



Team Member Git Collaboration Cheatsheet

Use this guide to work cleanly, avoid merge conflicts, and collaborate efficiently in a team React project.



1. General Rules

- Only **edit your assigned components/files**.
 - Do **not change someone else's code** without clear agreement.
 - Use clear branch names like:
 - feature-navbar
 - gamecard-ryan
 - favoritelist-cindy
-



2. Placeholder Files

- If a file like `Navbar.jsx` is owned by someone else, **do not write code** inside it.
- You may create an **empty placeholder file** so your app runs, but do **not commit over their work**.
- Example:

```
// Navbar.jsx (placeholder only)
const Navbar = () => null;
export default Navbar;
```

Then remove or replace this when the real version is ready.



3. App.jsx Guidelines

- You can safely add your own route like this:

```
import FavoriteList from '../pages/FavoriteList';

<Route path="/favorites" element={<FavoriteList />} />
```

- If you're not ready, **comment out your import/route**:

```
// import FavoriteList from '../pages/FavoriteList';
```

```
// <Route path="/favorites" element={<FavoriteList />} />
```

- Don't remove or edit others' imports or routes.
-



4. Respecting Component Ownership

- Don't overwrite teammates' code.
 - Only the **assigned person** should make commits to that component/file.
 - If you need their component to test yours, use a placeholder or ask them to push their latest code.
-



5. Cleaning Up Before Merge

Before pushing:

- Remove test/placeholder data.
 - Remove all `console.log`s and unused code.
 - Comment or remove any temporary code from `App.jsx`.
 - Ensure your code runs smoothly on its own.
-



6. Making a Pull Request

1. **Pull latest main** into your branch:

```
git checkout main
git pull origin main
git checkout your-branch
git merge main
```

2. Resolve conflicts in **your code only**.
3. Push your branch:

```
git push origin your-branch-name
```

4. Create a pull request (PR) with:
 - Clear title (e.g., `Add FavoriteList Page`)
 - Short description
 - No unrelated file changes
-



7. Final Push Rules

- Uncomment your route and import in `App.jsx`.
 - Make sure:
 - Your component works correctly.
 - All team code is untouched.
 - File structure is clean and readable.
-



8. Git Tips & Habits

- **Always pull before pushing.**
- Use descriptive commit messages:

```
git commit -m "Implement login form UI"
```

- Don't commit generated or `.env` files.
 - Communicate early if you hit blockers.
-

Bonus (Scrum Master Notes)

- As Scrum Master, you **may uncomment routes** or integrate components once the author has confirmed their part is done.
- You manage merges, test builds, and help solve conflicts — but everyone owns code quality.