

Automate now		Automate later		Don't automate	
<div>Add a new ToDo item</div>	<div>why: they are quite simple tests which would take longer to be manually tested, we would be able to cover more cases with automation</div>	<div>A ToDo item can be marked as completed</div>	<div>why: quite a large test, simple enough for manual testing but: could be automated later</div>	<div>Modify a ToDo item (by double- clicking)</div>	<div>why: we feel it would take longer to write automation for this than it would to manually test it as it is simple and there is only one element to test. its also not high priority</div>
<div>Check status bar</div>	<div>why: simple enough test, would be quicker automated than non-automated</div>	<div>Status bar can filter ToDo items: toggle between Active, All and Completed</div>	<div>why: quick enough for a human to test easily, wouldn't save much time with automation but if resources would allow then it could be somewhat easily implemented</div>	<div>A ToDo item can be deleted</div>	<div>why: Its not a high priority to be added, again it is a simple manual test but quite a complicated automated test as it relies on a hover over function . could be added to modify test above.</div>
<div>ToDo items from one variant do not "bleed over" into other applications (e.g if you create a ToDo item in the React version, it is <i>not</i> visible in the Vue.js version)</div>	<div>why: this would save time if automated, and would potentially make it simpler. the only potential issue would be if there was a problem/change with the other variant we were checking this against</div>	<div>Clicking the down arrow symbol next to the "What needs to be done?" box will toggle all items to Completed or Not Completed</div>	<div>why: wouldn't save much time being automated as it is a simple click view the results and click again, we feel like this would be quicker as a manual test but it could be automated in the future.</div>	<div>When there are any completed items, a "Clear completed" link appears in the status bar</div>	<div>why: because it is easier to check for elements that are there than elements which are not with automation so manual might be simpler</div>
		<div>Each individual ToDo item has its own entry in local storage, for instance if todos-react says that there is a ToDo item with UUID b3a592f9-fbfb-6461-4eef- fba1274b5868 then there will be a corresponding key todos-react- b3a592f9-fbfb-6461-4eef- fba1274b5868 containing a JSON representation of that ToDo item.</div>	<div>why: we don't currently know how we would go about automating this but once we learned how, this could potentially be an easier test with automation as it is just checking an assertion</div>	<div>The master list of ToDo items is stored in the browser's local storage, with the key name "todos-<version>" (e.g. todos- react for the React version). This key contains a comma-separated list of all the UUIDs for the ToDo items.</div>	<div>why: we have put this in the don't automate because we wouldn't know how to go about automating this. it could be that as we go on we find out how to and that it would be beneficial to automate this, so could potentially be moved to automate later</div>

End-to-end testing	Integration testing	Unit testing
<div>ToDo items from one variant do not "bleed over" into other applications (e.g if you create a ToDo item in the React version, it is <i>not</i> visible in the Vue.js version)</div>	<div>Status bar can filter ToDo items: toggle between Active, All and Completed</div>	<div>Add a new ToDo item</div>
<div>Clicking the down arrow symbol next to the "What needs to be done?" box will toggle all items to Completed or Not Completed</div>	<div>Check status bar</div>	<div>A ToDo item can be deleted</div>
<div>The master list of ToDo items is stored in the browser's local storage, with the key name "todos-<version>" (e.g. todos-react for the React version). This key contains a comma-separated list of all the UUIDs for the ToDo items.</div>	<div>Modify a ToDo item (by double-clicking)</div>	<div>A ToDo item can be marked as completed</div>
<div>Each individual ToDo item has its own entry in local storage, for instance if todos-react says that there is a ToDo item with UUID b3a592f9-fbfb-6461-4eef-fba1274b5868 then there will be a corresponding key todos-react-b3a592f9-fbfb-6461-4eef-fba1274b5868 containing a JSON representation of that ToDo item.</div>		
<div>When there are any completed items, a "Clear completed" link appears in the status bar</div>		