**EXERCISE I**

Create a CI workflow that automatically runs tests on every code change. Your project will use **Mocha** for testing and **Chai** for assertions. You'll also learn how to resolve a merge conflict.

**Step 1: Project Setup and Code**

First, set up your Node.js project, including the correct logic and test files.

1. **Initialize Project & Install Dependencies:** Open your terminal and run the following commands to create your project and install Mocha and Chai:

Bash

mkdir node-ci-exercise

cd node-ci-exercise

npm init -y

npm install mocha chai --save-dev

1. **Create Source Code (index.js):** Create a file named index.js with an add() function that checks for number types. Use ES Module syntax with export default.

JavaScript in Exercise Folder

// index.js

1. **Create Test File (test/index.test.js):** Create a test folder and a file named index.test.js inside it. Write your tests using Mocha and Chai syntax with import statements. .

JavaScript in Exercise Folder

// test/index.test.js

1. **Configure package.json:** Update your package.json file to use ES Modules and configure the test script for Mocha.

JSON in Exercise Folder

Run npm test locally to confirm everything works before moving on.

**Step 2: GitHub Actions Workflow and Git**

Now, create your GitHub Actions workflow and push your code to a new repository.

1. **Create GitHub Repository:** Go to GitHub and create a new, empty repository.
2. **Create CI Workflow File:** In your project, create the folder .github/workflows and add a ci.yml file inside it. This workflow will check out your code, set up Node.js, and run your tests.

YAML in Exercise Folder

# .github/workflows/main.yml

1. **Push Code to GitHub:** Initialize Git, commit your files, and push them to your new repository.

Bash

git init

git add .

git commit -m "initial commit"

git branch -M main

git remote add origin https://github.com/YOUR\_USERNAME/YOUR\_REPO\_NAME.git

git push -u origin main

**Step 3: Simulate and Resolve a Merge Conflict**

Learn how to handle a common Git issue that you'll see in CI workflows.

1. **Trigger the Conflict:** Go to your GitHub repository and directly add a README.md file. Commit the changes.
2. **Attempt to Push:** Go back to your terminal and try to git push origin main again. Git will reject the push with a non-fast-forward error.
3. **Resolve the Conflict:** Pull the latest changes from GitHub with git pull origin main. Git will automatically merge the new README.md file into your local branch.
4. **Push Again:** Now that your local branch is synchronized, you can successfully push your changes.

Bash

git pull origin main --allow-unrelated-histories

Esc => :wq

git add .

git commit -m "Resolve merge conflicts"

git push origin main

After the push, your GitHub Actions workflow will run automatically. You can go to the **"Actions"** tab on your repository to see that all the steps pass successfully. .