## Final Challenge: Put It All Together!

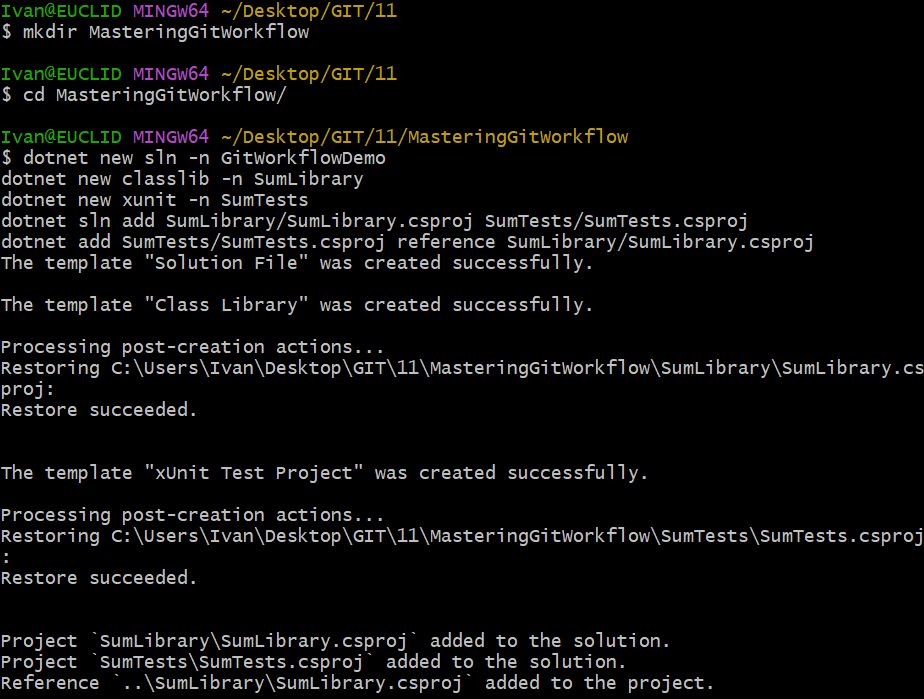
Apply everything learned in a **real-world scenario** by creating a Git workflow simulation.

### Scenario

* You are part of a development team working on a project.
* You need to:
  + Clone a repo and create a feature branch.
  + Implement a new feature and commit changes.
  + Use pre-commit and pre-push hooks to enforce standards.
  + Push changes to GitHub, triggering GitHub Actions.
  + Rebase your branch onto main, resolving any conflicts.
  + Merge the branch and clean up the old feature branches.
  + Optimize repo size using Git LFS if needed.

### Final Deliverables:

**First, we create the directory for the repository and the C# project structure:**



**Two .cs files containing the method and the test:**

SumLibrary/Sum.cs:

|  |
| --- |
| {  public class Sum  {  public static double Add(double a, double b) => a + b;  }  } |

SumTests/SumTests.cs:

|  |
| --- |
| using Xunit;  using SumLibrary;  namespace SumTests  {  public class SumTests  {  [Theory]  [InlineData(1, 2, 3)]  [InlineData(-1, 1, 0)]  [InlineData(0, 0, 0)]  [InlineData(2.5, 2.5, 5)]  [InlineData(-3, -7, -10)]  public void Add\_ReturnsExpected(double a, double b, double expected) =>  Assert.Equal(expected, Sum.Add(a, b));  }  } |

**Pre-commit and pre-push hooks:**

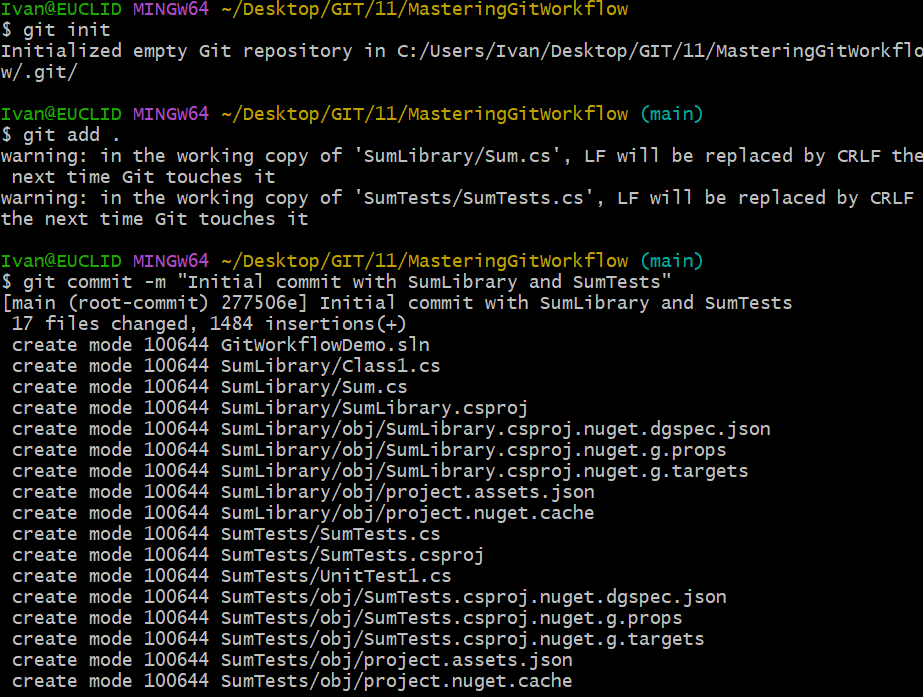
.git/hooks/pre-commit:

|  |
| --- |
| #!/bin/sh  echo "Running dotnet format"  dotnet format --verify-no-changes  if [ $? -ne 0 ]; then  echo "Commit blocked: Format issues!"  exit 1  fi  echo "Code formatting passed."  exit 0 |

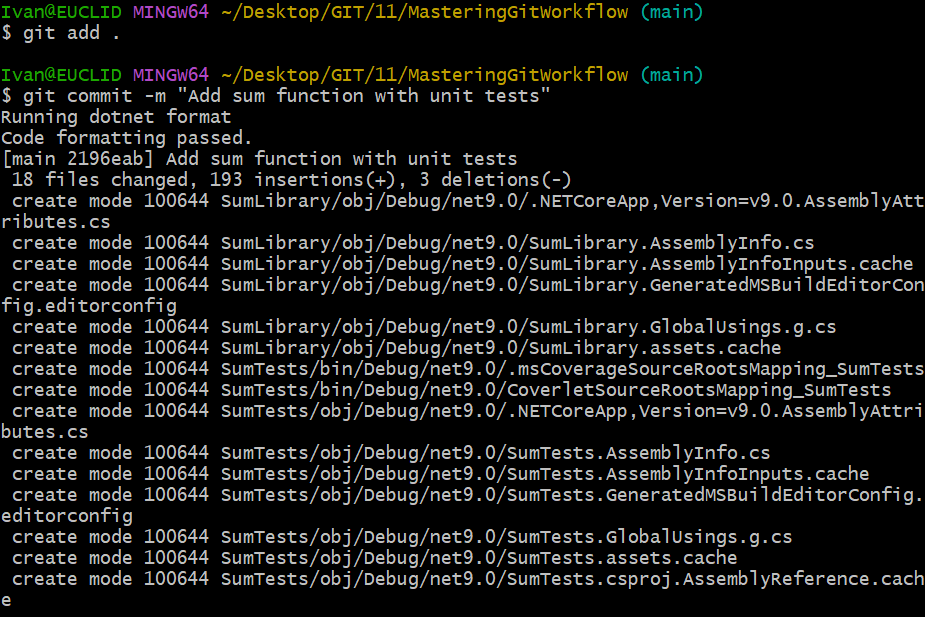
.git/hooks/pre-push:

|  |
| --- |
| #!/bin/sh  echo "Building the solution..."  dotnet build  if [ $? -ne 0 ]; then  echo "Build failed! Push blocked."  exit 1  fi  echo "Running tests..."  dotnet test --no-build  if [ $? -ne 0 ]; then  echo "Push blocked: Tests failed!"  exit 1  fi  echo "Tests passed. Push allowed."  exit 0 |

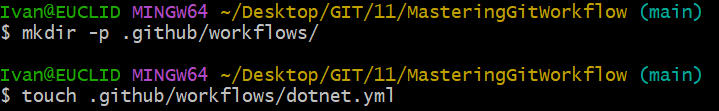
**The GIT repository was initialized:**



**The files were added and the first commit was created. The files pass the pre-commit hook check:**



**The YAML file for the workflow was created.**



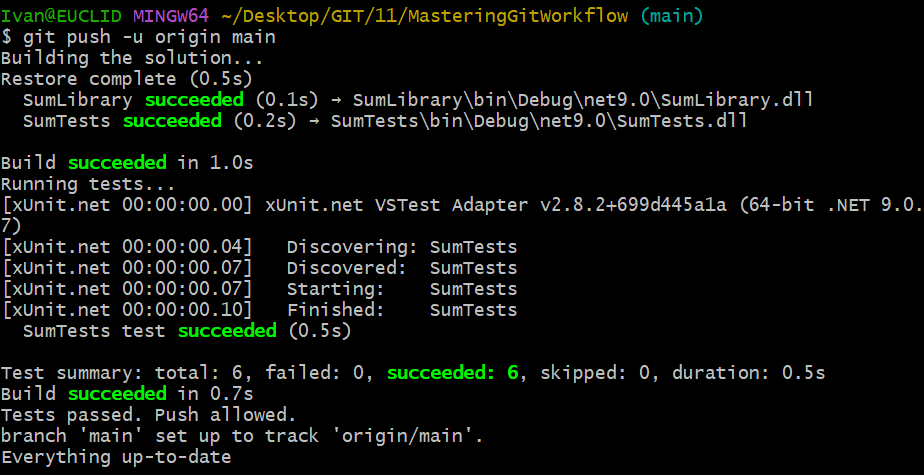
.github/workflows/dotnet.yml:

|  |
| --- |
| name: .NET CI  on:  push:  pull\_request:  jobs:  build:  runs-on: ubuntu-latest  steps:  - uses: actions/checkout@v4  - name: Setup .NET  uses: actions/setup-dotnet@v4  with:  dotnet-version: '9.0.x'  - run: dotnet restore  - run: dotnet build --no-restore  - run: dotnet test --no-build |

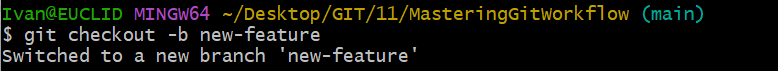
**The remote GitHub origin was added to the local directory:**



**The main branch was pushed to the remote repository, all tests passed:**



**New branch ‘new-feature’ was created:**



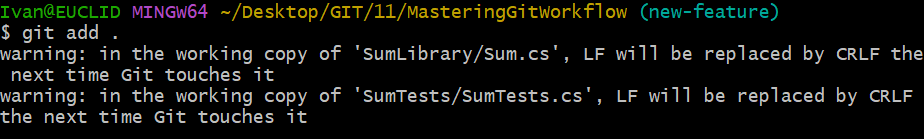
**SumLibrary/Sum.cs is modified to:**

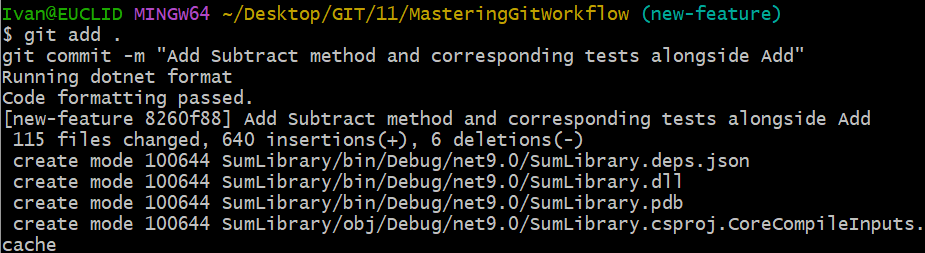
|  |
| --- |
| namespace SumLibrary  {  public class Sum  {  public static double Add(double a, double b) => a + b;  public static double Subtract(double a, double b) => a - b; // <-- NEW  }  } |

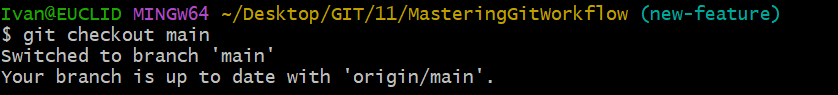
**SumTests/SumTests.cs is modified to:**

|  |
| --- |
| using Xunit;  using SumLibrary;  namespace SumTests  {  public class UnitTest1  {  [Theory]  [InlineData(1, 2, 3)]  [InlineData(-1, 1, 0)]  [InlineData(0, 0, 0)]  [InlineData(2.5, 2.5, 5)]  [InlineData(-3, -7, -10)]  public void Add\_ReturnsExpected(double a, double b, double expected) =>  Assert.Equal(expected, Sum.Add(a, b));  [Theory]  [InlineData(5, 3, 2)]  [InlineData(0, 0, 0)]  [InlineData(2.5, 1.2, 1.3)]  public void Subtract\_ReturnsExpected(double a, double b, double expected) =>  Assert.Equal(expected, Sum.Subtract(a, b));  }  } |

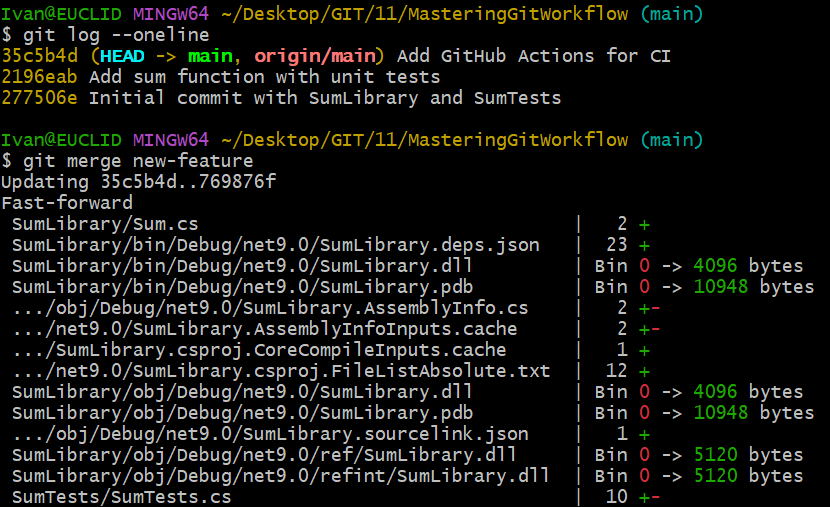
**The modification was added and new commit was performed in the ‘new-feature’ branch:**



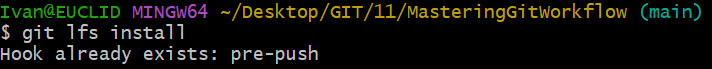


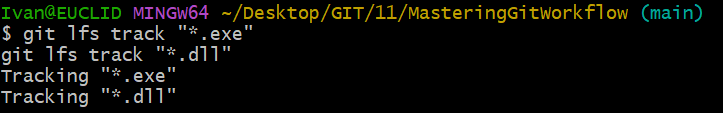


**The ‘new-feature’ branch was merged to the main branch:**

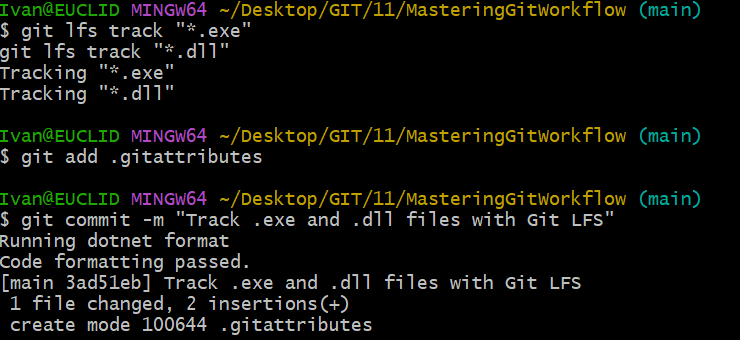


**Git LFS was installed and set to track .exe and .dll files:**

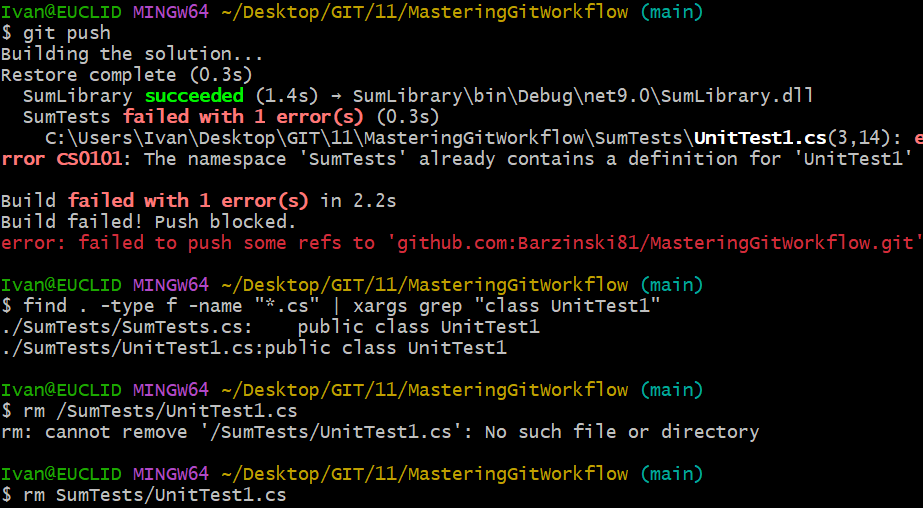




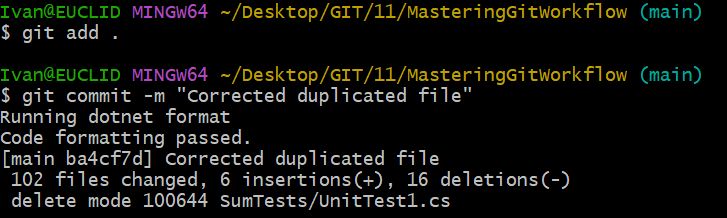
**The change was committed:**



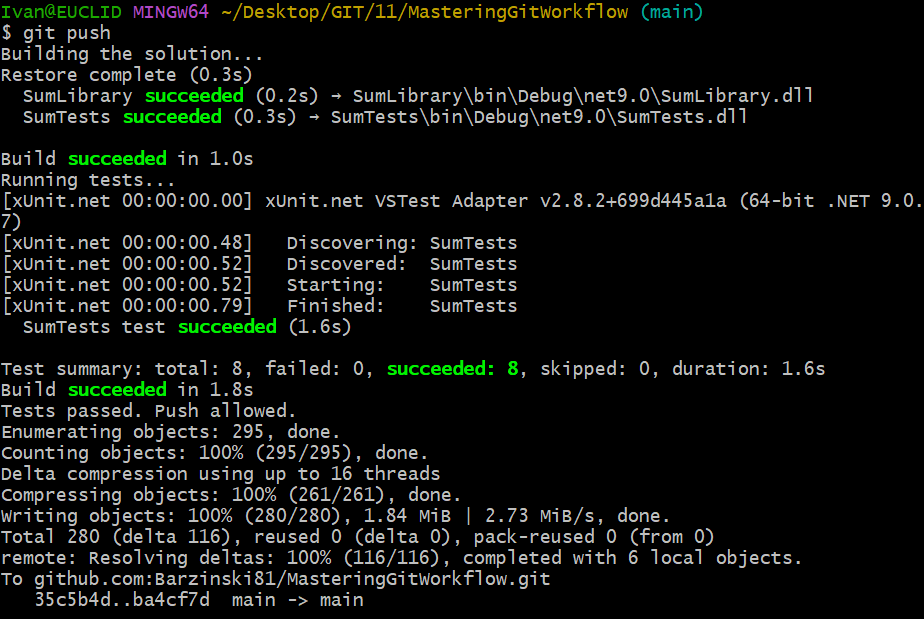
**There was duplicated namespace, so I deleted the file preventing the push:**



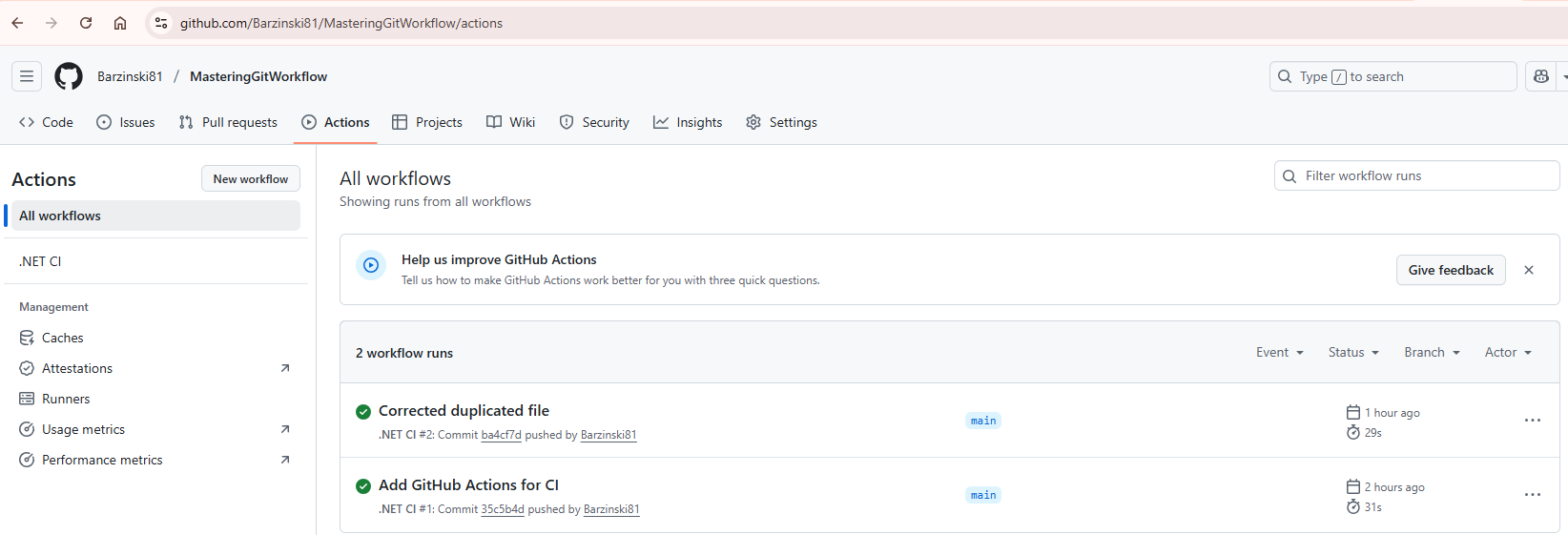
**The changes were added and a new commit was made:**



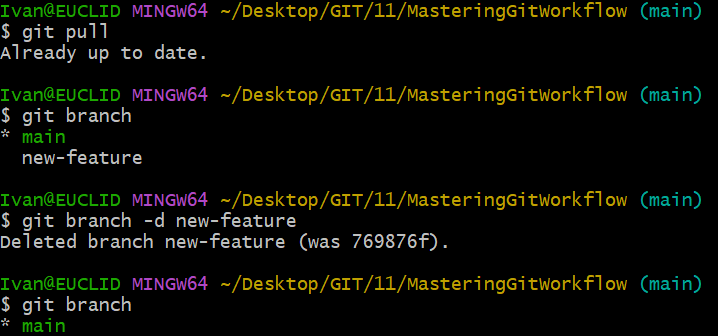
**The changes were pushed to GitHub:**



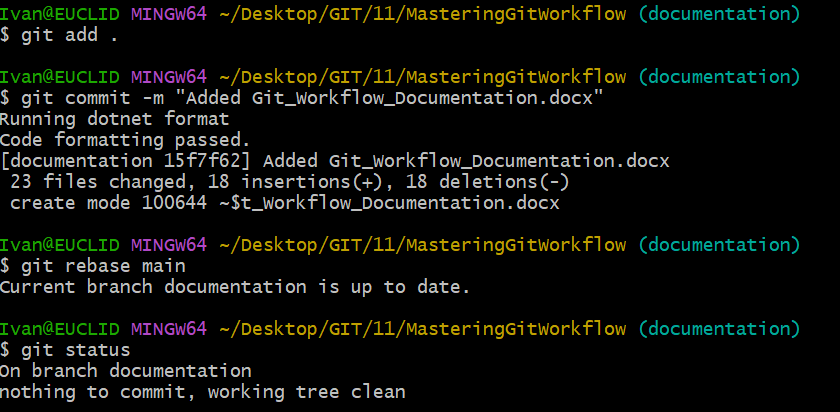
**Both workflows (the original one and the modified from the ‘new-feature’ branch) were executed successfully:**

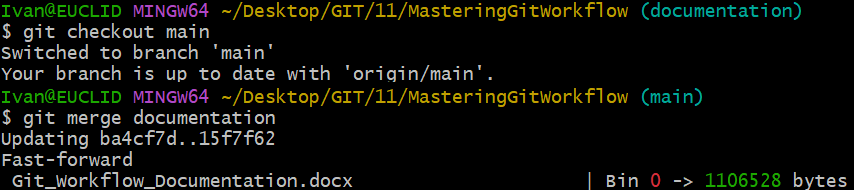
****

**The ‘new-feature’ branch was deleted as it is already merged to the ‘main’ branch:**



**New ‘documentation’ branch was created and the current document was added to it. After that the branch was rebased:**





**The current file was updated with the latest changes:**

