
        List<WebElement> completedTodos = driver.findElements(By.cssSelector(".todo-list li"));  
        List<WebElement> nonCompletedInCompletedView = completedTodos.stream()  
                .filter(item -> !item.getAttribute("class").contains("completed"))  
                .collect(Collectors.toList());

        Assert.assertTrue(nonCompletedInCompletedView.isEmpty(), "Completed view should not show active items.");  
    }

@AfterClass  
    public void tearDown() {  
        driver.quit();  
    }  
}

