

Practical-1 CA(CSV302)

Name: Aman Kumar Kanu

Reg:12208695

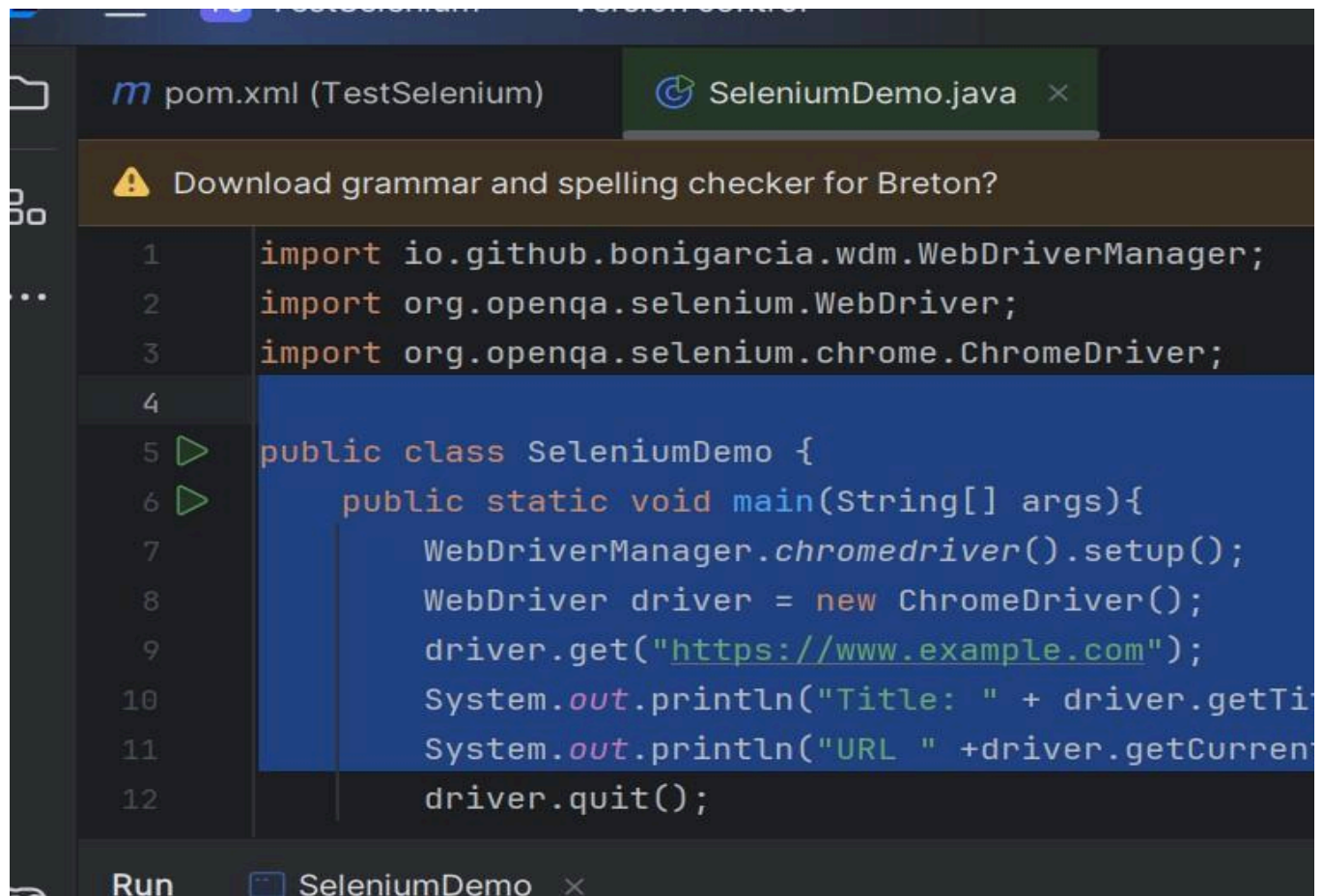
Roll.No:06

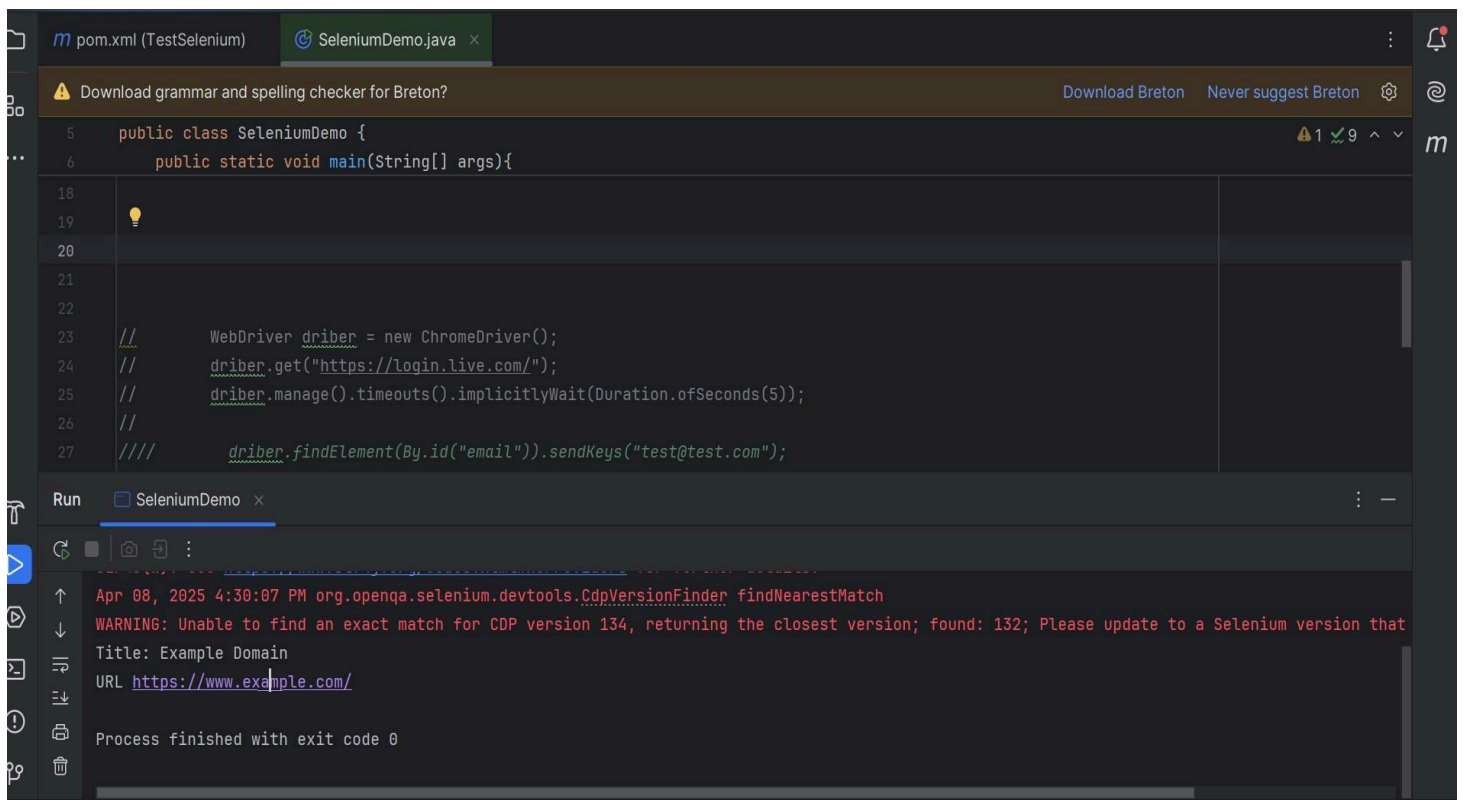
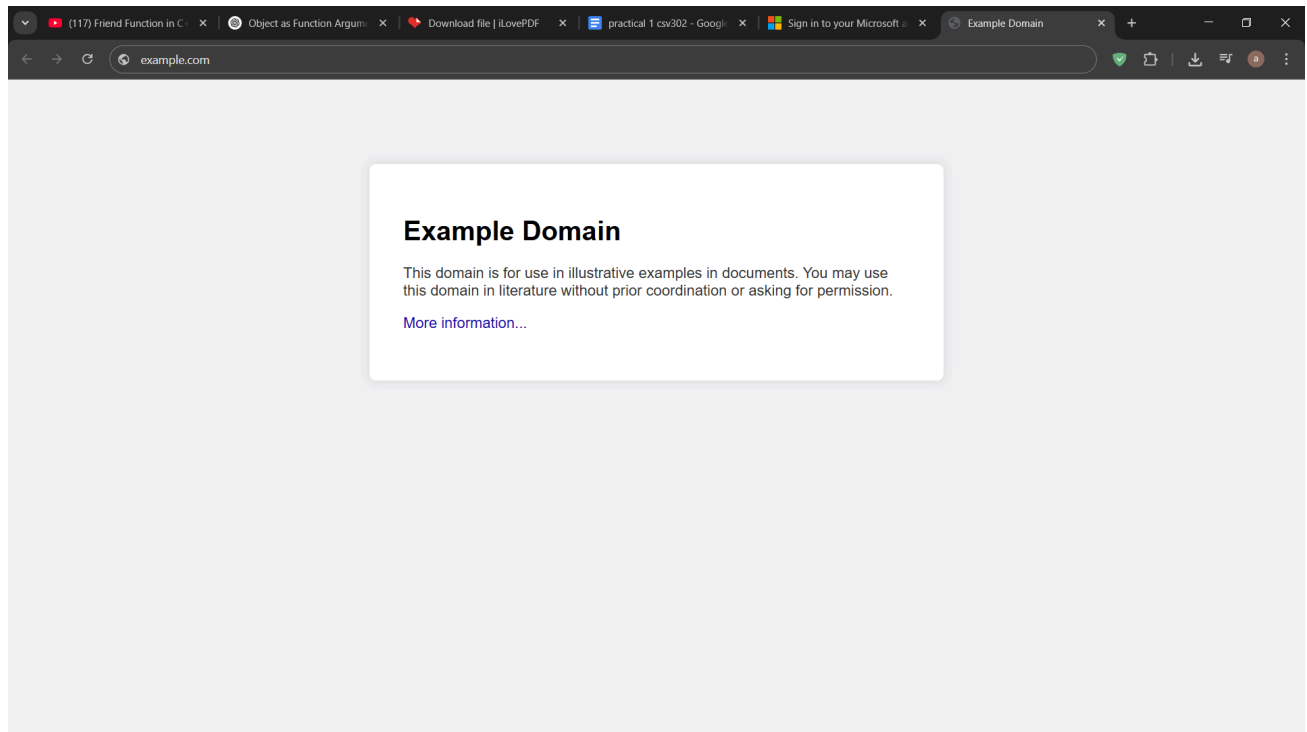
Step 1: Install Preferred IDE

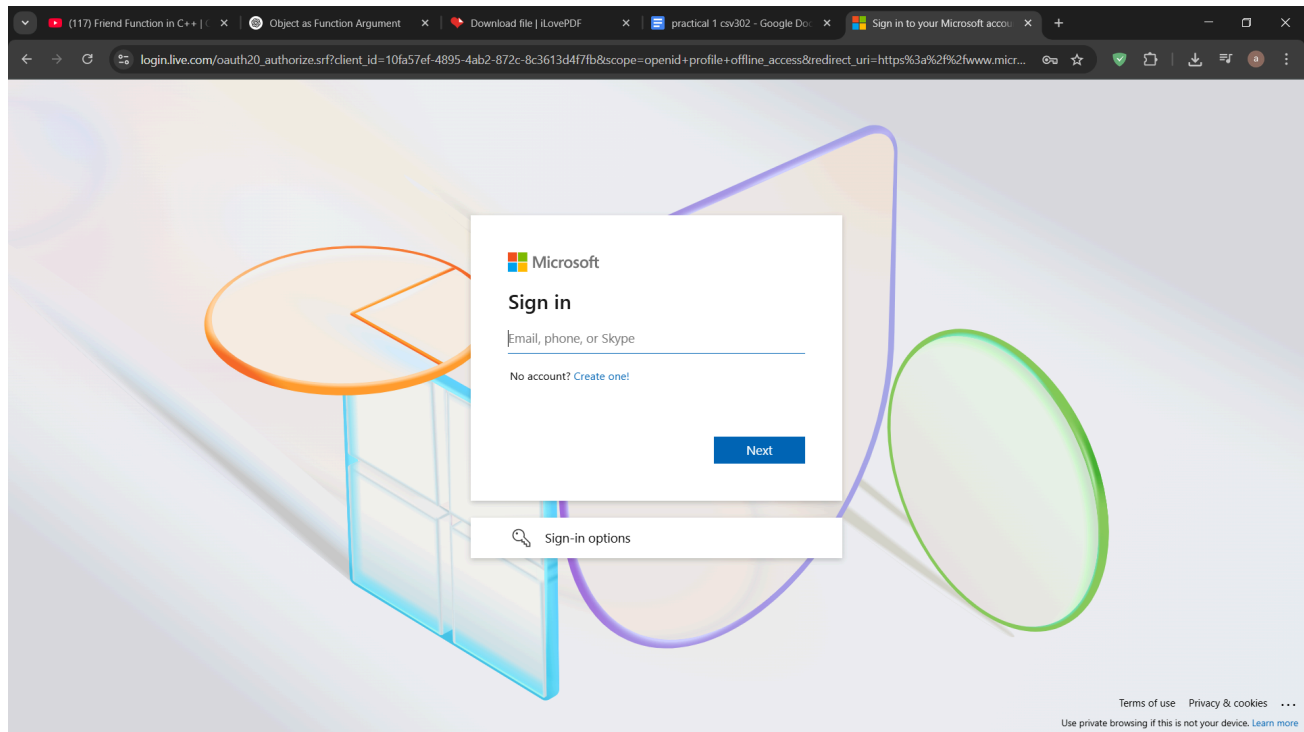
—>An IDE (Integrated Development Environment) helps you write and manage your code easily.

Step 2: Install Selenium Library

—>Selenium is not built-in, so we install it using pip:**pip install selenium**

A screenshot of an IDE window showing a Java file named SeleniumDemo.java. The code includes imports for WebDriverManager, WebDriver, and ChromeDriver, followed by a main method that sets up a ChromeDriver, navigates to a URL, prints the title and URL, and then quits the driver. A notification bar at the top asks to download a grammar and spelling checker for Breton. The bottom of the window shows a 'Run' button and the class name 'SeleniumDemo'.





Explanation of the practical:

◆ 1. Environment Setup

- Installed IDE: Used PyCharm to write and run Python code.
- Installed Selenium: Used `pip install selenium` to add Selenium to Python.
- Installed ChromeDriver: Downloaded ChromeDriver matching the Chrome browser version to allow browser automation.

2. Verification

- Created a simple script to open Chrome and visit `https://www.example.com`.
- Printed the page title to verify Selenium and ChromeDriver are working.
- Closed the browser.

3. Basic Script Creation

- Wrote a script that:
 - Opens Chrome.
 - Goes to a website.
 - Prints page title and URL.
 - Closes the browser smoothly.
 -

4. Element Interaction

- Opened Google.
- Located the search box using its name.
- Typed a keyword (e.g., “Selenium WebDriver”).
- Printed an attribute of the search box like `title`.

5. Documentation & Challenges

- Faced an issue with version mismatch (ChromeDriver vs Chrome browser).
- Solved it by downloading the correct ChromeDriver version.
- Added small delays (`sleep`) to avoid element loading issues.