Purpose:

The code is designed to automate tasks related to an election management web application. It uses Selenium WebDriver to interact with the web application and perform actions like deleting voters, adding voters, and fetching voter information.

Libraries and Dependencies:

The code relies on several libraries and dependencies:

- **Newtonsoft.Json**: Used for JSON serialization and deserialization.
- **OpenQA.Selenium**: Provides functionality for automating web browser interactions.
- **OpenQA.Selenium.Chrome**: A WebDriver implementation for Google Chrome.
- **System.Collections.ObjectModel**: Used for working with collections of web elements.
- **System.Reflection.Metadata**: Not actively used in the code.
- **System.Xml.Linq**: Used for working with XML documents.

Classes and Methods:

1. **Tests** Class:
    - This is the main class that contains the test methods for interacting with the web application.
2. **Setup** Method:

- Initializes the WebDriver (ChromeDriver) with specific options and maximizes the browser window.

3. **T1_DELETE** Method:
    - Navigates to a specific URL, logs in to the web application, and accesses the Voters section.
    - Repeatedly deletes voters by locating and clicking the delete button until no more such buttons are present.
4. **IsElementPresent** Method:
    - A helper method to check if an element specified by a given **By** locator exists on the web page.
5. **T2_ADD** Method:
    - Navigates to the main page, logs in, and accesses the Voters section.
    - Reads user data from a JSON file and adds each user as a voter.
6. **T3_FETCH** Method:
    - Navigates to the main page, logs in, and accesses the Voters section.
    - Reads user data from a JSON file and fetches voter information by searching for voters based on their first name and last name. It then updates the JSON file with voter IDs.
7. **UserData** Class:

- Represents the structure of user data, including first name, last name, password, and voter ID.

Flow of Execution:

1. The **Setup** method is called before each test method to set up the WebDriver and maximize the browser window.
2. Test methods (**T1_DELETE**, **T2_ADD**, **T3_FETCH**) perform specific actions on the web application.
3. The web application is logged into using a username and password.
4. Data is read from a JSON file (**userdata.json**) to interact with the application.
5. Depending on the test method, actions such as deleting voters, adding voters, or fetching voter information are performed.
6. The WebDriver is closed (**driver.Quit()**) at the end of each test method.

Usage:

1. Setting up the Environment:

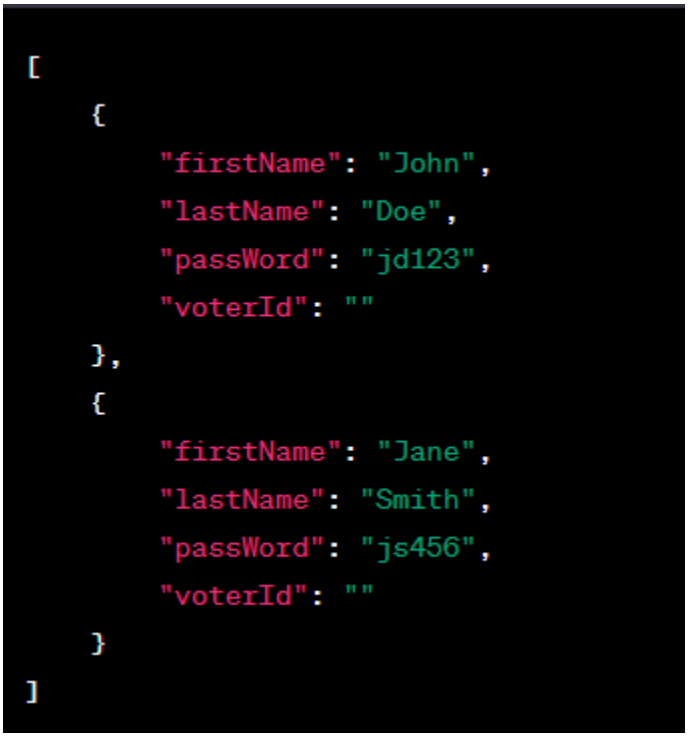
- Before running the code, ensure you have the necessary dependencies and libraries installed, such as .NET and Selenium WebDriver.
- Download and install the appropriate Chrome WebDriver for Selenium (ChromeDriver) and add its location to your system's PATH or provide the path explicitly in the code.

JSON Data Preparation:

- Create a JSON file named userdata.json that contains data about voters. Each entry should include the following fields:
    - firstName: First name of the voter.
    - lastName: Last name of the voter.
    - passWord: Password associated with the voter.
    - voterId: Voter ID (can be initially empty).

JSON Data:

The code expects user data to be stored in a JSON file (**userdata.json**). This data includes information about voters, such as their first name, last name, and password.



Note:

- The code includes explicit waits (**Thread.Sleep**) in some places to add delays between actions. These are used to ensure that the web page has fully loaded before interacting with elements. However, a more robust approach would be to use WebDriverWait for handling dynamic page elements.
- Paths to files and directories are hardcoded in the code. These should be parameterized or configured externally for better maintainability.
- Exception handling and error reporting are not included in this code. For production use, it's essential to implement error handling to handle unexpected issues gracefully.
- This code is written for educational purposes and may need adjustments to work with specific web applications or environments.
- Ensure that the necessary WebDriver and browser versions are compatible and installed on the system where the code is executed.