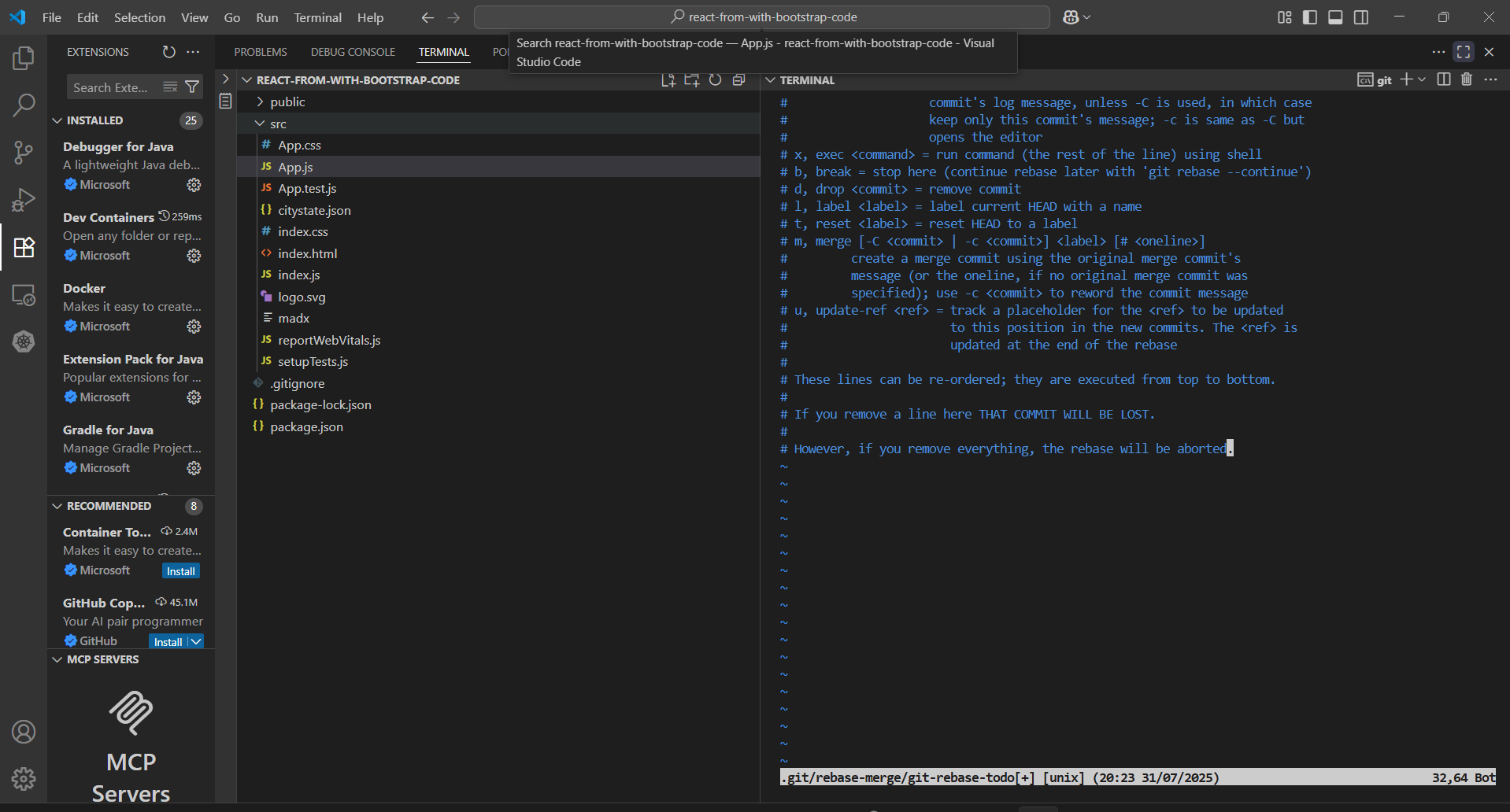
# Day 4– #60DaysOfDevOps Challenge

**Advanced Git Challenges Documentation**

**✅ Challenge 1: Interactive Rebase to Modify Commit History**

**Goal:** Use git rebase -i to reword, squash, and reorder commits.

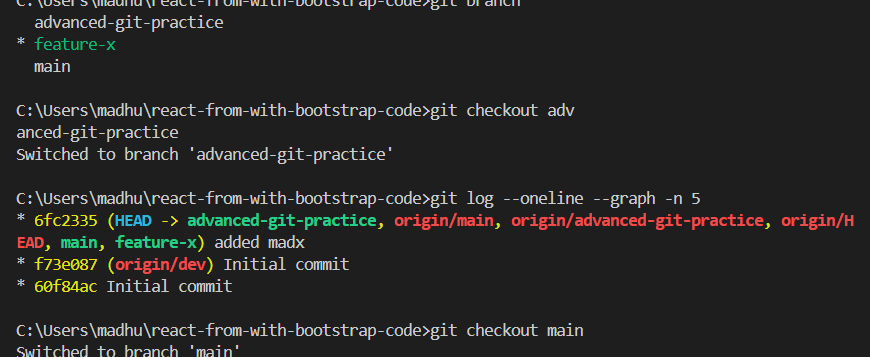
**Steps:**

1. View recent commits:
2. git log --oneline -n 5
3. Start interactive rebase (for last 3 commits):
4. git rebase -i HEAD~3
5. Use the editor to:
   * reword a commit message (change pick to reword)
   * squash commits together (change pick to squash on lower commits)
   * reorder lines to change commit order
6. Save and exit (:wq in Vim)
7. If prompted to edit commit message, change or save as needed.
8. Push updated history:
9. git push origin branch-name –force
10. 

**✅ Challenge 2: Use git cherry-pick**

**Goal:** Apply a commit from one branch onto another.

**Steps:**

1. Checkout feature branch:
2. git checkout feature-branch
3. git log --oneline -n 5
4. Copy commit hash you want to cherry-pick (e.g., abc1234).
5. Switch to target branch:
6. git checkout main
7. Apply commit:
8. git cherry-pick abc1234
9. If conflict occurs:
10. git add <resolved-file>
11. git cherry-pick --continue
12. Push changes:
13. git push origin main
14. 

**✅ Challenge 3: Squash Commits into One (In Progress)**

**Goal:** Combine multiple commits into a single, clean commit.

**Steps:**

1. Create feature branch:
2. git checkout -b feature-branch
3. Make several commits:
4. echo "Line 1" > file.txt
5. git add file.txt
6. git commit -m "Commit 1"
7. echo "Line 2" >> file.txt
8. git commit -am "Commit 2"
9. echo "Line 3" >> file.txt
10. git commit -am "Commit 3"
11. Start interactive rebase:
12. git rebase -i HEAD~3
13. In editor:
    * Keep pick on first line
    * Change rest to squash
    * Save and write final commit message

