**Task: Design BDD Automation Framework Using Page object Model**

Please take the following page <https://10.20.3.2/> to design an BDD automation framework for the given page:

1. **UI Analysis & Framework Design**  
   Review all UI controls, buttons, and components on the page thoroughly. Based on this analysis, design the BDD automation framework using the **Page object** model.
2. **BDD Test Cases**
   * Create a two feature files one for login and write **BDD-style only one login positive test case** (Given/When/Then) for each UI control and button.
   * Use test data for login only provided where enter user id =ADTest\adtest10 and password= Adtest#Xor@1684
   * Create another feature file including following test steps
   * Click on Xornet logo
   * Then after hover mouse on apps and tools primary menu
   * After that hover mouse on Revamp Innovation Portal sub menu
   * Finally click on Innovation Feedback menu
3. **Page & Test Class Structure**
   * Create a **Page class** to encapsulate the web elements and actions for each page.
   * Create a corresponding **Test class** for each page (if there are multiple pages).
4. **Logging Setup**
   * Create a **log folder** within the project directory.
   * Generate a **log file** for each test execution.
   * The log file should capture **each action and step performed on the screen** and be saved using the **current date and timestamp** in the filename.
5. **Reporting Structure**
   * Create a **Report folder** containing the following two subfolders:
     + Execution Report
     + Screenshot
6. **Extent Report Generation**
   * Generate an **Extent Report** that includes:
     + Pass, Fail, Blocked, and Not Run test case statuses.
   * Save the report in the **Execution Report** folder with the **current date and time** in the filename.
7. **Screenshot Capture**
   * Capture **screenshots** of each screen and each major action performed.
   * Save these screenshots in the **Screenshot** folder, each with the **current date and timestamp** in the filename.
8. **Browser & Driver Handling**
   * Ensure the **browser is closed** after the test execution.
   * Finally, **quit the driver** once all processes and reporting are completed.

Make sure here I am expecting code coverage with 100% of code quality and browser should open and close quickly. All tasks must be generated with one-time single efforts and without multiple rounds and without multiple corrections.