

Git Save Points Student Exercise

Intro to Vibe Coding / Name _____ Date _____

Today you will preserve your customized Project Studio with Git and push a checkpoint to GitHub. Keep this sheet open while you work. Ask AI to explain the real files and commands it uses.

Digital lesson and copy buttons: <https://bachkidscode.app/lessons/git-save-points/>

1 Watch the Flight Deck demonstration

After a commit but before a push, where is that checkpoint?

If you edit a file after staging it, which version will the next commit record?

2 Open your real Project Studio folder

Use VS Code and your local coding agent. In ChatGPT, use the local Codex workspace that can read your files. Keep last week's project and saved data. Copy one prompt at a time from the digital lesson.

PROMPT 1 FIND AND INSPECT MY PROJECT

COPY THE TEXT BELOW

Work in my existing local Project Studio folder. Do not edit files yet. Show the actual project path, read package.json, and name two real source files. Check Git and gh. Inspect the Git root, status, recent commits, remotes, and ignore rules if present. Do not expose credentials. Stop if this folder is inside an unrelated repository. Explain what is already set up and what is missing. If local file access or a command is blocked, show the exact problem and stop for Drew. Do not change sandbox settings or switch to a cloud session.

END OF PROMPT

My project folder: _____

Two source files I can open: _____

PROMPT 2 RECORD A LOCAL CHECKPOINT

COPY THE TEXT BELOW

Preserve this Project Studio and its data. Reuse its Git repository, or initialize one here only if no repository exists. Keep dependencies, .wrangler, .build-check, .env files, and .dev.vars files out of Git. If secrets are already tracked, stop for Drew. If author settings are missing, ask for my chosen author display name and GitHub-provided no-reply email, and set them only for this repository. Show the source files and staged summary for my review before committing. After I approve, record a clearly named checkpoint, then show its short hash, message, and status. Do not push, reset, or discard work.

END OF PROMPT

My local checkpoint message: _____

3 Push the checkpoint to your GitHub repository

Open GitHub in your browser and check that you are signed in to your own account. GitHub login is separate from ChatGPT or Claude login. If an account or permission blocks you, flag Drew and keep working on local checkpoints.

PROMPT 3 SEND MY CHECKPOINT TO GITHUB

COPY THE TEXT BELOW

Help push this Project Studio to my GitHub account. Check authentication without showing tokens. If login is needed, guide me through gh browser sign-in; I enter credentials only in the browser. Inspect any existing remote and confirm its owner and repository with me. Reuse the intended remote without overwriting it. If none exists, ask for a repository name and create a private repository in my account. Push the current branch and set its upstream if needed. Never force push or rewrite history. If the remote has unexpected work, stop for Drew. Show the repository URL and verify that the latest remote commit matches my local hash and message.

END OF PROMPT

Open the repository URL in your browser. Find the latest commit and compare its message and short hash with the local result. Private repositories need an invitation before Drew can review them. Ask him which username to invite.

4 Make one small change and repeat

Choose one visible improvement: rename a heading, adjust a readable accent color, or improve an empty-state message. Predict the result, make the edit, save, and test. Then review the diff, stage the intended file, commit with a useful message, and push again. Keep everything else working.

My change and how I tested it: _____

My evidence for Drew

☐ I can find the local folder and open a source file. ☐ I can explain stage, commit, and push.

☐ The checkpoint appears on GitHub. ☐ Drew can access the repository, or I recorded the blocker.

Repository URL: _____

Latest message and short hash: _____

If blocked, my exact error or next step: _____

Stretch choices

Make it useful: add README instructions explaining the app, install, and restart steps. Test the instructions, then commit and push.

Investigate staging: use Flight Deck to stage a theme, edit it again, and inspect the committed version. Explain the result in your own words.

Code challenge: make a small improvement by hand or add one focused test. Explain the diff and push a second or third named checkpoint.

Before you leave: show Drew your repository or blocker. Keep your local project folder; Git does not automatically back up ignored local D1 records.