Live Demo for Lab session

```
==========================================

# 1. Test locally first (show students this works)

python -m unittest discover tests -v


# Expected output:

# test_add_negative_numbers (test_calculator.TestCalculator) ... ok

# test_add_positive_numbers (test_calculator.TestCalculator) ... ok

# test_add_zero (test_calculator.TestCalculator) ... ok

# ... (more tests)

# Ran 10 tests in 0.001s

# OK


# 2. Initialize git repository

git init


# 3. Add all files

git add .


# 4. Check what we're committing

git status


# 5. Make first commit

git commit -m "Initial commit: Add Python calculator with tests and CI
workflow"


# 6. Create repository on GitHub (show in browser)
```

```
# - Go to github.com

# - Click "New Repository"

# - Name it "github-actions-demo"

# - Don't initialize with README (we already have one)

# - Click "Create Repository"


# 7. Connect local repo to GitHub

git remote add origin
https://github.com/YOUR_USERNAME/github-actions-demo.git

git branch -M main

git push -u origin main


# 8. Watch the magic happen!

# - Go to GitHub repository

# - Click "Actions" tab

# - See your workflow running (should show green checkmark)

# - Click on the workflow run to see detailed logs


# 9. Let's break something to see CI in action!

# Edit tests/test_calculator.py and change line 23:

# self.assertEqual(result, 5)  # Change to 6


# 10. Commit the broken test

git add tests/test_calculator.py

git commit -m "Break a test to demonstrate CI failure"

git push


# 11. Watch the workflow fail
```

```
# - Go back to Actions tab

# - See the red X indicating failure

# - Click on the failed workflow to see error details


# 12. Fix the test and push again

# Change it back to 5

git add tests/test_calculator.py

git commit -m "Fix the broken test"

git push


# 13. Watch it pass again!
```

Key Points During Demo:

1. **File Structure Matters:** Show how the `__init__.py` files make directories into Python packages
2. **Local Testing First:** Always test locally before pushing (`python -m unittest discover tests -v`)
3. **Immediate Feedback:** GitHub Actions runs within 30 seconds of pushing
4. **Visual Feedback:** Green checkmarks vs red X's in the Actions tab
5. **Detailed Logs:** Students can see exactly what happened by clicking on workflow runs

Questions for Students:

1. What happens if we don't have the `__init__.py` files?
2. Why do we test locally first before pushing?
3. What would happen if we had a syntax error in our Python code?