Testing Plan

JSON test files will be held in a folder called Test Plan JSONs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Outcome** | **Success** | **Comments/Explanation** |
| 1. | Standard JSON. Has been created by programmer. It will contain data types of date, select, dropdown, entrybox, select, numberRange and email. Each field will have some sort of validation on it, please see file for more information | JSON1.json | All fields will be displayed correctly, and the correct declared entry types have been rendered. | True | All labels and entry fields were displayed correctly. Validation was also successful. |
| 2. | This is the Bart Simpson JSON. It was given to me by my lecturer. This will be untouched so no specialized data types will be rendered | JSON2.json | All labels and fields will be displayed correctly. I do not expect any specialized data types to appear | True | All labels and fields rendered correctly without any issues 😊 |
| 3 | Ill be adding a Date of Birth entry field deep inside the Bart Simpson JSON. This will ensure the recursion and the identifying works correctly | JSON3.json | All labels and fields will be displayed correctly. I expect a single date field to be inside | True | Surprisingly, it worked. I added the required validation and that worked. It had rendered the field and the error message also appeared |
| 4 | Empty JSON File | JSON4.json | Nothing will be rendered except the submit button | True | Just the submit button was rendered however should probably make a check that displays a message that the JSON is empty |
| 5 | Sending an unknown file type, a .docx file type. It however will be filled with the contents of JSON1 | JSON5.docx | I expect the Axios reader to read the document however the formatting will be wrong. So nothing will be rendered except the submit button. | False | It generated 13500 labels and fields. These labels were just numbers ascending to 13500. The submit button was still functional. |
| 6 | Sending an invalid file/path to a file that does not exist. | “axios.get(  ‘/invalid’)” | The submit button should be the only thing that is rendered | False | It generated 2222 labels and fields. The submit button still worked for all the fields |

# Conclusion

I think the program works fine. There is a bug where it generates labels and fields when there is no data present. These fields are numbered and iterate to a very high number, not sure where they came from.