A screenshot of a computer program

Description automatically generated

For this milestone, I implemented unit tests for ‘std::vector’ using Google Test. The goal was to verify vector behavior under different conditions, including positive and negative test cases. I started by ensuring that a newly created vector was empty and its smart pointer was properly initialized. Then I added tests to check basic operations like adding elements, resizing, clearing, and erasing data.

One of the things I focused on was making sure that max size and capacity behaved correctly as elements were added. The tests checked that ‘max\_size()’ was always greater than or equal to ‘size()’, and that ‘capacity()’ properly adjusted as needed. I also implemented tests for ‘resize()’, verifying that it could increase and decrease the collection’s size, including reducing it to zero.

For the negative tests, I made sure that calling ‘.at()’ with an out-of-bounds index threw a ‘std::out\_of\_range’ exception. Additionally, I created a custom test to check that accessing a negative index also resulted in an exception. These tests confirm that the vector correctly handles invalid access instead of causing unexpected behavior.

Throughout the process, I made sure to follow best practices like using ASSERT for critical failures and EXPECT where execution should continue. The debugging process mostly involved making sure the test logic was correct and that I was checking conditions at the right moments.