**Test Table #1: Theatre Tickets**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Data | Reasons for the test | Expected Results | Actual Results | Any problems with the results and how they were fixed |
| 1. User Form | Testing the user forms was a necessary test since it is the key part of the program, where the user can input their details when for when they go to register for a theatre ticket. | The expected results from this test, was that when the user inputs their data, it stays and registers their input, or the user can auto-fill their information if they so wish to. | The results that occurred were positive, the form was responsive to the user input and the option to auto-fill the information so the form can be filled out quickly. | Since there were no problems, nothing needed to be fixed. |
| 2. Required fields | Since the form’s details are important, its vital that the user inputs the correct information without missing any important details, so to ensure this there needed to be some restriction of progression until it was done. | The expected results was that, if a user elects to ignore a field, such as a postal code or a second name and then tried to submit the form, then the user would get an alert stating that they need to fill it in. | The results gained from this test were positive and relatively easy to accomplish, the since it was one line of code to allow it to function, the user gets an alert at the top of their screen telling them to please fill in the field so that the form can be completed properly. | There were no problems with this test. |
| 3.Available Shows drop down list | The reasoning behind this test is that its required that the user picks the content that they need, if the drop-down list does not function as expected then the user is going to have no contents to choose from. | The results expected from this test is that the user can choose from 4 different show choices in the drop-down list, which then shows the total price when the button is clicked. | The results that came from this test were positive because when the user clicks onto the drop-down list, they can select four shows from that list, which once chosen displays the cost of the show when the button is clicked. | There were no issues with this test. |
| 4.Methods of delivery | Testing this was necessary due to costs of the deliveries needed to be displayed on the page with the other information. | I expected that once the choice had been selected, then the price of that delivery method would be displayed on the page along with the cost of the tickets. | At first there was some issues with the delivery costs merging with the formula of the ticket costs in the JS code. It was showing an impossible number. | It worked once I had done tweaking to the code, I changed the id’s of where the result was being displayed rather than in the same <P> tag as the previous equation. |
| 5.Show Price | The show price button is the most important aspect of the program, since this is used to display the results of what the user has requested in the previous functions. | In this test the user selects their show and the amount of tickets then it would be clicked and then it would display the costs underneath, displaying both the cost of the show, the discounts, and the delivery costs. | The button had worked brilliantly, it displayed all transactions and the information was displayed beneath it. | There were no issues with this test. |
| 6.Ticket Number drop down list | The goal of this test was to allow the user to select the amount of tickets they wanted. The user must select a number from the drop-down list, this enables the user to choose the amount of tickets they wanted for the show that they are booking. | As expected, the number clicker is meant to show the amount of tickets the user wants to order, however once its reached 6-10 tickets it must display a 10% discount, this also applies when it reaches 10-15 which then it must display a 15% discount. | The list once selected, displayed the amount of tickets that needed to be purchased, it displayed the costs correctly, however it struggled to display the discounts properly due to some incorrect ID’s. | To fix the issue, I re-wrote the same code again, however I double checked to make sure that each displayed value had its own ID so that it didn’t get confused as to display what and where. |

Activity Diagram – Theatre Tickets:

