Testing Performed

| Area Tested | How Tested | Result |
| --- | --- | --- |
| Add to Cart | Added multiple products from the catalog | Cart displayed correct items/qty |
| Remove from Cart | Clicked remove on an item in the cart | Item removed, totals updated |
| Update Cart Quantity | Changed quantities and reviewed totals | Quantity, subtotal, and totals updated accordingly |
| Invalid Inputs | Tried negative/empty/large values in forms | Errors handled or inputs rejected |
| Database Persistence | Checked cart content after page reload/log out | Data remained consistent |
| User Interface | Viewed on desktop & mobile | Layout was consistent and clear |

*Summary:*  
Every major feature, error condition, and user flow was tested. Any issues found during testing were resolved before submission to ensure the application meets the project criteria.

4. Retrospective Document

What went well:  
Breaking the project into steps made the larger work feel manageable and prevented overwhelm. Implementing the MVC pattern kept the code organized and made fixing bugs easier. Figuring out how to keep the cart functioning correctly (add/update/remove) was a successful milestone.

What went poorly:  
Switching between different forms and matching POST/GET to the correct controller action caused errors that took time to diagnose. Adjusting database structure late in the project led to some unnecessary rework and minor data loss during local testing.

Was your project plan helpful?  
The project plan provided a basic structure and clear weekly targets, which helped pace my work and made progress tangible.

How closely did you stick to the project plan?  
I followed much of the plan, but sometimes shifted tasks between weeks due to bugs or extra work. A few tasks needed to be broken down further or repeated.

How much additional work did you find?  
A fair amount of unplanned work came up, especially when integrating the cart logic and testing error conditions. Debugging unexpected behaviors and restructuring for MVC took extra effort.

What would you do differently if starting over?  
I would define database structure and MVC architecture from the beginning, and set up regular checkpoints for code review and incremental testing. Logging progress each day would also make status updates easier and more accurate.