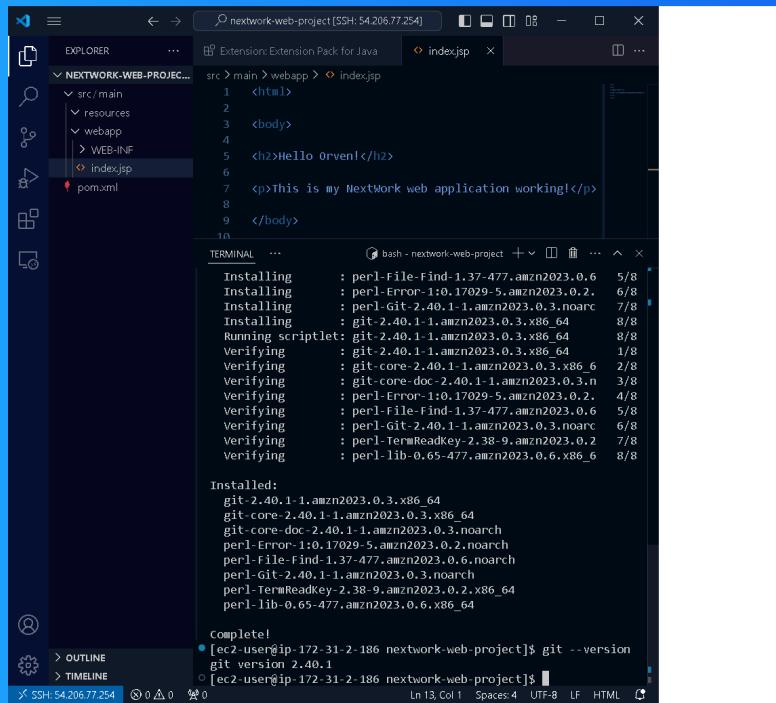




# Connect a GitHub Repo with AWS

 orven



```
Installing perl-File-Find-1.37-477.amzn2023.0.6      5/8
Installing perl-Error-110.17029-5.amzn2023.0.2       6/8
Installing perl-git-2.40.1-1.amzn2023.0.3.noarch    7/8
Installing git-2.40.1-1.amzn2023.0.3.x86_6        8/8
Running scriptlet: git-2.40.1-1.amzn2023.0.3.x86_6 8/8
Verifying git-2.40.1-1.amzn2023.0.3.x86_6          1/8
Verifying git-core-2.40.1-1.amzn2023.0.3.x86_6       2/8
Verifying git-core-doc-2.40.1-1.amzn2023.0.3.noarch  3/8
Verifying perl-Error-110.17029-5.amzn2023.0.2.noarch 4/8
Verifying perl-file-find-1.37-477.amzn2023.0.6.noarch 5/8
Verifying perl-git-2.40.1-1.amzn2023.0.3.noarch    6/8
Verifying perl-termreadkey-2.38-9.amzn2023.0.2.x86_6 7/8
Verifying perl-lib-0.65-477.amzn2023.0.6.x86_6       8/8

Installed:
git-2.40.1-1.amzn2023.0.3.x86_6
git-core-2.40.1-1.amzn2023.0.3.x86_6
git-core-doc-2.40.1-1.amzn2023.0.3.noarch
perl-Error-110.17029-5.amzn2023.0.2.noarch
perl-file-find-1.37-477.amzn2023.0.6.noarch
perl-git-2.40.1-1.amzn2023.0.3.noarch
perl-termreadkey-2.38-9.amzn2023.0.2.x86_6
perl-lib-0.65-477.amzn2023.0.6.x86_6

Completed!
[ec2-user@ip-172-31-2-186 nextwork-web-project]$ git --version
git version 2.40.1
[ec2-user@ip-172-31-2-186 nextwork-web-project]$
```



# Introducing Today's Project!

## What is GitHub?

GitHub is a web-based platform for version control that allows developers to host and manage code repositories with Git. In today's project, I used GitHub to store my code, track changes, and push updates to my index.jsp file.

## One thing I didn't expect...

The complexity of setting up authentication with GitHub. I anticipated a straightforward process, but navigating the use of personal access tokens and ensuring proper permissions added an extra layer of complexity.

## This project took me...

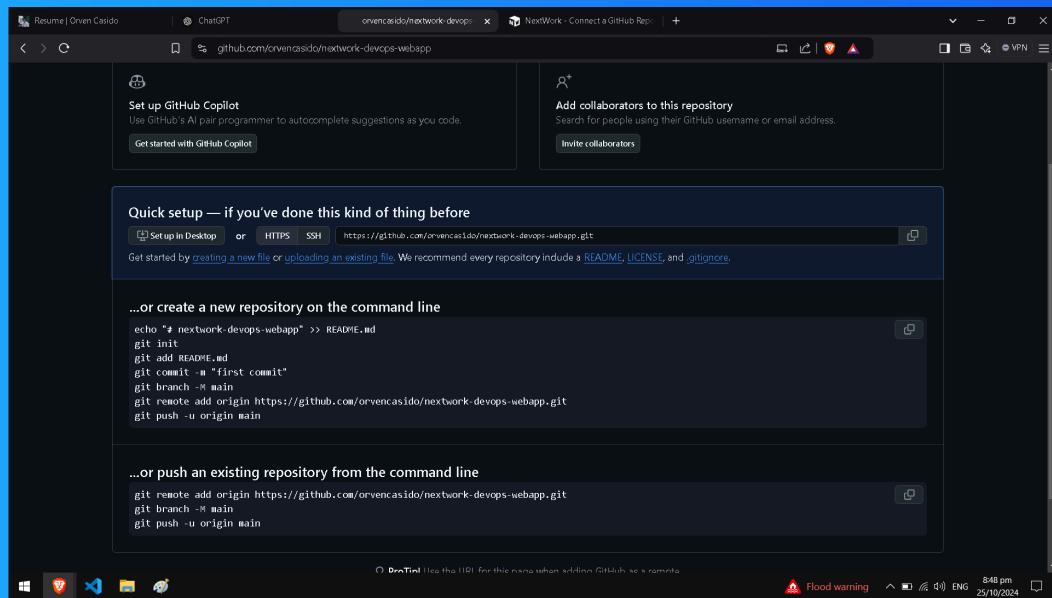
It took me less than 30 minutes to finish the project because, after completing the first one, I can comfortably navigate the EC2 and VSCode environments, especially with my GitHub setup.



# Git and GitHub

Git is a version control system that tracks changes in code, allowing multiple developers to collaborate. I installed Git using the command "sudo dnf install git -y". And verify it using the command "git --version".

GitHub is an online platform for hosting and sharing code, built on top of Git, which enables version control and collaboration. I'm using GitHub in this project to keep track of code changes and work with others.





# My local repository

A Git repository is a storage space where your project's files, along with their revision history, are stored and managed. It tracks all changes made to the files, allowing you to revert and branch.

git init is a command that initializes a new Git repository in a directory, creating the necessary files to track changes. I ran git init in my project folder to start version control, allowing Git to monitor my files.

A branch in Git is like a parallel universe for the project. Imagine each branch as a different timeline where changes to your code can be developed, tested, or explored without affecting the main project (often called the "main" or "master" branch).

```
git init
Initialized empty Git repository in /home/ec2-user/nextwork-web-project/.git/
hint: Using 'master' as the name for the initial branch. This de
fault branch name
hint: is subject to change. To configure the initial branch name
to 'main' in all
hint: your new repositories, which will suppress this warning
, call:
hint:
hint: git config --global init.defaultBranch main
hint: Names commonly chosen instead of 'master' are 'main', 'true
nk' and
hint: 'development'. The just-created branch can be renamed via
this command:
hint:
hint: git branch -m <name>
hint: Initialized empty Git repository in /home/ec2-user/nextwork-web-
project/.git/
[ec2-user@ip-172-31-2-186 nextwork-web-project]$
```



# To push local changes to GitHub, I ran three commands

## **git add**

The first command I ran was " git add . ". A staging area is a staging area is like a holding zone in Git where you prepare changes before committing them.

## **git commit**

The second command I ran was "git commit -m "Updated index.jsp with new content". Using -m means that I'm providing a message directly in the command to describe the changes made in this commit.

## **git push**

The third command I ran was "git push -u origin master". Using '-u' means that I'm setting the upstream tracking relationship between my local master branch and the master branch on the remote repository.

A circular portrait of a young man with dark hair, wearing a dark suit jacket, a white shirt, and a dark tie. He is looking directly at the camera with a neutral expression.

# orven

## NextWork Student

[NextWork.org](http://NextWork.org)

# Authentication

When I commit changes to GitHub, Git asks for my credentials because it needs to authenticate my identity to ensure that I have permission to push changes to the repository. This process protects the repository from unauthorized access.

# Local Git identity

Git needs my name and email because these details are used to identify the author of each commit. When I make changes and commit them, Git records my name and email in the commit history.

Running `git log` showed me that it lists the commit history of the repository, displaying each commit's unique hash, author name, email, date, and commit message. This information provides a chronological overview of all changes made to the project.

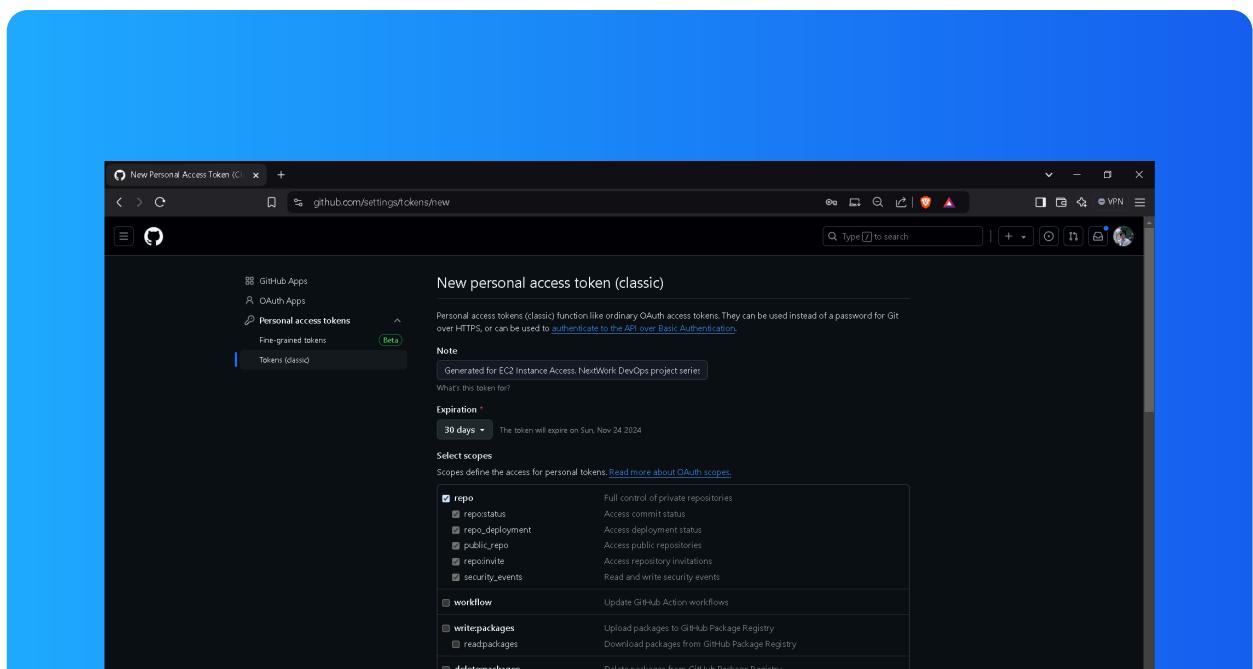


# GitHub tokens

GitHub authentication failed when I entered my password because I need to use a personal access token instead of my password for Git operations. This added layer of security helps protect my account from unauthorized access.

A GitHub token is a personal access token that serves as a secure alternative to using a password for authenticating API requests and Git operations. I'm using one in this project because it allows me to access and manage my GitHub repositories.

I could set up a GitHub token by logging into my GitHub account, navigating to the Settings menu, and then going to Developer settings. From there, I selected Personal access tokens and clicked on Generate new token.





# Making changes again

I wanted to see Git working in action, so I updated the index.jsp file with new content. I couldn't see the changes in my GitHub repo initially because I hadn't committed and pushed the updated file to the remote repository yet

I finally saw the changes in my GitHub repo after staging the modified index.jsp, committing the changes with a message, and pushing the commit to the remote repository. This made sure my latest updates were reflected online.

The screenshot shows a GitHub code editor interface. The left sidebar displays a file tree for the 'nextwork-devops-webapp' repository, specifically the 'src/main/webapp' directory, which contains 'WEB-INF' and 'index.jsp'. The right pane shows the content of 'index.jsp'. The code is as follows:

```
1 <html>
2 <body>
3 <h2>Hello Orven!</h2>
4 <p>This is my NextWork web application working!</p>
5 <p>If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :)</p>
6
7
8
9
10
11
12 </body>
13
14 </html>
```



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

