

Environment Setup:

- Python: Version 3.11
- Robot Framework: Version 7.x
- SeleniumLibrary: Latest compatible with Robot Framework 7.x
- Web Browser: Firefox (latest stable release)
- WebDriver: GeckoDriver (compatible with the installed Firefox version)

Note: script must be cross-platform and function correctly on both Windows and Linux systems.

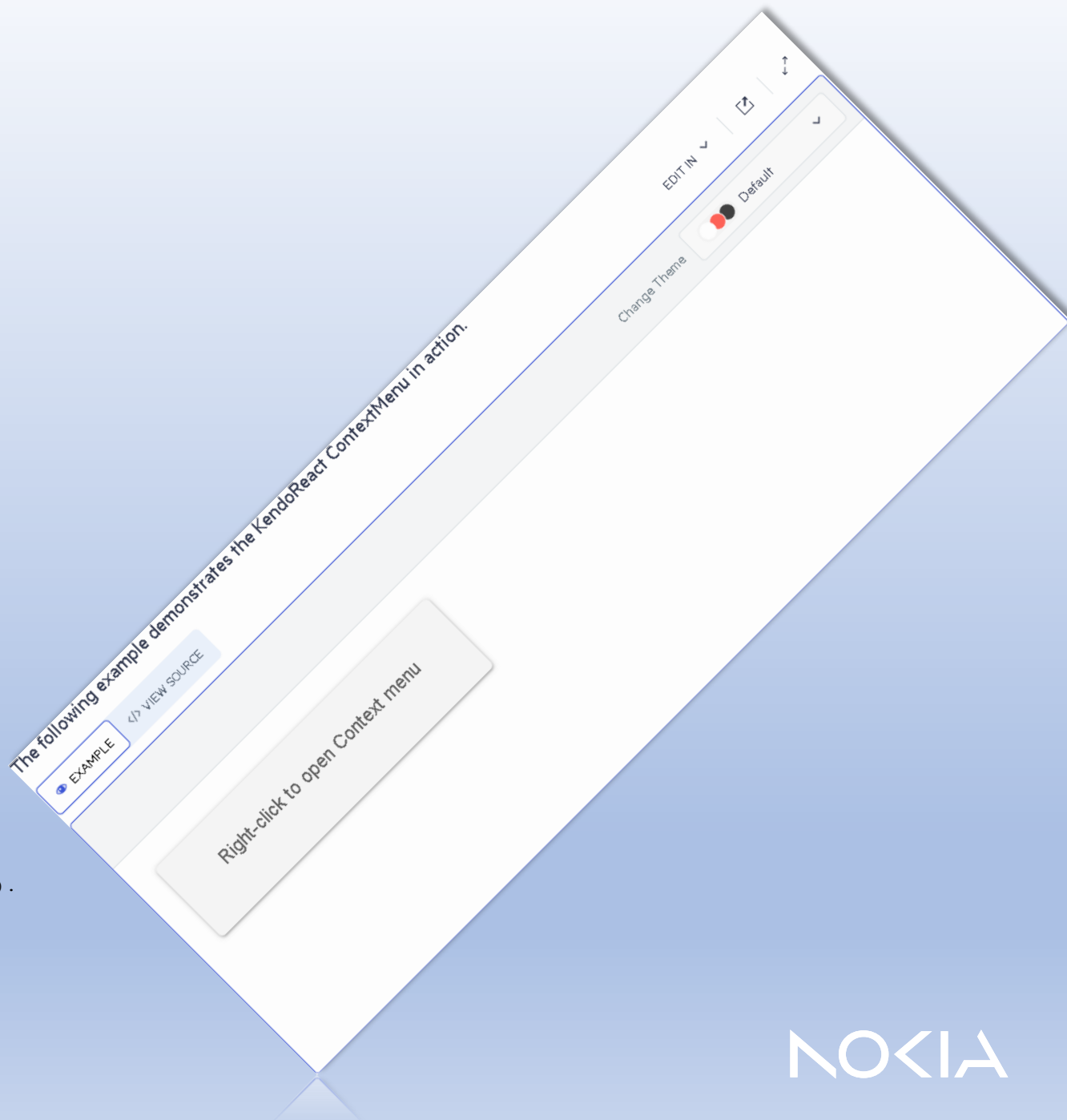
Task:

Develop a Robot Framework test script that automates the following steps:

1. Open the Firefox browser and navigate to:
<https://www.telerik.com/kendo-react-ui/components/layout/contextmenu>
2. Locate and scroll into view the example element in the section:
"KendoReact ContextMenu Overview"
3. Change the element theme, using dropdown menu, to:
"MATERIAL THEME → Main"
4. Modify the style of the inner element labeled:
"Right-click to open Context menu"
 - Apply underline styling to this text.
5. Capture a screenshot of the entire page after modifications.

Technical Requirements:

- Use XPath exclusively for element locators.
- Do not hardcode any file paths (e.g., browser, geckodriver, or screenshot locations).
- Include an option to run in headless mode (via command-line argument or variable).
- Ensure the script is clean, modular, and readable.



```
+-----+
|           Python OOP and Testing Task           |
+-----+
```

Environment Setup:

- Environment used in the previous task
- + pytest
- + pytest-mock

Task:

Create a second version of the Robot Framework script from the base task, with the following enhancements:

1. Refactor the script to use two custom keywords:
 - One for changing the theme
 - One for interacting with the context menu
2. Implement these custom keywords in a custom Python library.
3. Define and use dedicated Python classes to encapsulate the page interactions:
 - DropdownElement: handles theme selection functionality
 - ContextMenu: handles operations related to the context menu element
4. Write unit tests for the above classes using:
 - pytest as the test framework
 - mocker for mocking Selenium WebDriver interactions

Note: Unit tests must work without launching a real browser.

Requirements:

- The Robot test case must perform the same actions as in the previous task.
- The Python custom keywords must internally use the defined classes for DOM interaction.
- XPath should be used for all selectors within the Python classes.
- Unit tests must test method behavior using mocks (e.g., mock WebElement, mock WebDriver).
- Do not hardcode file paths or browser driver locations.
- Script must be cross-platform (Windows and Linux).
- Provide an option to run in headless mode.

```
+-----+
```





Good Luck!