#working code 1

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# ✅ Setup WebDriver with best practices

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--disable-blink-features=AutomationControlled")

options.add\_argument("--start-maximized")

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

service = Service("drivers\chromedriver.exe")  # Ensure correct ChromeDriver path

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://njdg.ecourts.gov.in/njdg\_v3/")

# ✅ Wait object for dynamic element handling

wait = WebDriverWait(driver, 30)

try:

    # ✅ Click the second Disposal Dashboard tab

    dashboard\_buttons = wait.until(EC.presence\_of\_all\_elements\_located((By.ID, "penDash1-tab")))

    if len(dashboard\_buttons) >= 2:

        element\_to\_click = dashboard\_buttons[1]  # Second button

    else:

        element\_to\_click = dashboard\_buttons[0]  # Fallback to first

    driver.execute\_script("arguments[0].click();", element\_to\_click)

    print("✅ Clicked Disposal Dashboard.")

    # ✅ Click "Agewise Table Button"

    wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn"))).click()

    print("✅ Clicked Agewise Table Button.")

    # ✅ Click "Agewise Chart Button"

    wait.until(EC.element\_to\_be\_clickable((By.ID, "agewisechartbtn"))).click()

    print("✅ Clicked Agewise Chart Button.")

    # ✅ Switch back to table view

    wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn"))).click()

    print("✅ Reopened Agewise Table after switching views.")

    time.sleep(5)

    # ✅ Locate search filter and enter "2024"

    for \_ in range(3):  # Retry mechanism

        try:

            search\_box = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblAgewisecount\_filter input")))

            search\_box.clear()

            search\_box.send\_keys("2024")

            break  # Exit loop if successful

        except:

            driver.refresh()

            time.sleep(3)

    print("✅ Entered '2024' in search filter.")

    time.sleep(5)

    # ✅ Click statewise case details

    statewise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='statewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", statewise\_link)

    print("✅ Clicked statewise details link.")

    time.sleep(5)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_statewiseCase\_data")))

    print("✅ Modal opened successfully!")

    # ✅ Search for "Karnataka" in Statewise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblStatewisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Karnataka")

    print("✅ Entered 'Karnataka' in search.")

    time.sleep(5)

    # ✅ Click district-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='distwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked district-wise details link.")

    time.sleep(5)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_distwiseCase\_data")))

    print("✅ District-wise Modal opened successfully!")

    # ✅ Search "Bangalore" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tbldistwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Bengaluru")

    print("✅ Entered 'Bengaluru' in search.")

    # ✅ Wait for results to update

    time.sleep(5)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='estwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked City-wise details link.")

    time.sleep(5)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_estwiseCase\_data")))

    print("✅ City-wise Modal opened successfully!")

    # ✅ Search "PRL. CITY CIVIL AND SESSIONS JUDGE" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("prl. city civil and sessions judge")

    print("✅ Entered 'PRL. CITY CIVIL AND SESSIONS JUDGE' in search.")

    # ✅ Wait for results to update

    time.sleep(5)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(5)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    csv\_download\_button = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a.buttons-csv")))

    driver.execute\_script("arguments[0].click();", csv\_download\_button)

    print("✅ Clicked on CSV download button.")

    time.sleep(5)

    driver.save\_screenshot("districtwise\_table\_bengaluru.png")

except Exception as e:

    print("❌ Error:", e)

    driver.save\_screenshot("error.png")

# ✅ Keep browser open until user exits

input("Press Enter to exit...")

driver.quit()

Working code 2: With JSON downloadables

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# ✅ Setup WebDriver with best practices

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--disable-blink-features=AutomationControlled")

options.add\_argument("--start-maximized")

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

service = Service("drivers\chromedriver.exe")  # Ensure correct ChromeDriver path

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://njdg.ecourts.gov.in/njdg\_v3/")

# ✅ Wait object for dynamic element handling

wait = WebDriverWait(driver, 30)

try:

    # ✅ Click the second Disposal Dashboard tab

    dashboard\_buttons = wait.until(EC.presence\_of\_all\_elements\_located((By.ID, "penDash1-tab")))

    if len(dashboard\_buttons) >= 2:

        element\_to\_click = dashboard\_buttons[1]  # Second button

    else:

        element\_to\_click = dashboard\_buttons[0]  # Fallback to first

    driver.execute\_script("arguments[0].click();", element\_to\_click)

    print("✅ Clicked Disposal Dashboard.")

    # ✅ Click "Agewise Table Button"

    time.sleep(2)

    age\_table\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn")))

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Clicked Agewise Table Button.")

    # ✅ Click "Agewise Chart Button"

    time.sleep(2)

    chart\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewisechartbtn")))

    driver.execute\_script("arguments[0].click();", chart\_btn)

    print("✅ Clicked Agewise Chart Button.")

    # ✅ Switch back to table view

    time.sleep(2)

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Reopened Agewise Table after switching views.")

    time.sleep(5)  # Wait for the search filter to load

    # ✅ Locate the search filter and enter "2024"

    search\_box = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblAgewisecount\_filter input")))

    search\_box.clear()

    search\_box.send\_keys("2024")

    print("✅ Entered '2024' in the search filter.")

    time.sleep(5)  # Wait for table update

    # ✅ Click statewise case details

    statewise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='statewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", statewise\_link)

    print("✅ Clicked statewise details link.")

    time.sleep(5)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_statewiseCase\_data")))

    print("✅ Modal opened successfully!")

    # ✅ Search for "Karnataka" in Statewise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblStatewisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Karnataka")

    print("✅ Entered 'Karnataka' in search.")

    time.sleep(5)

    # ✅ Click district-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='distwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked district-wise details link.")

    time.sleep(5)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_distwiseCase\_data")))

    print("✅ District-wise Modal opened successfully!")

    # ✅ Search "Bangalore" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tbldistwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Bengaluru")

    print("✅ Entered 'Bengaluru' in search.")

    # ✅ Wait for results to update

    time.sleep(5)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='estwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked City-wise details link.")

    time.sleep(5)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_estwiseCase\_data")))

    print("✅ City-wise Modal opened successfully!")

    # ✅ Search "PRL. CITY CIVIL AND SESSIONS JUDGE" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("prl. city civil and sessions judge")

    print("✅ Entered 'PRL. CITY CIVIL AND SESSIONS JUDGE' in search.")

    # ✅ Wait for results to update

    time.sleep(5)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(5)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    # ✅ Validate modal opens

    import json

    modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

    # ✅ Find all table rows inside the modal

    rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

    extracted\_data = []

    for row in rows:

        columns = row.find\_elements(By.TAG\_NAME, "td")

        row\_data = [col.text.strip() for col in columns]  # Extract text from each column

        if row\_data:

            extracted\_data.append(row\_data)

    # ✅ Convert to JSON

    json\_data = json.dumps(extracted\_data, indent=4)

    with open("case\_details\_2.json", "w", encoding="utf-8") as json\_file:

        json\_file.write(json\_data)

    print("✅ JSON Data Extracted and Saved as 'case\_details\_2.json'!")

    time.sleep(5)

    driver.save\_screenshot("casewise\_table.png")

    time.sleep(5)

    driver.save\_screenshot("districtwise\_table\_bengaluru.png")

except Exception as e:

    print("❌ Error:", e)

    driver.save\_screenshot("error.png")

# ✅ Keep browser open until user exits

input("Press Enter to exit...")

driver.quit()

Working code 3:

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# ✅ Setup WebDriver with best practices

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--disable-blink-features=AutomationControlled")

options.add\_argument("--start-maximized")

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

service = Service("drivers\chromedriver.exe")  # Ensure correct ChromeDriver path

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://njdg.ecourts.gov.in/njdg\_v3/")

# ✅ Wait object for dynamic element handling

wait = WebDriverWait(driver, 30)

try:

    # ✅ Click the second Disposal Dashboard tab

    dashboard\_buttons = wait.until(EC.presence\_of\_all\_elements\_located((By.ID, "penDash1-tab")))

    if len(dashboard\_buttons) >= 2:

        element\_to\_click = dashboard\_buttons[1]  # Second button

    else:

        element\_to\_click = dashboard\_buttons[0]  # Fallback to first

    driver.execute\_script("arguments[0].click();", element\_to\_click)

    print("✅ Clicked Disposal Dashboard.")

    # ✅ Click "Agewise Table Button"

    time.sleep(2)

    age\_table\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn")))

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Clicked Agewise Table Button.")

    # ✅ Click "Agewise Chart Button"

    time.sleep(2)

    chart\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewisechartbtn")))

    driver.execute\_script("arguments[0].click();", chart\_btn)

    print("✅ Clicked Agewise Chart Button.")

    # ✅ Switch back to table view

    time.sleep(2)

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Reopened Agewise Table after switching views.")

    time.sleep(2)  # Wait for the search filter to load

    # ✅ Locate the search filter and enter "2024"

    search\_box = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblAgewisecount\_filter input")))

    search\_box.clear()

    search\_box.send\_keys("2024")

    print("✅ Entered '2024' in the search filter.")

    time.sleep(2)  # Wait for table update

    # ✅ Click statewise case details

    statewise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='statewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", statewise\_link)

    print("✅ Clicked statewise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_statewiseCase\_data")))

    print("✅ Modal opened successfully!")

    # ✅ Search for "Karnataka" in Statewise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblStatewisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Karnataka")

    print("✅ Entered 'Karnataka' in search.")

    time.sleep(2)

    # ✅ Click district-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='distwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked district-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_distwiseCase\_data")))

    print("✅ District-wise Modal opened successfully!")

    # ✅ Search "Bangalore" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tbldistwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Bengaluru")

    print("✅ Entered 'Bengaluru' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='estwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked City-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_estwiseCase\_data")))

    print("✅ City-wise Modal opened successfully!")

    # ✅ Search "PRL. CITY CIVIL AND SESSIONS JUDGE" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("prl. city civil and sessions judge")

    print("✅ Entered 'PRL. CITY CIVIL AND SESSIONS JUDGE' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    # ✅ Validate modal opens

    import json

    modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

    # ✅ Find all table rows inside the modal

    rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

    extracted\_data = []

    for row in rows:

        columns = row.find\_elements(By.TAG\_NAME, "td")

        row\_data = [col.text.strip() for col in columns]  # Extract text from each column

        if row\_data:

            extracted\_data.append(row\_data)

    # ✅ Convert to JSON

    json\_data = json.dumps(extracted\_data, indent=4)

    with open("case\_details.json", "w", encoding="utf-8") as json\_file:

        json\_file.write(json\_data)

    print("✅ JSON Data Extracted and Saved as 'case\_details.json'!")

    time.sleep(2)

    driver.save\_screenshot("casewise\_table.png")

    time.sleep(2)

    driver.save\_screenshot("districtwise\_table\_bengaluru.png")

except Exception as e:

    print("❌ Error:", e)

    driver.save\_screenshot("error.png")

# ✅ Keep browser open until user exits

input("Press Enter to exit...")

driver.quit()

working code 4

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# ✅ Setup WebDriver with best practices

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--disable-blink-features=AutomationControlled")

options.add\_argument("--start-maximized")

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

service = Service("drivers\chromedriver.exe")  # Ensure correct ChromeDriver path

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://njdg.ecourts.gov.in/njdg\_v3/")

# ✅ Wait object for dynamic element handling

wait = WebDriverWait(driver, 30)

try:

    # ✅ Click the second Disposal Dashboard tab

    dashboard\_buttons = wait.until(EC.presence\_of\_all\_elements\_located((By.ID, "penDash1-tab")))

    if len(dashboard\_buttons) >= 2:

        element\_to\_click = dashboard\_buttons[1]  # Second button

    else:

        element\_to\_click = dashboard\_buttons[0]  # Fallback to first

    driver.execute\_script("arguments[0].click();", element\_to\_click)

    print("✅ Clicked Disposal Dashboard.")

    # ✅ Click "Agewise Table Button"

    time.sleep(2)

    age\_table\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn")))

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Clicked Agewise Table Button.")

    # ✅ Click "Agewise Chart Button"

    time.sleep(2)

    chart\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewisechartbtn")))

    driver.execute\_script("arguments[0].click();", chart\_btn)

    print("✅ Clicked Agewise Chart Button.")

    # ✅ Switch back to table view

    time.sleep(2)

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Reopened Agewise Table after switching views.")

    time.sleep(2)  # Wait for the search filter to load

    # ✅ Locate the search filter and enter "2024"

    search\_box = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblAgewisecount\_filter input")))

    search\_box.clear()

    search\_box.send\_keys("2024")

    print("✅ Entered '2024' in the search filter.")

    time.sleep(2)  # Wait for table update

    # ✅ Click statewise case details

    statewise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='statewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", statewise\_link)

    print("✅ Clicked statewise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_statewiseCase\_data")))

    print("✅ Modal opened successfully!")

    # ✅ Search for "Karnataka" in Statewise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblStatewisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Karnataka")

    print("✅ Entered 'Karnataka' in search.")

    time.sleep(2)

    # ✅ Click district-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='distwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked district-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_distwiseCase\_data")))

    print("✅ District-wise Modal opened successfully!")

    # ✅ Search "Bangalore" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tbldistwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Bengaluru")

    print("✅ Entered 'Bengaluru' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='estwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked City-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_estwiseCase\_data")))

    print("✅ City-wise Modal opened successfully!")

    # ✅ Search "PRL. CITY CIVIL AND SESSIONS JUDGE" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("prl. city civil and sessions judge")

    print("✅ Entered 'PRL. CITY CIVIL AND SESSIONS JUDGE' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    # ✅ Validate modal opens

    import json

    modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

    # ✅ Find all table rows inside the modal

    rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

    extracted\_data = []

    for row in rows:

        columns = row.find\_elements(By.TAG\_NAME, "td")

        row\_data = [col.text.strip() for col in columns]  # Extract text from each column

        if row\_data:

            extracted\_data.append(row\_data)

    # ✅ Function to extract JSON from modal and append it to file

    def extract\_and\_append\_json():

        modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

        rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

        extracted\_data = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            row\_data = [col.text.strip() for col in columns]  # Extract text from each column

            if row\_data:

                extracted\_data.append(row\_data)

        # ✅ Append JSON data to the file instead of overwriting

        try:

            with open("case\_details.json", "r", encoding="utf-8") as file:

                existing\_data = json.load(file)

        except (FileNotFoundError, json.decoder.JSONDecodeError):

            existing\_data = []

        existing\_data.extend(extracted\_data)

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print("✅ JSON Data Extracted and Appended to 'case\_details.json'!")

    # ✅ Extract data from the first page before clicking "Next"

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for the new data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

    # # Wait for the button to be present in the DOM

    back\_button = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "button[data-bs-target='#modal\_estwiseCase\_data']")))

    # Click the button using JavaScript Executor

    driver.execute\_script("arguments[0].click();", back\_button)

    print("✅ Clicked the 'Back' button using JavaScript Executor.")

except Exception as e:

    print("❌ Error:", e)

    driver.save\_screenshot("error.png")

# ✅ Keep browser open until user exits

input("Press Enter to exit...")

driver.quit()

working code 5

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# ✅ Setup WebDriver with best practices

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--disable-blink-features=AutomationControlled")

options.add\_argument("--start-maximized")

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

service = Service("drivers\chromedriver.exe")  # Ensure correct ChromeDriver path

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://njdg.ecourts.gov.in/njdg\_v3/")

# ✅ Wait object for dynamic element handling

wait = WebDriverWait(driver, 30)

try:

    # ✅ Click the second Disposal Dashboard tab

    dashboard\_buttons = wait.until(EC.presence\_of\_all\_elements\_located((By.ID, "penDash1-tab")))

    if len(dashboard\_buttons) >= 2:

        element\_to\_click = dashboard\_buttons[1]  # Second button

    else:

        element\_to\_click = dashboard\_buttons[0]  # Fallback to first

    driver.execute\_script("arguments[0].click();", element\_to\_click)

    print("✅ Clicked Disposal Dashboard.")

    # ✅ Click "Agewise Table Button"

    time.sleep(2)

    age\_table\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn")))

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Clicked Agewise Table Button.")

    # ✅ Click "Agewise Chart Button"

    time.sleep(2)

    chart\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewisechartbtn")))

    driver.execute\_script("arguments[0].click();", chart\_btn)

    print("✅ Clicked Agewise Chart Button.")

    # ✅ Switch back to table view

    time.sleep(2)

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Reopened Agewise Table after switching views.")

    time.sleep(2)  # Wait for the search filter to load

    # ✅ Locate the search filter and enter "2024"

    search\_box = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblAgewisecount\_filter input")))

    search\_box.clear()

    search\_box.send\_keys("2024")

    print("✅ Entered '2024' in the search filter.")

    time.sleep(2)  # Wait for table update

    # ✅ Click statewise case details

    statewise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='statewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", statewise\_link)

    print("✅ Clicked statewise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_statewiseCase\_data")))

    print("✅ Modal opened successfully!")

    # ✅ Search for "Karnataka" in Statewise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblStatewisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Karnataka")

    print("✅ Entered 'Karnataka' in search.")

    time.sleep(2)

    # ✅ Click district-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='distwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked district-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_distwiseCase\_data")))

    print("✅ District-wise Modal opened successfully!")

    # ✅ Search "Bangalore" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tbldistwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Bengaluru")

    print("✅ Entered 'Bengaluru' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='estwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked City-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_estwiseCase\_data")))

    print("✅ City-wise Modal opened successfully!")

    # ✅ Search "PRL. CITY CIVIL AND SESSIONS JUDGE" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("prl. city civil and sessions judge")

    print("✅ Entered 'PRL. CITY CIVIL AND SESSIONS JUDGE' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    # ✅ Validate modal opens

    import json

    modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

    # ✅ Find all table rows inside the modal

    rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

    extracted\_data = []

    for row in rows:

        columns = row.find\_elements(By.TAG\_NAME, "td")

        row\_data = [col.text.strip() for col in columns]  # Extract text from each column

        if row\_data:

            extracted\_data.append(row\_data)

    # ✅ Function to extract JSON from modal and append it to file

    def extract\_and\_append\_json():

        modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

        rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

        extracted\_data = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            row\_data = [col.text.strip() for col in columns]  # Extract text from each column

            if row\_data:

                extracted\_data.append(row\_data)

        # ✅ Append JSON data to the file instead of overwriting

        try:

            with open("case\_details.json", "r", encoding="utf-8") as file:

                existing\_data = json.load(file)

        except (FileNotFoundError, json.decoder.JSONDecodeError):

            existing\_data = []

        existing\_data.extend(extracted\_data)

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print("✅ JSON Data Extracted and Appended to 'case\_details.json'!")

    # ✅ Extract data from the first page before clicking "Next"

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for the new data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

    # # Wait for the button to be present in the DOM

    back\_button = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "button[data-bs-target='#modal\_estwiseCase\_data']")))

    # Click the button using JavaScript Executor

    driver.execute\_script("arguments[0].click();", back\_button)

    print("✅ Clicked the 'Back' button using JavaScript Executor.")

   # ✅ Locate and clear the search input field

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    # Scroll into view before interacting

    driver.execute\_script("arguments[0].scrollIntoView(true);", search\_statewise)

    time.sleep(1)  # Allow time for scrolling

    # Clear and enter text

    search\_statewise.clear()

    search\_statewise.send\_keys("addl. city civil and sessions judges, mayo hall, bengaluru")

    print("✅ Entered text after scrolling.")

    print("✅ Entered 'ADDL. CITY CIVIL AND SESSIONS JUDGES, MAYO HALL, BENGALURU' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    # ✅ Validate modal opens

    import json

    modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

    # ✅ Find all table rows inside the modal

    rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

    extracted\_data = []

    for row in rows:

        columns = row.find\_elements(By.TAG\_NAME, "td")

        row\_data = [col.text.strip() for col in columns]  # Extract text from each column

        if row\_data:

            extracted\_data.append(row\_data)

    # ✅ Function to extract JSON from modal and append it to file

    def extract\_and\_append\_json():

        modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

        rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

        extracted\_data = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            row\_data = [col.text.strip() for col in columns]  # Extract text from each column

            if row\_data:

                extracted\_data.append(row\_data)

        # ✅ Append JSON data to the file instead of overwriting

        try:

            with open("case\_details.json", "r", encoding="utf-8") as file:

                existing\_data = json.load(file)

        except (FileNotFoundError, json.decoder.JSONDecodeError):

            existing\_data = []

        existing\_data.extend(extracted\_data)

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print("✅ JSON Data Extracted and Appended to 'case\_details.json'!")

    # ✅ Extract data from the first page before clicking "Next"

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for the new data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

except Exception as e:

    print("❌ Error:", e)

    driver.save\_screenshot("error.png")

# ✅ Keep browser open until user exits

input("Press Enter to exit...")

driver.quit()

working code 6 ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# ✅ Setup WebDriver with best practices

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--disable-blink-features=AutomationControlled")

options.add\_argument("--start-maximized")

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

service = Service("drivers\chromedriver.exe")  # Ensure correct ChromeDriver path

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://njdg.ecourts.gov.in/njdg\_v3/")

# ✅ Wait object for dynamic element handling

wait = WebDriverWait(driver, 30)

try:

    # ✅ Click the second Disposal Dashboard tab

    dashboard\_buttons = wait.until(EC.presence\_of\_all\_elements\_located((By.ID, "penDash1-tab")))

    if len(dashboard\_buttons) >= 2:

        element\_to\_click = dashboard\_buttons[1]  # Second button

    else:

        element\_to\_click = dashboard\_buttons[0]  # Fallback to first

    driver.execute\_script("arguments[0].click();", element\_to\_click)

    print("✅ Clicked Disposal Dashboard.")

    # ✅ Click "Agewise Table Button"

    time.sleep(2)

    age\_table\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn")))

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Clicked Agewise Table Button.")

    # ✅ Click "Agewise Chart Button"

    time.sleep(2)

    chart\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewisechartbtn")))

    driver.execute\_script("arguments[0].click();", chart\_btn)

    print("✅ Clicked Agewise Chart Button.")

    # ✅ Switch back to table view

    time.sleep(2)

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Reopened Agewise Table after switching views.")

    time.sleep(2)  # Wait for the search filter to load

    # ✅ Locate the search filter and enter "2024"

    search\_box = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblAgewisecount\_filter input")))

    search\_box.clear()

    search\_box.send\_keys("2024")

    print("✅ Entered '2024' in the search filter.")

    time.sleep(2)  # Wait for table update

    # ✅ Click statewise case details

    statewise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='statewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", statewise\_link)

    print("✅ Clicked statewise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_statewiseCase\_data")))

    print("✅ Modal opened successfully!")

    # ✅ Search for "Karnataka" in Statewise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblStatewisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Karnataka")

    print("✅ Entered 'Karnataka' in search.")

    time.sleep(2)

    # ✅ Click district-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='distwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked district-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_distwiseCase\_data")))

    print("✅ District-wise Modal opened successfully!")

    # ✅ Search "Bangalore" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tbldistwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Bengaluru")

    print("✅ Entered 'Bengaluru' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='estwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked City-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_estwiseCase\_data")))

    print("✅ City-wise Modal opened successfully!")

    # ✅ Search "PRL. CITY CIVIL AND SESSIONS JUDGE" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("prl. city civil and sessions judge")

    print("✅ Entered 'PRL. CITY CIVIL AND SESSIONS JUDGE' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    # ✅ Validate modal opens

    import json

    modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

    # ✅ Find all table rows inside the modal

    rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

    extracted\_data = []

    for row in rows:

        columns = row.find\_elements(By.TAG\_NAME, "td")

        row\_data = [col.text.strip() for col in columns]  # Extract text from each column

        if row\_data:

            extracted\_data.append(row\_data)

    # ✅ Function to extract JSON from modal and append it to file

    def extract\_and\_append\_json():

        modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

        rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

        extracted\_data = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            row\_data = [col.text.strip() for col in columns]  # Extract text from each column

            if row\_data:

                extracted\_data.append(row\_data)

        # ✅ Append JSON data to the file instead of overwriting

        try:

            with open("case\_details.json", "r", encoding="utf-8") as file:

                existing\_data = json.load(file)

        except (FileNotFoundError, json.decoder.JSONDecodeError):

            existing\_data = []

        existing\_data.extend(extracted\_data)

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print("✅ JSON Data Extracted and Appended to 'case\_details.json'!")

    # ✅ Extract data from the first page before clicking "Next"

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for the new data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

    # # Wait for the button to be present in the DOM

    back\_button = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "button[data-bs-target='#modal\_estwiseCase\_data']")))

    # Click the button using JavaScript Executor

    driver.execute\_script("arguments[0].click();", back\_button)

    print("✅ Clicked the 'Back' button using JavaScript Executor.")

   # ✅ Locate and clear the search input field

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    # Scroll into view before interacting

    driver.execute\_script("arguments[0].scrollIntoView(true);", search\_statewise)

    time.sleep(1)  # Allow time for scrolling

    # Clear and enter text

    search\_statewise.clear()

    search\_statewise.send\_keys("addl. city civil and sessions judges, mayo hall, bengaluru")

    print("✅ Entered text after scrolling.")

    print("✅ Entered 'ADDL. CITY CIVIL AND SESSIONS JUDGES, MAYO HALL, BENGALURU' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    # ✅ Validate modal opens

    import json

    modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

    # ✅ Find all table rows inside the modal

    rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

    extracted\_data = []

    for row in rows:

        columns = row.find\_elements(By.TAG\_NAME, "td")

        row\_data = [col.text.strip() for col in columns]  # Extract text from each column

        if row\_data:

            extracted\_data.append(row\_data)

    # ✅ Function to extract JSON from modal and append it to file

    def extract\_and\_append\_json():

        modal\_body = wait.until(EC.presence\_of\_element\_located((By.ID, "modal-body-dashcaselist")))

        rows = modal\_body.find\_elements(By.TAG\_NAME, "tr")

        extracted\_data = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            row\_data = [col.text.strip() for col in columns]  # Extract text from each column

            if row\_data:

                extracted\_data.append(row\_data)

        # ✅ Append JSON data to the file instead of overwriting

        try:

            with open("case\_details.json", "r", encoding="utf-8") as file:

                existing\_data = json.load(file)

        except (FileNotFoundError, json.decoder.JSONDecodeError):

            existing\_data = []

        existing\_data.extend(extracted\_data)

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print("✅ JSON Data Extracted and Appended to 'case\_details.json'!")

    # ✅ Extract data from the first page before clicking "Next"

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for the new data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

except Exception as e:

    print("❌ Error:", e)

    driver.save\_screenshot("error.png")

# ✅ Keep browser open until user exits

input("Press Enter to exit...")

driver.quit()

Working code 7:

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# ✅ Setup WebDriver with best practices

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--disable-blink-features=AutomationControlled")

options.add\_argument("--start-maximized")

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

service = Service("drivers\chromedriver.exe")  # Ensure correct ChromeDriver path

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://njdg.ecourts.gov.in/njdg\_v3/")

# ✅ Wait object for dynamic element handling

wait = WebDriverWait(driver, 30)

try:

    # ✅ Click the second Disposal Dashboard tab

    dashboard\_buttons = wait.until(EC.presence\_of\_all\_elements\_located((By.ID, "penDash1-tab")))

    if len(dashboard\_buttons) >= 2:

        element\_to\_click = dashboard\_buttons[1]  # Second button

    else:

        element\_to\_click = dashboard\_buttons[0]  # Fallback to first

    driver.execute\_script("arguments[0].click();", element\_to\_click)

    print("✅ Clicked Disposal Dashboard.")

    # ✅ Click "Agewise Table Button"

    time.sleep(2)

    age\_table\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn")))

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Clicked Agewise Table Button.")

    # ✅ Click "Agewise Chart Button"

    time.sleep(2)

    chart\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewisechartbtn")))

    driver.execute\_script("arguments[0].click();", chart\_btn)

    print("✅ Clicked Agewise Chart Button.")

    # ✅ Switch back to table view

    time.sleep(2)

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Reopened Agewise Table after switching views.")

    time.sleep(2)  # Wait for the search filter to load

    # ✅ Locate the search filter and enter "2024"

    search\_box = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblAgewisecount\_filter input")))

    search\_box.clear()

    search\_box.send\_keys("2024")

    print("✅ Entered '2024' in the search filter.")

    time.sleep(2)  # Wait for table update

    # ✅ Click statewise case details

    statewise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='statewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", statewise\_link)

    print("✅ Clicked statewise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_statewiseCase\_data")))

    print("✅ Modal opened successfully!")

    # ✅ Search for "Karnataka" in Statewise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblStatewisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Karnataka")

    print("✅ Entered 'Karnataka' in search.")

    time.sleep(2)

    # ✅ Click district-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='distwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked district-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_distwiseCase\_data")))

    print("✅ District-wise Modal opened successfully!")

    # ✅ Search "Bangalore" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tbldistwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Bengaluru")

    print("✅ Entered 'Bengaluru' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='estwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked City-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_estwiseCase\_data")))

    print("✅ City-wise Modal opened successfully!")

    # ✅ Search "PRL. CITY CIVIL AND SESSIONS JUDGE" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("prl. city civil and sessions judge")

    print("✅ Entered 'PRL. CITY CIVIL AND SESSIONS JUDGE' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    import json

    # ✅ Extract Establishment Name (from search field)

    establishment\_field = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    establishment\_name = establishment\_field.get\_attribute("value").strip()

    print(f"✅ Extracted Court Name: {establishment\_name}")

    # ✅ Extract table headers dynamically

    header\_elements = driver.find\_elements(By.CSS\_SELECTOR, "#tbl\_cases\_dash thead th")

    headers = [header.text.strip() for header in header\_elements]

    print("✅ Extracted headers:", headers)

    # ✅ Define function to extract and append JSON

    def extract\_and\_append\_json():

        rows = driver.find\_elements(By.CSS\_SELECTOR, "#tbl\_cases\_dash tbody tr")

        extracted\_cases = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            case\_data = {headers[i]: columns[i].text.strip() for i in range(len(columns))}

            extracted\_cases.append(case\_data)

        # ✅ JSON format with court grouping

        formatted\_json = {

            "Sr. No": len(existing\_data) + 1,  # Increment based on existing data

            "Establishment": establishment\_name,

            "Count": extracted\_cases

        }

        existing\_data.append(formatted\_json)

        # ✅ Save JSON to file

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print("✅ JSON Data Extracted and Appended!")

    # ✅ Load existing data (to avoid overwriting)

    try:

        with open("case\_details.json", "r", encoding="utf-8") as file:

            existing\_data = json.load(file)

    except (FileNotFoundError, json.decoder.JSONDecodeError):

        existing\_data = []

    # ✅ Extract first page data before pagination

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

    ##########################################

    # # Wait for the button to be present in the DOM

    back\_button = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "button[data-bs-target='#modal\_estwiseCase\_data']")))

    # Click the button using JavaScript Executor

    driver.execute\_script("arguments[0].click();", back\_button)

    print("✅ Clicked the 'Back' button using JavaScript Executor.")

    #########################################

   # ✅ Locate and clear the search input field

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    # Scroll into view before interacting

    driver.execute\_script("arguments[0].scrollIntoView(true);", search\_statewise)

    time.sleep(1)  # Allow time for scrolling

    # Clear and enter text

    search\_statewise.clear()

    search\_statewise.send\_keys("addl. city civil and sessions judges, mayo hall, bengaluru")

    print("✅ Entered text after scrolling.")

    print("✅ Entered 'ADDL. CITY CIVIL AND SESSIONS JUDGES, MAYO HALL, BENGALURU' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    # ✅ Extract Establishment Name (from search field)

    establishment\_field = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    establishment\_name = establishment\_field.get\_attribute("value").strip()

    print(f"✅ Extracted Court Name: {establishment\_name}")

    # ✅ Extract table headers dynamically

    header\_elements = driver.find\_elements(By.CSS\_SELECTOR, "#tbl\_cases\_dash thead th")

    headers = [header.text.strip() for header in header\_elements]

    print("✅ Extracted headers:", headers)

    # ✅ Define function to extract and append JSON

    def extract\_and\_append\_json():

        rows = driver.find\_elements(By.CSS\_SELECTOR, "#tbl\_cases\_dash tbody tr")

        extracted\_cases = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            case\_data = {headers[i]: columns[i].text.strip() for i in range(len(columns))}

            extracted\_cases.append(case\_data)

        # ✅ JSON format with court grouping

        formatted\_json = {

            "Sr. No": len(existing\_data) + 1,  # Increment based on existing data

            "Establishment": establishment\_name,

            "Count": extracted\_cases

        }

        existing\_data.append(formatted\_json)

        # ✅ Save JSON to file

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print("✅ JSON Data Extracted and Appended!")

    # ✅ Load existing data (to avoid overwriting)

    try:

        with open("case\_details.json", "r", encoding="utf-8") as file:

            existing\_data = json.load(file)

    except (FileNotFoundError, json.decoder.JSONDecodeError):

        existing\_data = []

    # ✅ Extract first page data before pagination

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

except Exception as e:

    print("❌ Error:", e)

    driver.save\_screenshot("error.png")

# ✅ Keep browser open until user exits

input("Press Enter to exit...")

driver.quit()

Working code 8: completed and deugged for correct json format using search(not loops yet)

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# ✅ Setup WebDriver with best practices

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--disable-blink-features=AutomationControlled")

options.add\_argument("--start-maximized")

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

service = Service("drivers\chromedriver.exe")  # Ensure correct ChromeDriver path

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://njdg.ecourts.gov.in/njdg\_v3/")

# ✅ Wait object for dynamic element handling

wait = WebDriverWait(driver, 30)

try:

    # ✅ Click the second Disposal Dashboard tab

    dashboard\_buttons = wait.until(EC.presence\_of\_all\_elements\_located((By.ID, "penDash1-tab")))

    if len(dashboard\_buttons) >= 2:

        element\_to\_click = dashboard\_buttons[1]  # Second button

    else:

        element\_to\_click = dashboard\_buttons[0]  # Fallback to first

    driver.execute\_script("arguments[0].click();", element\_to\_click)

    print("✅ Clicked Disposal Dashboard.")

    # ✅ Click "Agewise Table Button"

    time.sleep(2)

    age\_table\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewiseTblbtn")))

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Clicked Agewise Table Button.")

    # ✅ Click "Agewise Chart Button"

    time.sleep(2)

    chart\_btn = wait.until(EC.element\_to\_be\_clickable((By.ID, "agewisechartbtn")))

    driver.execute\_script("arguments[0].click();", chart\_btn)

    print("✅ Clicked Agewise Chart Button.")

    # ✅ Switch back to table view

    time.sleep(2)

    driver.execute\_script("arguments[0].click();", age\_table\_btn)

    print("✅ Reopened Agewise Table after switching views.")

    time.sleep(2)  # Wait for the search filter to load

    # ✅ Locate the search filter and enter "2024"

    search\_box = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblAgewisecount\_filter input")))

    search\_box.clear()

    search\_box.send\_keys("2024")

    print("✅ Entered '2024' in the search filter.")

    time.sleep(2)  # Wait for table update

    # ✅ Click statewise case details

    statewise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='statewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", statewise\_link)

    print("✅ Clicked statewise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_statewiseCase\_data")))

    print("✅ Modal opened successfully!")

    # ✅ Search for "Karnataka" in Statewise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblStatewisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Karnataka")

    print("✅ Entered 'Karnataka' in search.")

    time.sleep(2)

    # ✅ Click district-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='distwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked district-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_distwiseCase\_data")))

    print("✅ District-wise Modal opened successfully!")

    # ✅ Search "Bangalore" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tbldistwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("Bengaluru")

    print("✅ Entered 'Bengaluru' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='estwise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked City-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_estwiseCase\_data")))

    print("✅ City-wise Modal opened successfully!")

    # ✅ Search "PRL. CITY CIVIL AND SESSIONS JUDGE" in District-wise Table

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    search\_statewise.clear()

    search\_statewise.send\_keys("prl. city civil and sessions judge")

    print("✅ Entered 'PRL. CITY CIVIL AND SESSIONS JUDGE' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    import json

    import os

    # ✅ Load existing data

    try:

        with open("case\_details.json", "r", encoding="utf-8") as file:

            existing\_data = json.load(file)

    except (FileNotFoundError, json.decoder.JSONDecodeError):

        existing\_data = []

    # ✅ Extract Establishment Name

    establishment\_field = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    establishment\_name = establishment\_field.get\_attribute("value").strip()

    print(f"✅ Extracted Court Name: {establishment\_name}")

    # ✅ Extract table headers dynamically

    header\_elements = driver.find\_elements(By.CSS\_SELECTOR, "#tbl\_cases\_dash thead th")

    headers = [header.text.strip() for header in header\_elements]

    print("✅ Extracted headers:", headers)

    # ✅ Find if establishment already exists

    establishment\_entry = next((item for item in existing\_data if item["Establishment"] == establishment\_name), None)

    if not establishment\_entry:

        # ✅ If establishment does not exist, create a new one with an incremented Sr. No

        establishment\_sr\_no = len(existing\_data) + 1

        establishment\_entry = {

            "Sr. No": establishment\_sr\_no,

            "Establishment": establishment\_name,

            "Count": []

        }

        existing\_data.append(establishment\_entry)

    else:

        establishment\_sr\_no = establishment\_entry["Sr. No"]  # Keep existing Sr. No

    # ✅ Get the next available "Sr. No." for cases in "Count"

    next\_case\_sr\_no = len(establishment\_entry["Count"]) + 1

    # ✅ Function to extract and append JSON

    def extract\_and\_append\_json():

        global next\_case\_sr\_no  # Ensure the Sr. No. continues across pages

        rows = driver.find\_elements(By.CSS\_SELECTOR, "#tbl\_cases\_dash tbody tr")

        extracted\_cases = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            case\_data = {headers[i]: columns[i].text.strip() for i in range(len(columns))}

            # ✅ Assign unique "Sr. No." to each case

            case\_data["Sr. No."] = str(next\_case\_sr\_no)

            next\_case\_sr\_no += 1  # Increment for next case

            extracted\_cases.append(case\_data)

        # ✅ Append new cases to the correct establishment

        establishment\_entry["Count"].extend(extracted\_cases)

        # ✅ Save JSON to file

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print(f"✅ JSON Data Extracted and Appended for {establishment\_name}!")

    # ✅ Extract first page data before pagination

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for new data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

    ##########################################

    # # Wait for the button to be present in the DOM

    back\_button = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "button[data-bs-target='#modal\_estwiseCase\_data']")))

    # Click the button using JavaScript Executor

    driver.execute\_script("arguments[0].click();", back\_button)

    print("✅ Clicked the 'Back' button using JavaScript Executor.")

    #########################################

   # ✅ Locate and clear the search input field

    search\_statewise = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    # Scroll into view before interacting

    driver.execute\_script("arguments[0].scrollIntoView(true);", search\_statewise)

    time.sleep(1)  # Allow time for scrolling

    # Clear and enter text

    search\_statewise.clear()

    search\_statewise.send\_keys("addl. city civil and sessions judges, mayo hall, bengaluru")

    print("✅ Entered text after scrolling.")

    print("✅ Entered 'ADDL. CITY CIVIL AND SESSIONS JUDGES, MAYO HALL, BENGALURU' in search.")

    # ✅ Wait for results to update

    time.sleep(2)

    # ✅ Click City-wise case details

    districtwise\_link = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "a[onclick\*='casewise\_case\_details']")))

    driver.execute\_script("arguments[0].click();", districtwise\_link)

    print("✅ Clicked Case-wise details link.")

    time.sleep(2)

    # ✅ Validate modal opens

    wait.until(EC.visibility\_of\_element\_located((By.ID, "modal\_dashcaselist")))

    print("✅ Case-wise Modal opened successfully!")

    time.sleep(3)  # Small delay before clicking

    ############

    import json

    import os

    # ✅ Load existing data

    try:

        with open("case\_details.json", "r", encoding="utf-8") as file:

            existing\_data = json.load(file)

    except (FileNotFoundError, json.decoder.JSONDecodeError):

        existing\_data = []

    # ✅ Extract Establishment Name

    establishment\_field = wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, "#tblestwisecount\_filter input")))

    establishment\_name = establishment\_field.get\_attribute("value").strip()

    print(f"✅ Extracted Court Name: {establishment\_name}")

    # ✅ Extract table headers dynamically

    header\_elements = driver.find\_elements(By.CSS\_SELECTOR, "#tbl\_cases\_dash thead th")

    headers = [header.text.strip() for header in header\_elements]

    print("✅ Extracted headers:", headers)

    # ✅ Find if establishment already exists

    establishment\_entry = next((item for item in existing\_data if item["Establishment"] == establishment\_name), None)

    if not establishment\_entry:

        # ✅ If establishment does not exist, create a new one with an incremented Sr. No

        establishment\_sr\_no = len(existing\_data) + 1

        establishment\_entry = {

            "Sr. No": establishment\_sr\_no,

            "Establishment": establishment\_name,

            "Count": []

        }

        existing\_data.append(establishment\_entry)

    else:

        establishment\_sr\_no = establishment\_entry["Sr. No"]  # Keep existing Sr. No

    # ✅ Get the next available "Sr. No." for cases in "Count"

    next\_case\_sr\_no = len(establishment\_entry["Count"]) + 1

    # ✅ Function to extract and append JSON

    def extract\_and\_append\_json():

        global next\_case\_sr\_no  # Ensure the Sr. No. continues across pages

        rows = driver.find\_elements(By.CSS\_SELECTOR, "#tbl\_cases\_dash tbody tr")

        extracted\_cases = []

        for row in rows:

            columns = row.find\_elements(By.TAG\_NAME, "td")

            case\_data = {headers[i]: columns[i].text.strip() for i in range(len(columns))}

            # ✅ Assign unique "Sr. No." to each case

            case\_data["Sr. No."] = str(next\_case\_sr\_no)

            next\_case\_sr\_no += 1  # Increment for next case

            extracted\_cases.append(case\_data)

        # ✅ Append new cases to the correct establishment

        establishment\_entry["Count"].extend(extracted\_cases)

        # ✅ Save JSON to file

        with open("case\_details.json", "w", encoding="utf-8") as json\_file:

            json.dump(existing\_data, json\_file, indent=4)

        print(f"✅ JSON Data Extracted and Appended for {establishment\_name}!")

    # ✅ Extract first page data before pagination

    extract\_and\_append\_json()

    # ✅ Click "Next" button until disabled

    while True:

        try:

            next\_button = driver.find\_element(By.ID, "tbl\_cases\_dash\_next")

            if "disabled" in next\_button.get\_attribute("class"):

                print("✅ Reached the last page. Stopping pagination.")

                break

            driver.execute\_script("arguments[0].click();", next\_button)

            print("✅ Clicked 'Next' button. Fetching more data...")

            time.sleep(3)  # Wait for new data to load

            extract\_and\_append\_json()

        except Exception as e:

            print("❌ No 'Next' button found or pagination ended.", e)

            break

    print("✅ Pagination completed. All pages processed.")

except Exception as e:

    print("❌ Error:", e)

    driver.save\_screenshot("error.png")

# ✅ Keep browser open until user exits

input("Press Enter to exit...")

driver.quit()

Working 9: