



Andriy Bahayev

Manual Tester Portfolio

PROJECT GOAL

The goal of the project is to test the to-do list app as a part of the "Testowanie Manualne Aplikacji" (Manual Software Testing) course by ifs&whiles. Tests are conducted based on provided business documentation and mock-ups. All bugs found on the platform are being documented in Trello table. As a part of the portfolio only three test cases are presented, the rest will be added to the GitHub repository.

Documentation and mock-ups:

https://github.com/ABfeMate/manual_tester_portfolio/tree/main/to_do_app_documentation

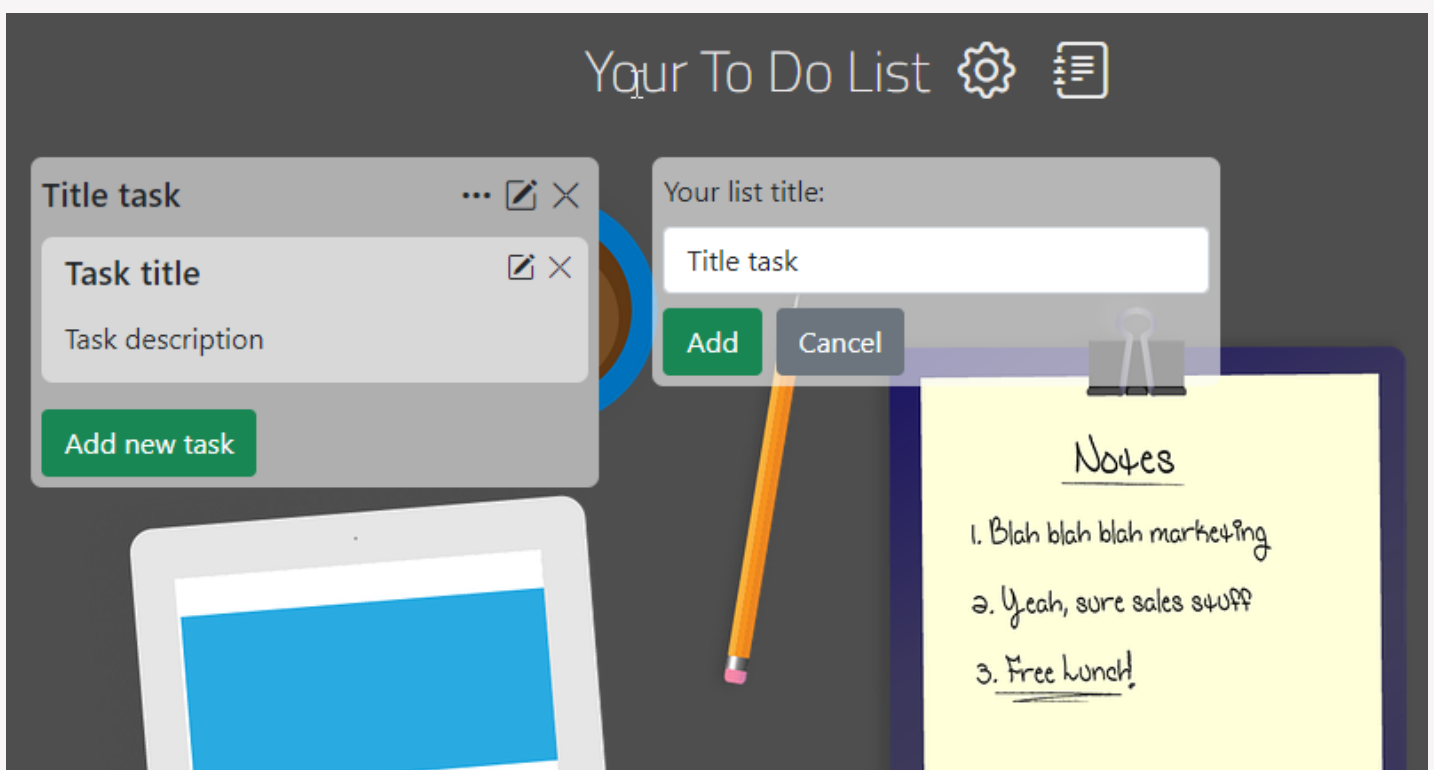
Trello:

<https://trello.com/invite/b/LpHrtg1C/ATTI977d3d20750843d06a5f22baa7d3e9bbF4F7D9BC/manual-tester-portfolio>

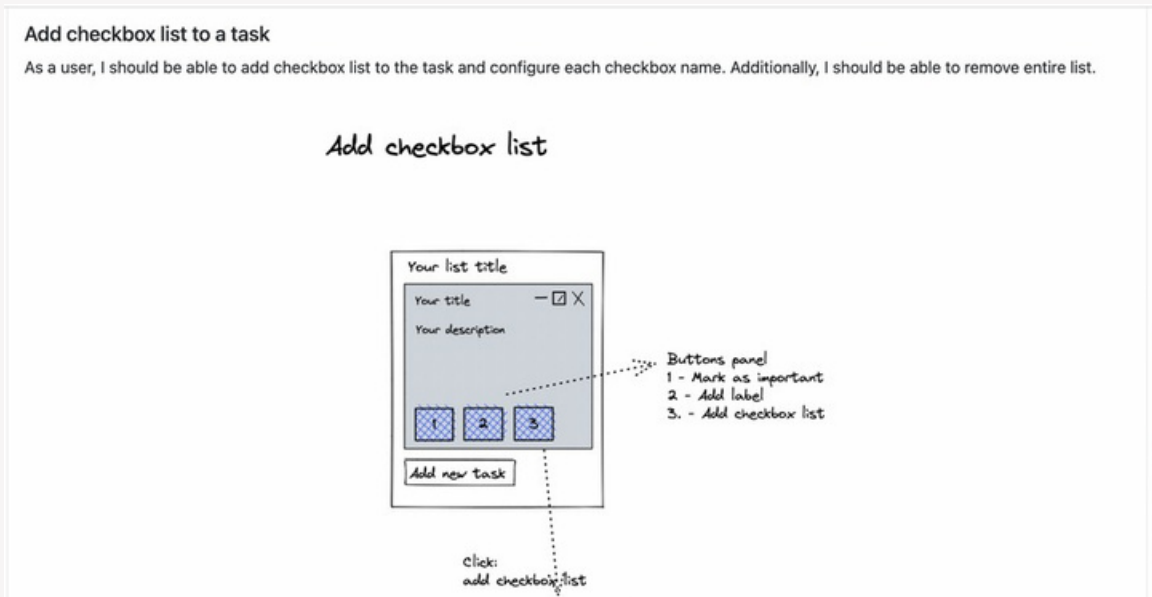
APP DESCRIPTION

The app allows adding a list to a table. Each list can be named. Each list can be filled with tasks to accomplish. In each task title, description and additional tags can be added or edited.

Full documentation: https://github.com/ABfeMate/manual_tester_portfolio/tree/main/to_do_app_documentation



TESTED FUNCTIONALITY: 'Add checkbox list to a task'

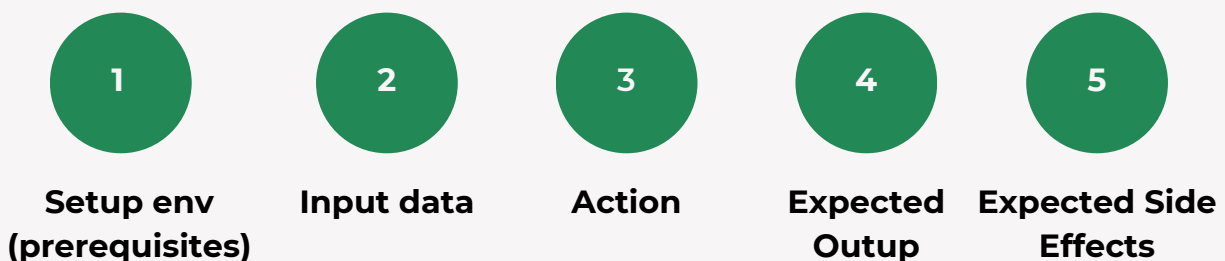


HOW I CREATE TEST CASES:

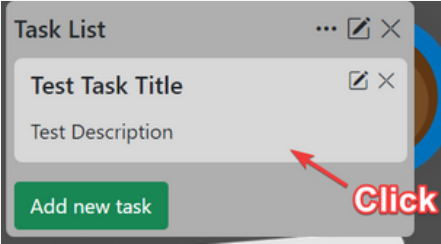
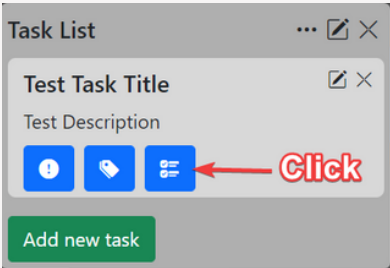
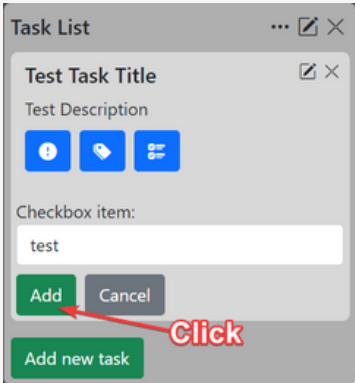
Test cases are considered in 4 categories:

- **Functional** (basic functionality according to documentation) - within functional testing, I check all possible scenarios described in the documentation, as well as those that are not obvious. I verify all possible correct combinations of input data, as well as those that may cause errors such as blank spaces, special characters, etc. [see TestCase1]
- **Usability/Visual** (based on UX/UI mockups) - within UX/UI testing, I verify whether the application is consistent with the prepared mockups, as well as whether the user interface looks aesthetically pleasing and is usable. [see TestCase2]
- **Technical** (changes in HTML code, HTTP errors from the server, etc.) - within technical testing, if possible, I try to modify the HTML code, check errors returned from the server, simulate internet network slowdowns in the browser, etc. [see TestCase3]
- **Performance** (if possible)

Usually, I prepare test cases according to the following scheme:



TEST CASE 1 - Adding "checkbox list" to a task list. **Passed**

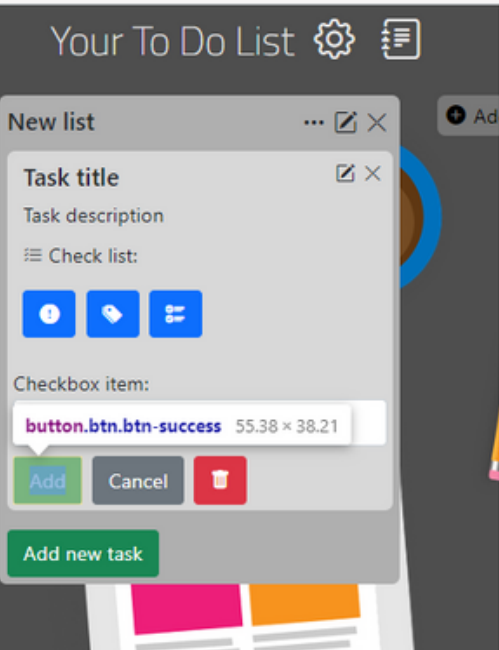
Action	Input	Output
1. Add an empty list	Title: Test List	New task list is created
2. Add a Task to a list	Title: Test, Desc.: Test	New task is added to a task list
3. Widen task panel by clicking a gray area		"Add checkbox list" button appeared
<div></div>		
4. Click "Add checkbox list"		Panel containing "Add checklist" button appeared
<div></div>		
5. Check if it is possible to add new item with an empty name field.		No possibility to add checkbox item without a name. [as described in documentation]
6. Add checkbox item name	Checkbox item name: test	Checkbox item was correctly added and the list was successfully created. Next task can be added.
<div></div>		

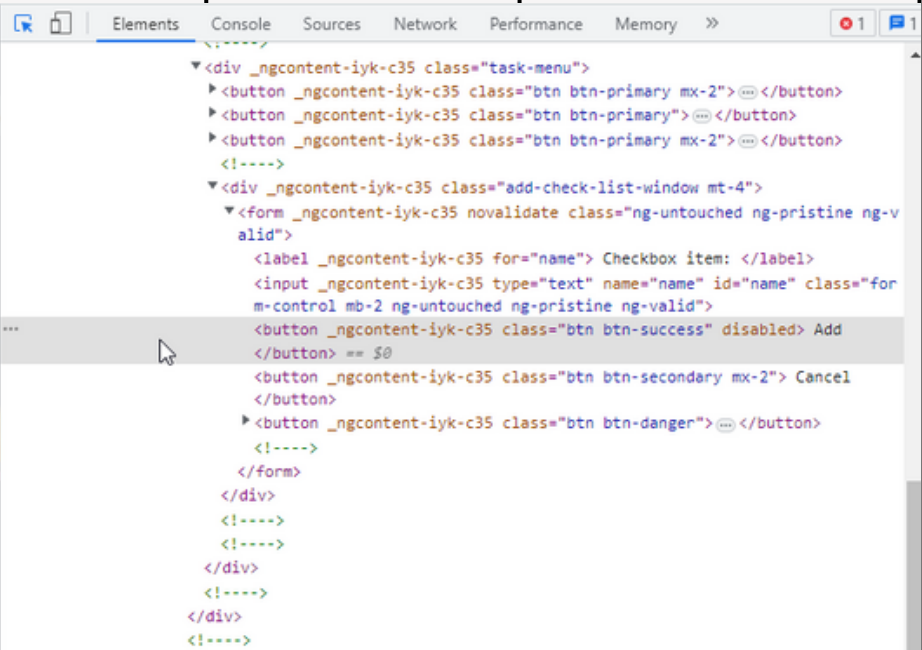
TEST CASE 2 - Creating 'Checklist item' title with a large number of characters. **BUG**

Action	Input	Output
1. Add an empty list	Title: New list	New task list is created
2. Add a Task to a list	Title: Task Title, Description: Task description	New task is added to a task list
3. Click "Add checkbox list" button		Panel containing "Add checklist" button appears
4. Add a long "checkbox item" name	Text: Lorem Ipsum Length: 2000 symbols	Currently, there is no limit when it comes to checklist item title length. (as per documentation the length should not be more than 20 symbols) Any amount of text can be entered which causes the task to stretch across the entire screen. Additionally, the checkbox is flattened, which makes it difficult to interact with it and doesn't look aesthetically pleasing.

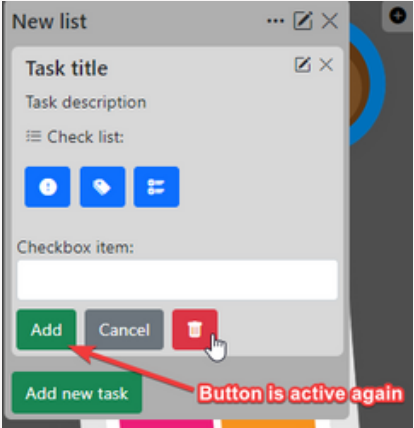
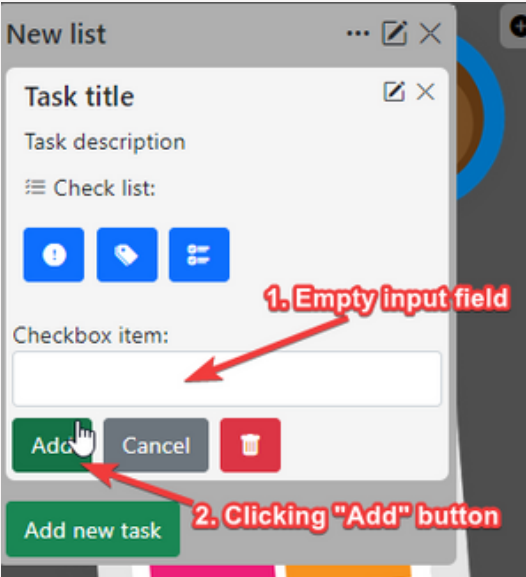
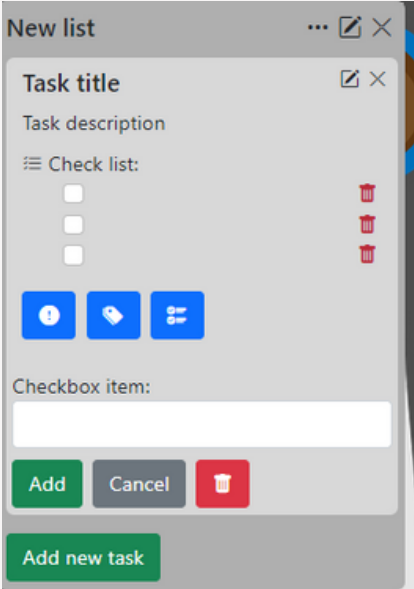
TEST CASE 3 - Adding a 'checkbox item' with an empty title. **BUG**

Action	Input	Output
1. Add an empty list	Title: New list	New task list is added
2. Add Task to a list	Title: Task Title, Description: Task description	New task is added to a task list
3. Click "Add checkbox list" button		Panel containing "Add checklist" button appears
4. Open DevTools in the browser and find the markup for the "Add" button.		





```
<div _ngcontent-iyk-c35 class="task-menu">
  <button _ngcontent-iyk-c35 class="btn btn-primary mx-2"></button>
  <button _ngcontent-iyk-c35 class="btn btn-primary"></button>
  <button _ngcontent-iyk-c35 class="btn btn-primary mx-2"></button>
</div>
<div _ngcontent-iyk-c35 class="add-check-list-window mt-4">
  <form _ngcontent-iyk-c35 novalidate class="ng-untouched ng-pristine ng-valid">
    <label _ngcontent-iyk-c35 for="name"> Checkbox item: </label>
    <input _ngcontent-iyk-c35 type="text" name="name" id="name" class="form-control mb-2 ng-untouched ng-pristine ng-valid">
    <button _ngcontent-iyk-c35 class="btn btn-success disabled"> Add
  </button> == $0
    <button _ngcontent-iyk-c35 class="btn btn-secondary mx-2"> Cancel
  </button>
    <button _ngcontent-iyk-c35 class="btn btn-danger"></button>
  </form>
</div>
</div>
<!-->
</div>
<!-->
</div>
<!-->
```

Action	Input	Output
<div>5. Remove 'disabled' attribute from HTML</div> <div><pre>ngcontent-gtt-c35 for="name"> Checkbox item: </label> ngcontent-gtt-c35 type="text" name="name" id="name" class="for 1 mb-2 ng-untouched ng-pristine ng-valid"> ngcontent-gtt-c35 class="btn btn-success" disabled> Add > == \$0 ngcontent-gtt-c35 class="btn btn-secondary mx-2"> Cancel > ngcontent-gtt-c35 class="btn btn-danger"> ... </button></pre></div>		<div></div>
<div>6. Click "Add" button without providing any input</div> <div></div>		<div><p>New checkbox items are added without a title, which is bug that can compromise data integrity of the app. The app should have client-side as well as server side validation to prevent such bugs.</p><div></div></div>