Task 1: Requirements Analysis and Coverage

Based on the detailed requirements for the workplace search web-form, I would like to propose Testcases and test data as below

# **Requirements Analysis.**

# **Search Functionality (Requirement 1, 2, 3)**

* **Test Data:** Various combinations of dates, floors, equipment, and smoking restrictions.
* **Test Cases:**
  + 1. Search with all fields empty (to test requirement 2).
    2. Search with each field individually and in combination.
    3. Search with invalid data (e.g., past dates, invalid floor numbers).
    4. Search with edge case data (e.g., today's date for date range, highest and lowest floor numbers).
    5. Perform a search with each field set to its most common value.
    6. Perform a search with extreme values in each field (e.g., highest floor, largest date range).
    7. Perform searches with special characters and unusual inputs in text fields.
    8. Perform a search with conflicting criteria (e.g., smoking and non-smoking selected simultaneously).
    9. Perform a search with rapidly changing criteria (to test system stability).

# **Date Range Validation (Requirement 4)**

* **Test Data:** Past dates, current date, future dates.
* **Test Cases:**
  + 1. Search with a date range that includes past dates.
    2. Search with a date range starting from the current date.
    3. Search with a date range in the future.
    4. Search with a single-day range for both past and future dates.
    5. Search with a date range spanning multiple months or years.
    6. Search with date formats in different cultural formats (e.g., DD/MM/YYYY vs. MM/DD/YYYY).
    7. Attempt to manipulate the date field through browser developer tools to enter past dates.

# **Sorting Search Results (Requirement 5)**

* **Test Data:** Search results based on different criteria.
* **Test Cases:**
  + 1. Sort results by each available field (date, floor, equipment, smoking restrictions)
    2. Validate sorting order (ascending and descending).
    3. Sort results by each field and then perform a secondary sort (e.g., first by date, then by floor).
    4. Sort results, then change search criteria and verify sorting persists.
    5. Sort results, navigate away from the page, then return and check if the sorting order is maintained.

# **Data Validation (Requirement 6)**

* **Test Cases:**
  + 1. Enter invalid data in each field and attempt a search.
    2. Validate error messages or indications for incorrect data.
    3. Enter excessively long strings in text fields.
    4. Enter HTML or JavaScript code in text fields to check for script injection vulnerabilities.
    5. Use automation tools to send a high volume of searches with random invalid data.
    6. Test with leading and trailing spaces in text fields
    7. Sorting Search Results (Requirement 5).

# **Performance (Requirement 7)**

* **Test Cases:**
  + 1. Measure response time for searches.
    2. Ensure search results are displayed promptly.
    3. Perform searches during peak and off-peak hours to measure any variance in response time.
    4. Use network throttling tools to simulate slower internet connections.
    5. Perform searches with the maximum allowable data in each field to test system load handling.

# **6. Navigation (Requirement 8)**

* **Test Cases:**
  + 1. Test pagination or scrolling functionality.
    2. Validate easy transition between pages of results.
    3. Test with different devices and screen sizes to ensure consistent navigation experience.
    4. Use keyboard navigation only (no mouse) to ensure accessibility.
    5. Test the back button functionality after navigating through multiple result pages.

**7. User Interface (Requirement 9)**

* **Test Cases:**
  + 1. Check for intuitive layout and design.
    2. Validate ease of use for setting search criteria and viewing results.
    3. Test on different browsers and browser versions for UI consistency.
    4. Validate that all text is legible, including placeholder text in search fields.
    5. Conduct usability testing with users unfamiliar with the system to assess intuitiveness.

**8. Read-Only Search Result Table (Requirement 10)**

* **Test Cases:**
  + 1. Verify that search result table cannot be modified.
    2. Attempt to edit, delete or add data in the search result table.
    3. Try to interact with the table through keyboard shortcuts.
    4. Test if any hyperlinks in the search results are clickable and lead to appropriate actions.

# **Identifying Bad Requirements**

* **Requirement 7 ("without delay")** is vague. It should specify a maximum acceptable delay.
* **Requirement 9 ("user-friendly interface")** is subjective. Specific UI/UX standards or guidelines should be defined.
* **Requirement 10** could be clarified to define the extent of 'read-only' - whether it pertains to data manipulation or also includes actions like exporting data.

# **Assumptions**

* The range of valid floor numbers is not specified. Assuming floors 1-10 for testing.
* Equipment types are not detailed. Assuming standard office equipment like projectors, computers, etc.
* Smoking restrictions: Assuming options are 'smoking allowed', 'non-smoking', and 'no preference'.