Git Guidelines

1. Write Clear Commit Messages

- **Summary:** Start with a brief summary of what the commit does (e.g., “Fix login bug”).
- **Details:** Optionally, provide more details in the body if needed (e.g., “Corrected issue where login would fail on invalid input”).

2. Commit Frequently

- Make commits that represent logical chunks of work. Small, frequent commits are easier to understand and review.

3. Use Branches Wisely

- **Feature Branches:** Create branches for new features or fixes (e.g., feature/new-payment-method).
- **Naming:** Use clear names to describe what the branch is for.

4. Merge with Care

- Regularly pull updates from the main branch into your feature branches to avoid conflicts.
- Resolve any conflicts carefully to ensure nothing breaks.

5. Use Rebasing Appropriately

- Rebasing can help keep your history clean but use it cautiously, especially on shared branches.

6. Keep Commits Manageable

- Avoid committing large changes all at once. Instead, break them into smaller, more manageable pieces.

7. Test Before Committing

- Ensure your code runs correctly and passes tests before you commit your changes.

8. Maintain a Clean History

- Avoid altering commits that are already shared with others to prevent confusion.