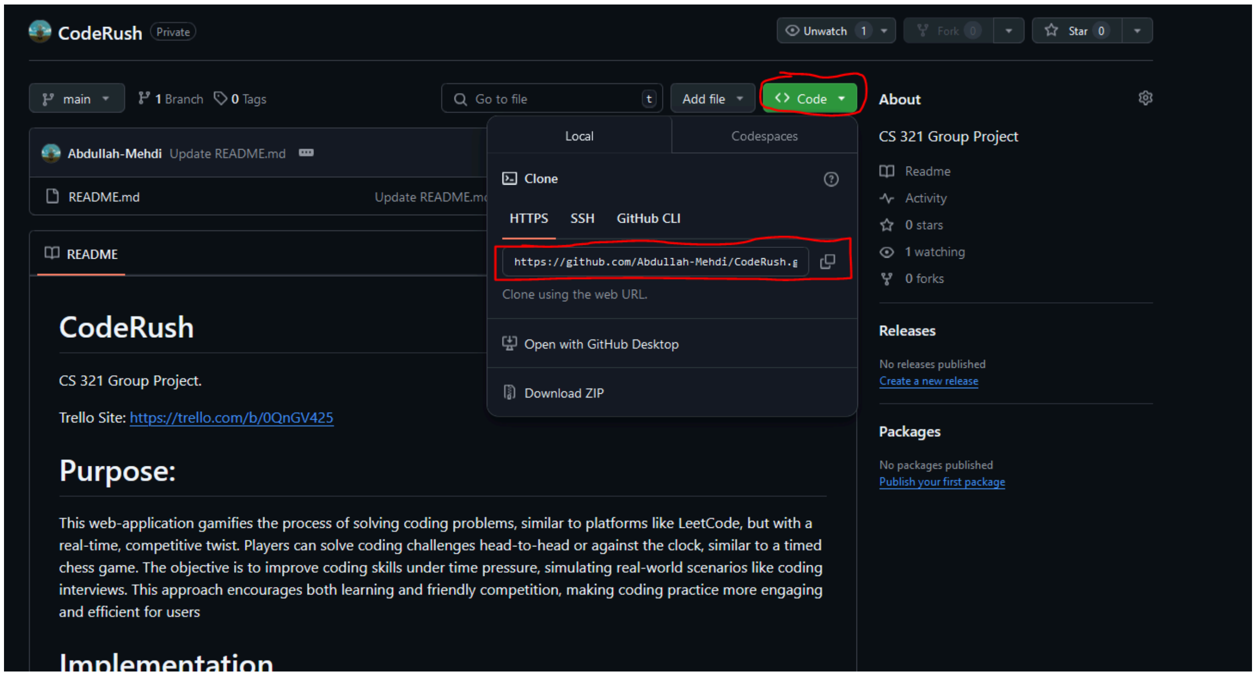
GETTING THE REPOSITORY

Cloning a Repository into your Machine

1.) Go to the GitHub page you want to clone from (https://github.com/Abdullah-Mehdi/CodeRush)

2.) Hit the green "Code" button and copy the "HTTPS" URL:



CodeRush

CS 321 Group Project.

Trello Site: https://trello.com/b/0QnGV425

Purpose:

This web-application gamifies the process of solving coding problems, similar to platforms like LeetCode, but with a real-time, competitive twist. Players can solve coding challenges head-to-head or against the clock, similar to a timed chess game. The objective is to improve coding skills under time pressure, simulating real-world scenarios like coding interviews. This approach encourages both learning and friendly competition, making coding practice more engaging and efficient for users

Implementation

3.) In your terminal/cmd/powershell change directories to where you want this project to be saved and worked from

- That folder will be linked to the GitHub repository, make sure it's **JUST** the GitHub project files that will be in there and that there are no child folders in there

4.) Run the following command `git clone [URL]`

5.) Done! The repository and its files are now downloaded on your machine. Below this page are the commands you will be using more or less

COMMANDS

Before running commands: in your terminal/powershell/cmd cd into the folder where the Git project is cloned into on your machine
- This will keep the commands simple (otherwise the commands become a bit more involved)

Updating Your Local Repository

git pull

- Get the latest changes from the remote repository and merge them into your current branch (aka the project code on your machine)
- **RUN THIS BEFORE CHANGING CODE EVERY TIME,** if you don't and you do make changes and then try to push the changes to the remote repository it will cause issues!
- Example: `git pull`

git fetch

- Downloads changes from the remote repository but doesn't merge them.
- Useful to check for updates without affecting your working directory.
- I recommend doing this before pull, good habit for working later on.
- Example: `git fetch`

Sharing Your Changes

git push

- Uploads your committed changes to the remote repository.
- Use after you've made and committed changes locally that you want to share.
- Example: `git push`

Working with Branches

Note: Each feature usually gets its' own branch as a rule of thumb

git branch

- Lists all local branches.
- Example: `git branch`

git checkout -b [branch-name]

- Creates a new branch and switches to it.
- Used when starting work on a new feature or bug fix.
- Example: `git checkout -b feature-login` (This will create a branch called feature-login on the GitHub repository and you'll be working on that branch in your local machine)

git checkout [branch-name]

- Switches to an existing branch.
- Example: `git checkout main` (Switches you to the main branch)

git merge [branch-name]

- Merges changes from the specified branch into your current branch.
- Use when you've completed work on a feature branch and want to integrate it into the main branch.
- Example: `git merge feature-login` (Merges the feature-login branch to whatever branch you're in such as "main" for example)

EXAMPLE FLOWS

—---

What a Typical Flow Looks Like

Scenario 1: The Basic Scenario

1.) `git fetch`
2.) `git pull` (If there are changes on the remote repository that aren’t on your machine)
3.) Code whatever you’re working on
4.) `git fetch` (Just to make sure no changes were done by someone else while you were working)
5.) `git pull` (If there are changes on the remote repository that aren’t on your machine)
6.) `git push`

Scenario 2: Working on a (new/existing) feature:

1.) `git fetch`
2.) `git pull` (If there are changes on the remote repository that aren’t on your machine)
3.) `git checkout -b feature-login` (Whatever name you want the branch to be)
OR if the branch already exists (you can check with `git branch`) you do `git checkout [branch-name]`
4.) Code whatever you’re working on
5.) `git fetch` (Just to make sure no changes were done by someone else while you were working)
6.)`git pull` (If there are changes on the remote repository that aren’t on your machine)
7.) `git push`

If you are done with the branch (feature) you can just do this at the end:

- `git merge feature-login`