Task 1: Run Gitea in Codespace and Create an Initial Repo  
  
1. Set up Gitea:  
   - Run a Gitea server inside your Codespace.  
   - Use HTTPS for communication (SSH is not supported in Codespace).  
  
2. Create a Repository:  
   - Create a new repository on your Gitea server.  
   - Add a README.md file listing each student's name and roll number.  
  
3. Add Remote Repo:  
   - Use the following command to add your Gitea repository as a remote:  
     git remote add gitea <your\_gitea\_repo\_https\_url>  
   - Push your initial commit containing the README.md to Gitea.

Step 1: paste the below code in the terminal

services:

  gitea:

    image: gitea/gitea:latest

    container\_name: gitea

    environment:

      - DB\_TYPE=postgres

      - DB\_HOST=db:5432

      - DB\_NAME=gitea

      - DB\_USER=gitea

      - DB\_PASSWD=gitea

    restart: always

    volumes:

      - gitea:/data

    ports:

      - 3000:3000

    extra\_hosts:

      - "www.jenkins.com:host-gateway"

    networks:

      - webnet

  db:

    image: postgres:alpine

    container\_name: gitea\_db

    environment:

      - POSTGRES\_USER=gitea

      - POSTGRES\_PASSWORD=gitea

      - POSTGRES\_DB=gitea

    restart: always

    volumes:

      - gitea\_postgres:/var/lib/postgresql/data

    expose:

      - 5432

    networks:

      - webnet

volumes:

  gitea\_postgres:

    name: gitea\_postgres

  gitea:

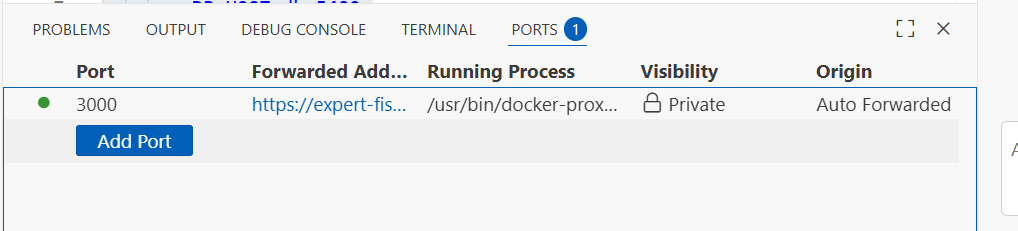
    name: gitea

networks:

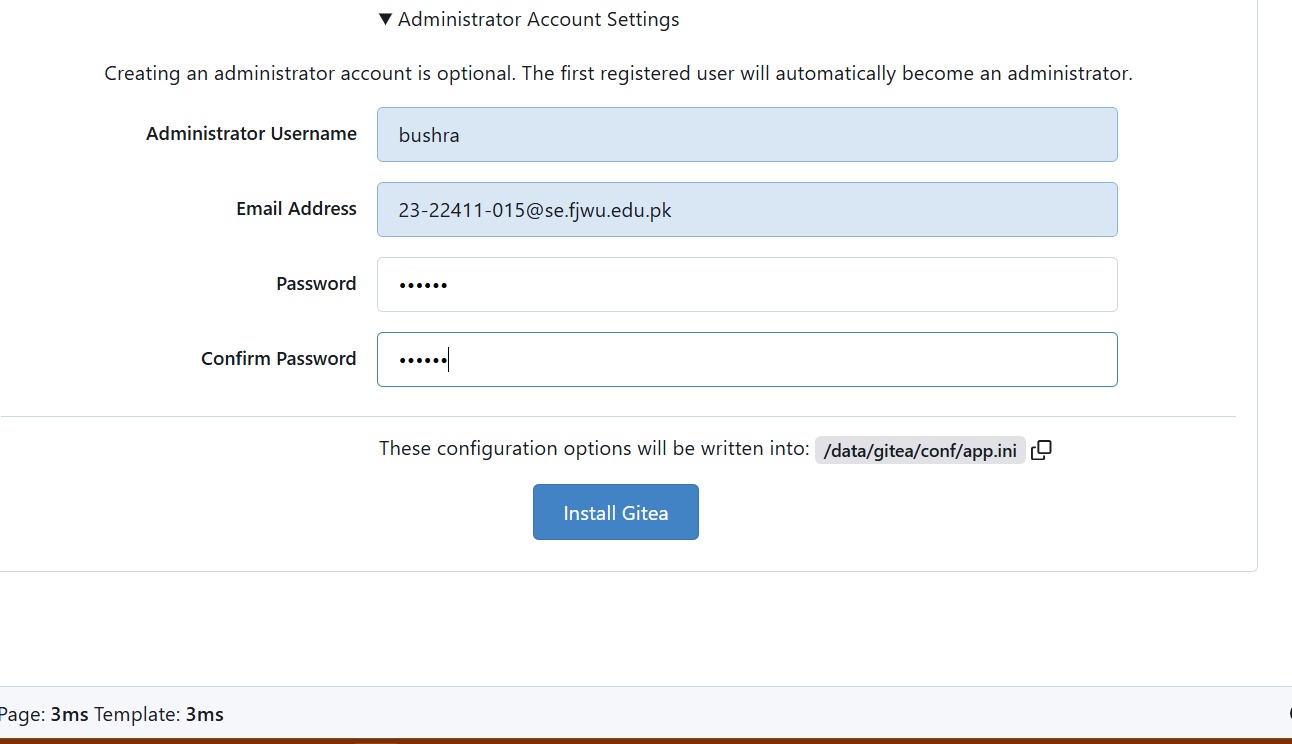
  webnet:

    name: webnet

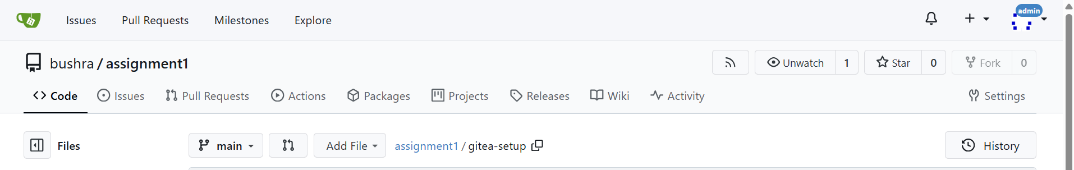
#    external: true



Step 2:install GITTEA:

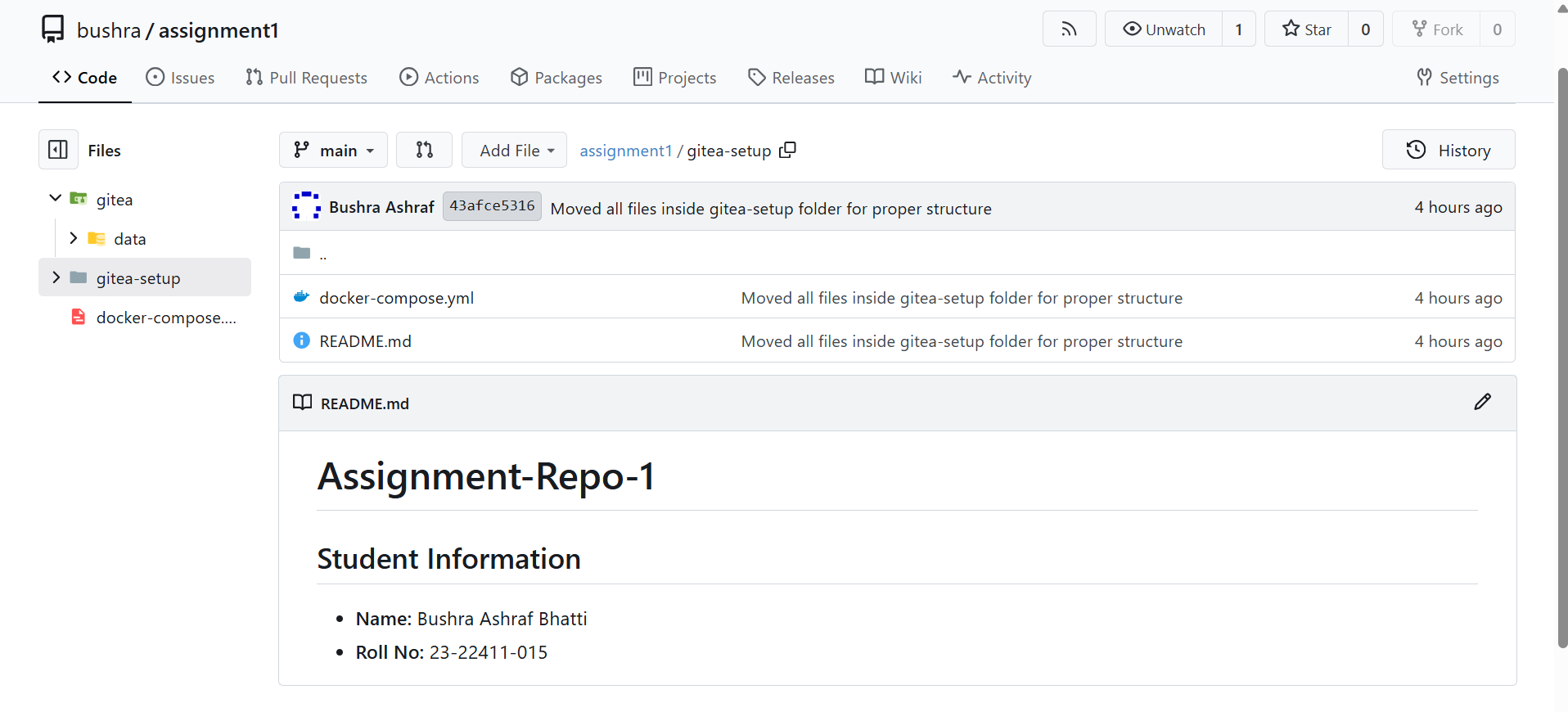


Step 3:Create a New Repository on Gitea and make it public:



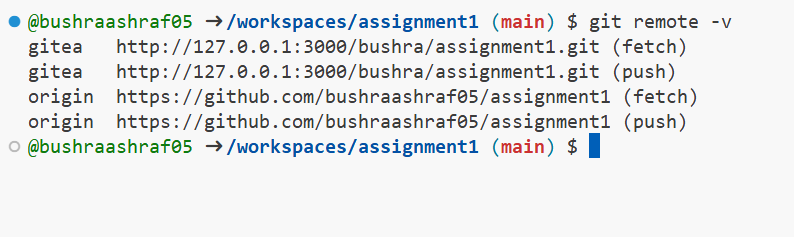
Step 4: Add a README.md file listing each student's name and roll number.

Step 5: Push your initial commit containing the README.md to Gitea.

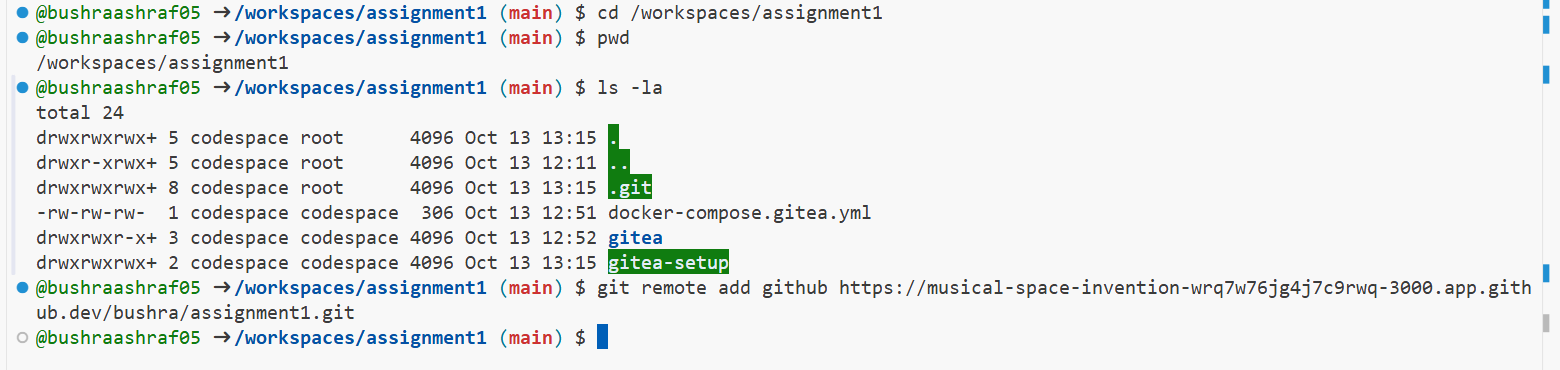


Task 2: Mirror README.md from Gitea to GitHub  
  
1. Continue Working with Your Existing Repository:  
   - You will use the same repository that you created and pushed to your Gitea server in Task 1.  
  
2. Create GitHub Repository:  
   - Create a new GitHub repository named assignment 1.  
  
3. Add GitHub as a Second Remote:  
   - Add your GitHub repository as a remote to your local repository:  
     git remote add github <your\_github\_repo\_https\_url>  
  
4. Push the README.md File to GitHub:  
   - Push the contents (including the README.md) from your local repository to GitHub.  
  
5. Verify Remotes:  
   - Run git remote -v and ensure both remotes (gitea and github) are listed.

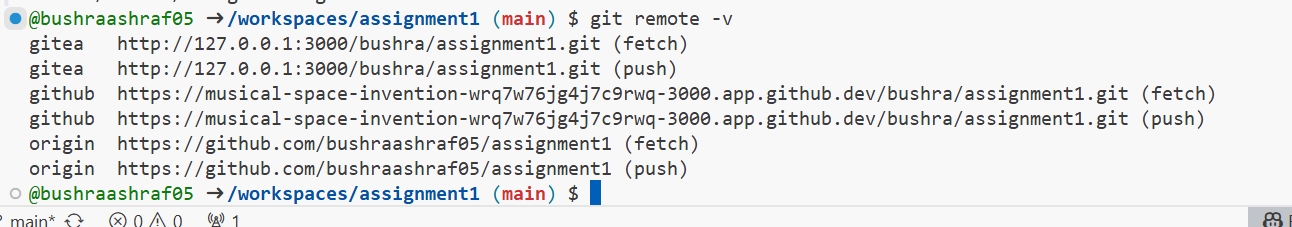
Step 1: Make sure your Codespace is open in your existing repo folder



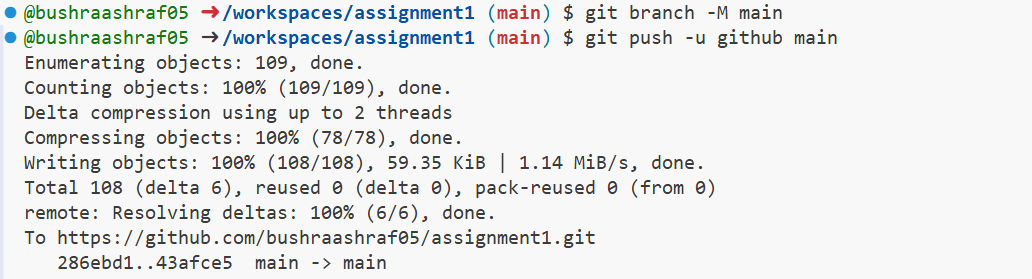
Step 2: Add GitHub as a remote (in Codespace terminal).

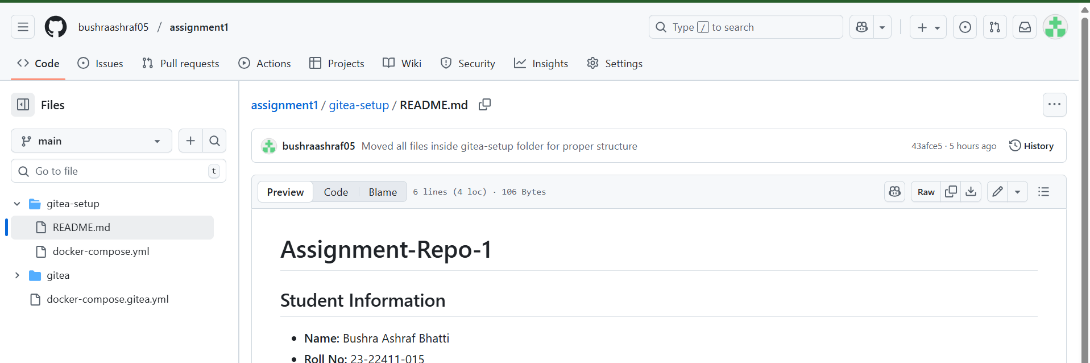


Step 3: Verify remotes



Step 4: Push your branch to GitHub:

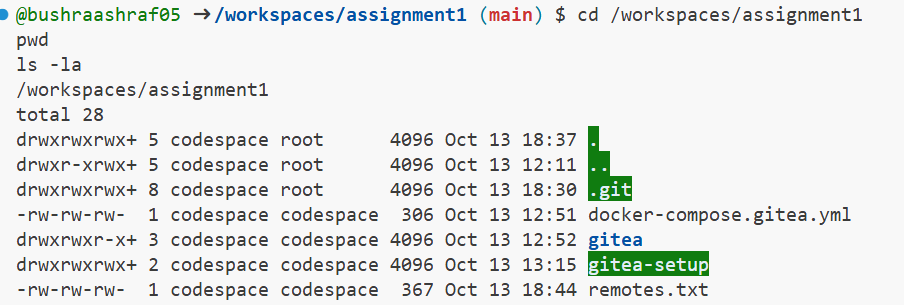




Link: <https://github.com/bushraashraf05/assignment1/tree/main/gitea-setup>

Task 3: Use Git LFS for Large Files  
  
1. Install Git LFS:  
   - Set up Git LFS in your local repository.  
  
2. Add Large Files:  
   - Add three files larger than 100 MB each to your repository.  
   - Track them using Git LFS:  
     git lfs track "\*.ext"  
   - Replace .ext with the appropriate file extension.  
  
3. Reference in Assignment Repo:  
   - Commit and push these large files to your GitHub assignment 1 repo.  
   - Ensure the files are referenced correctly in your repository history.

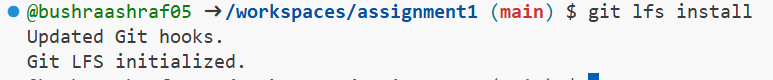
Step 1: Make sure we are in correct repo.

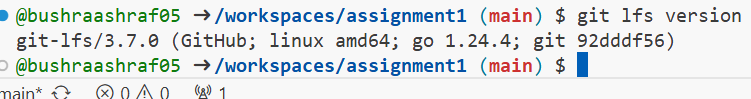


Step 2: Check if git-lfs is already installed:

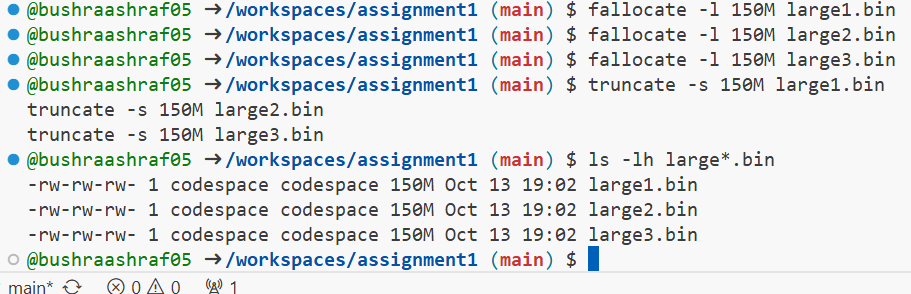


Step 3: Enable it for this repo:

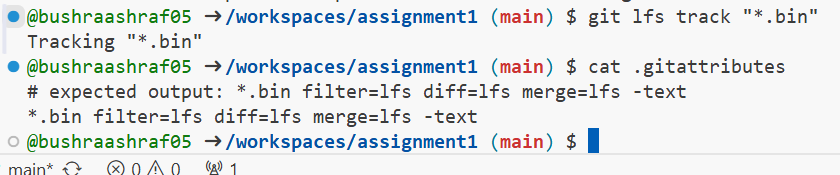




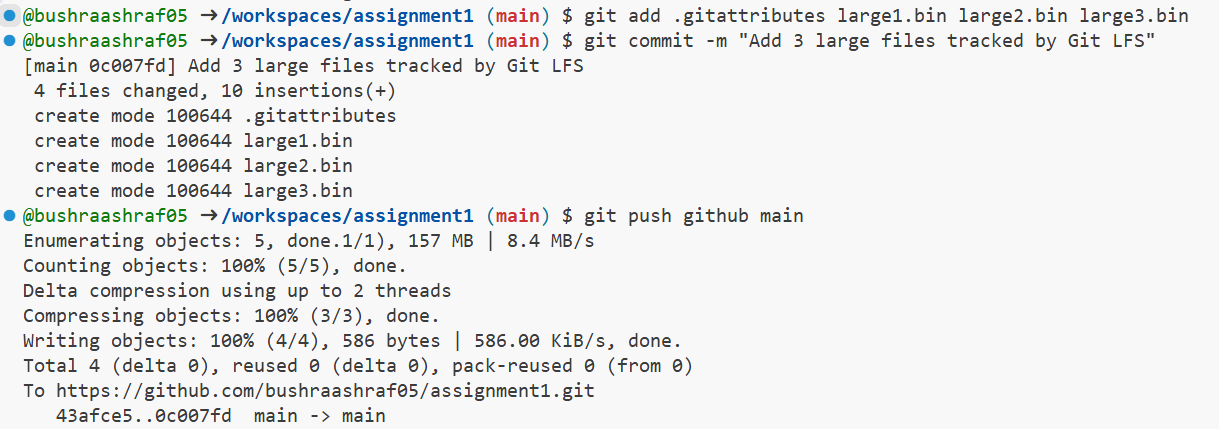
Step 4: Create three large dummy files



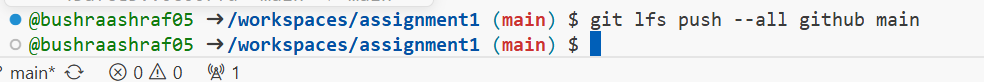
Step 5: Tell Git LFS to track these files:



Step 6: Stage, commit and push the LFS-tracked files to GitHub



Step 7: If you want to ensure all LFS objects are pushed

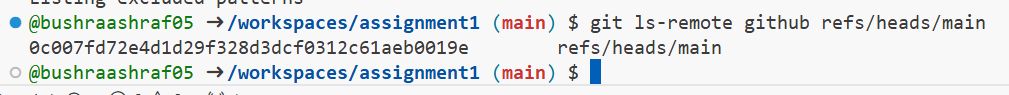


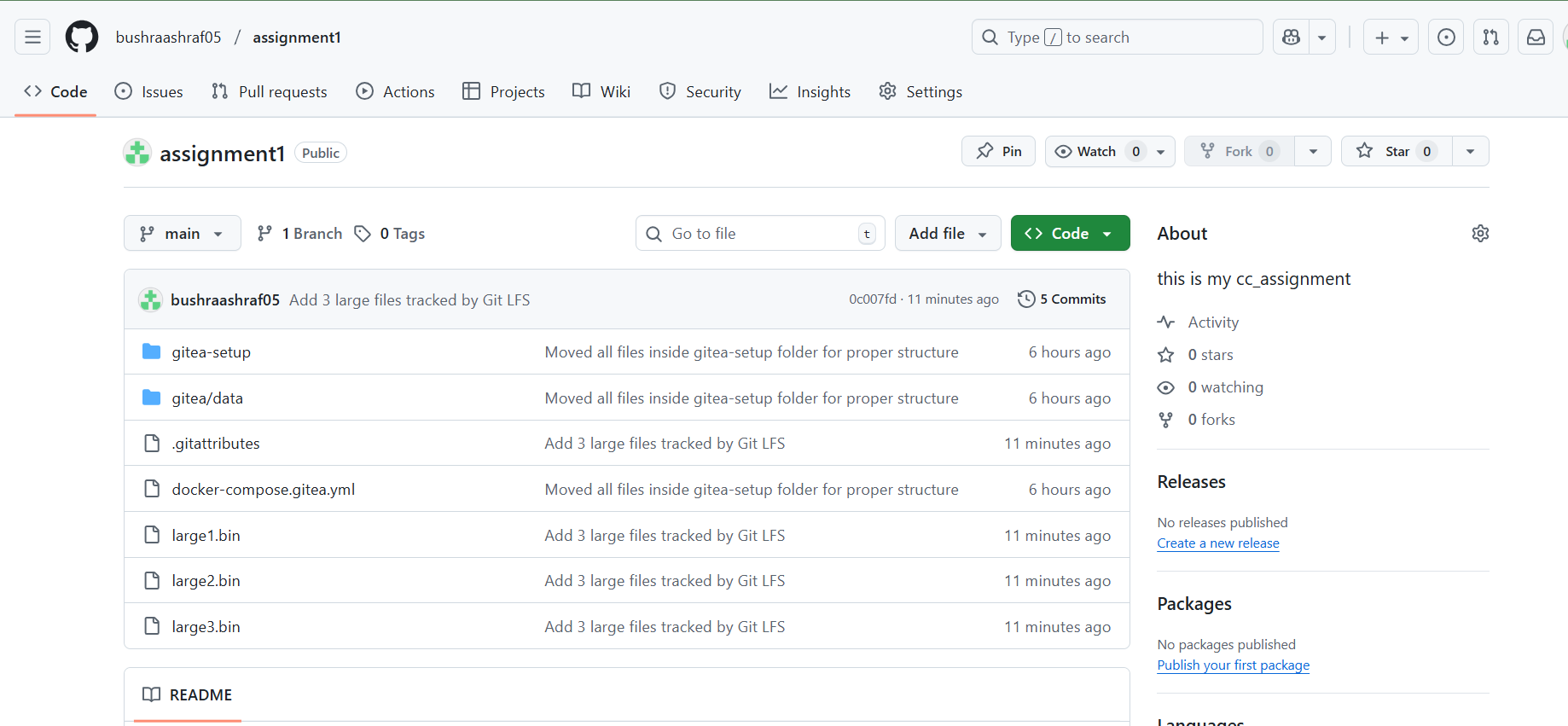
Step 8: Verify LFS tracking & that files were uploaded

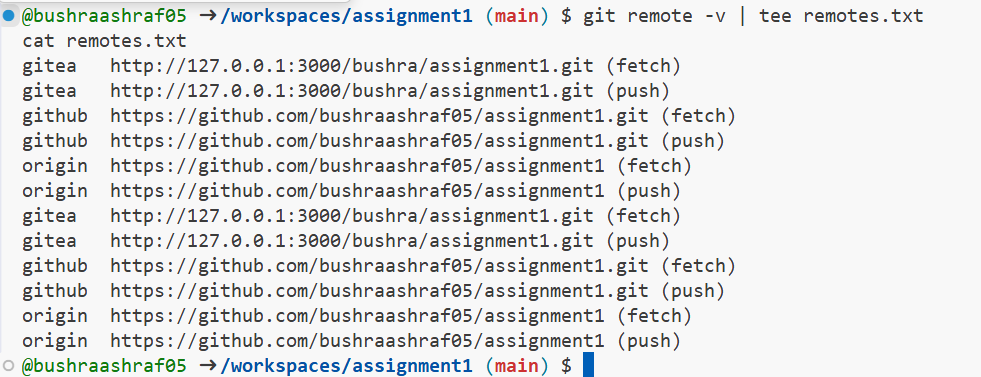
Check locally:

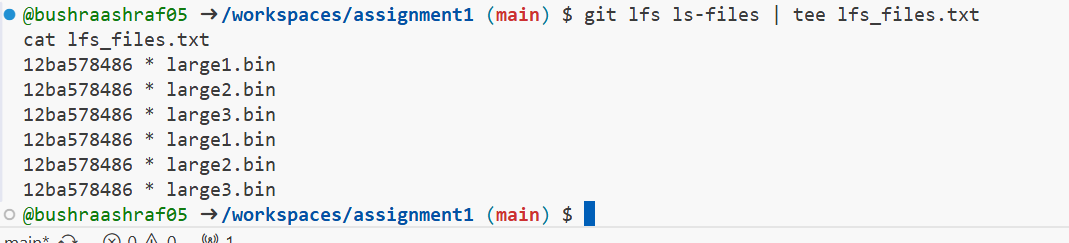


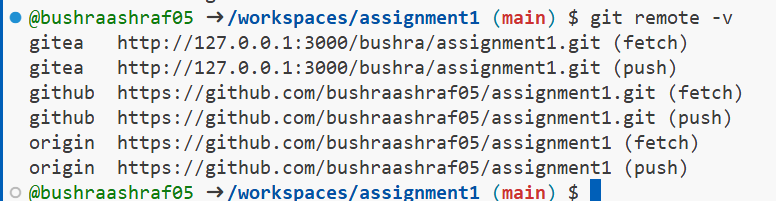
Check remotely:





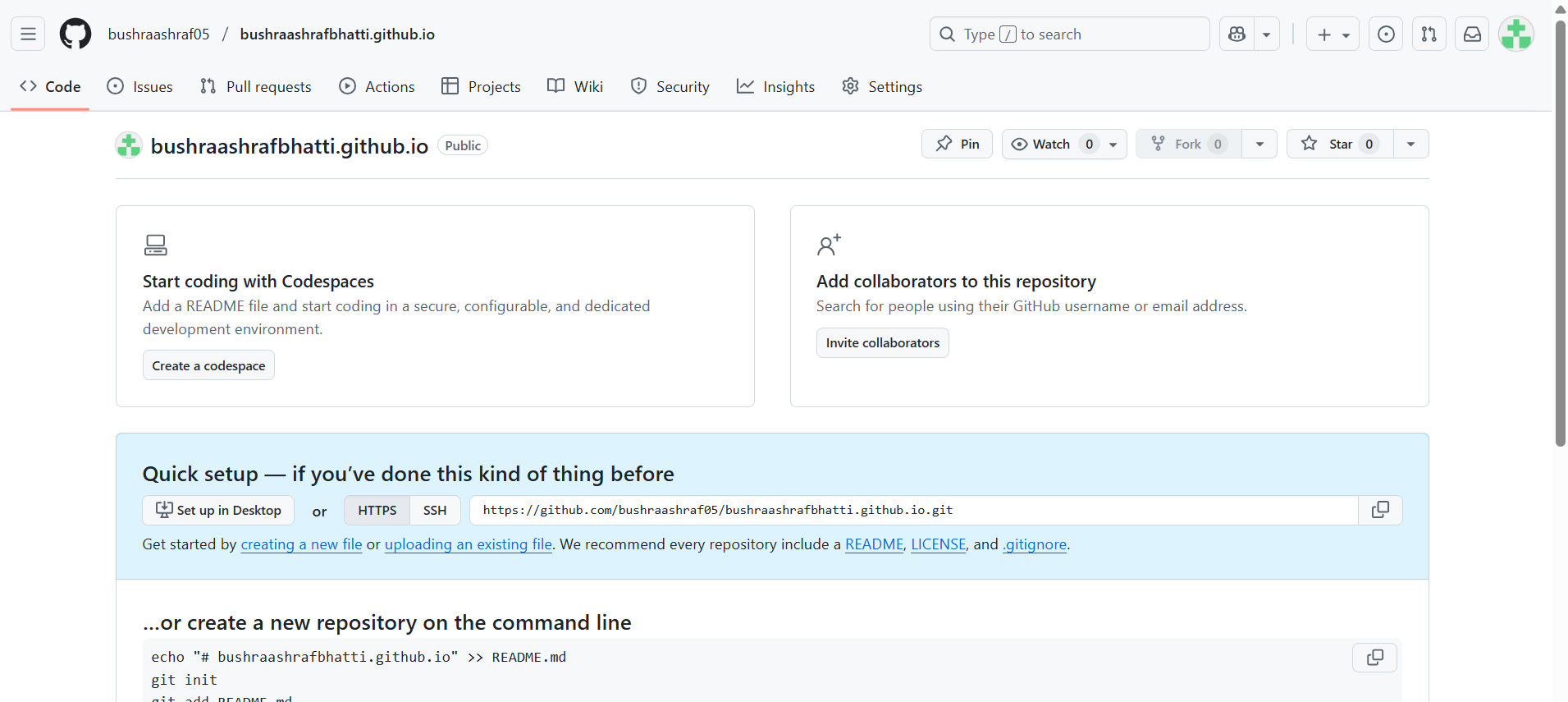




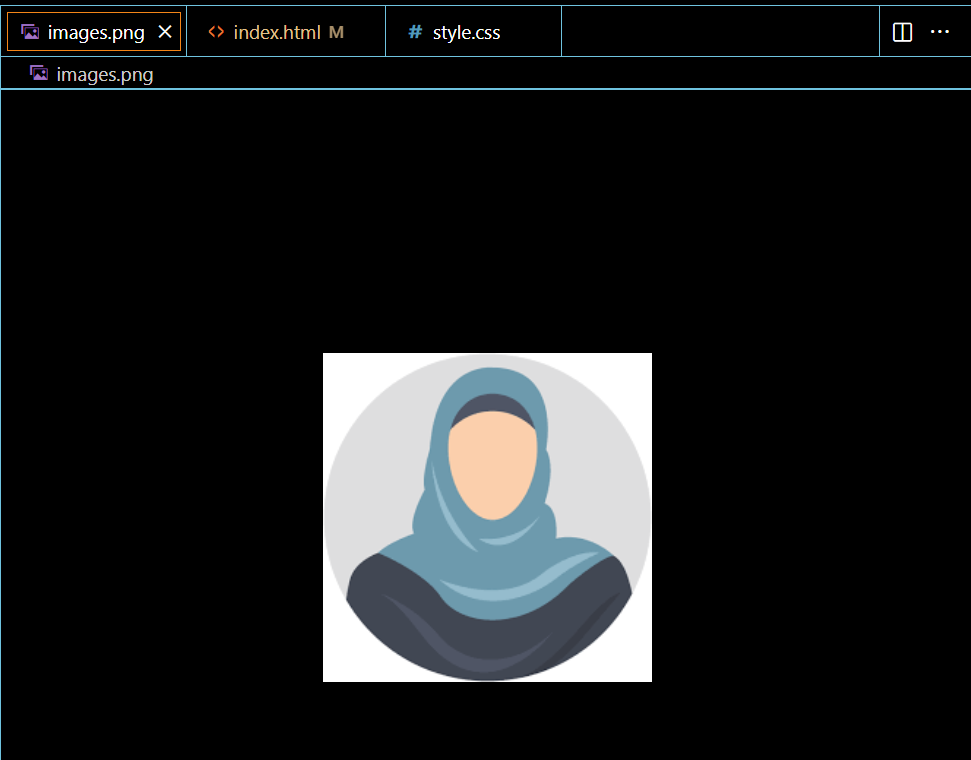


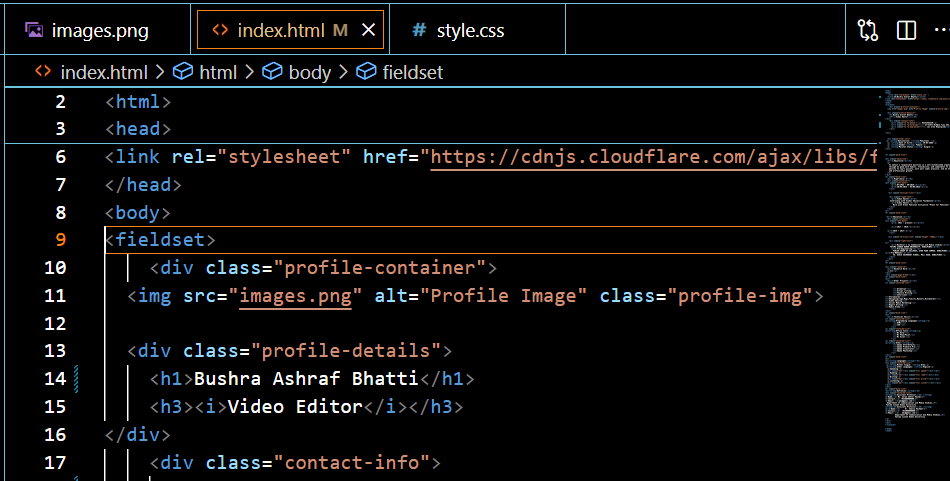
Task 4: Create a Portfolio/CV with GitHub Pages  
  
1. Create a new repository for GitHub Pages:  
   - Create a new repository named <your-username>.github.io.  
   - Replace <your-username> with your actual GitHub username (e.g., johnsmith.github.io).  
  
2. Design Your Portfolio/CV:  
   - Create your portfolio or CV in HTML/CSS (or use a static site generator).  
  
3. Publish with GitHub Pages:  
   - Push your portfolio/CV files to the <your-username>.github.io repository.  
   - Enable GitHub Pages in your repository settings if not automatically enabled.  
   - Publish your site and share the link.

step 1: create the public repository with the name “bushraashrafbhatti.githib.io:



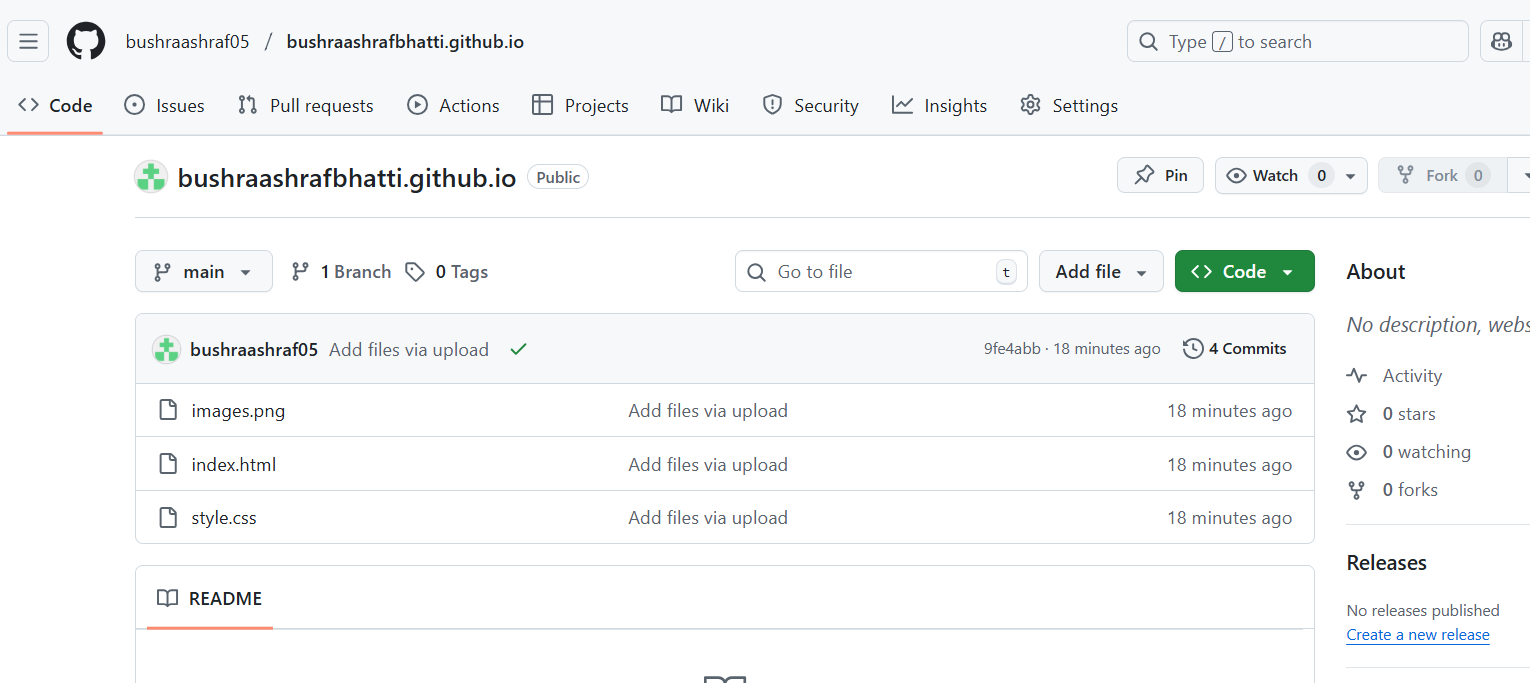
Three vs code files:



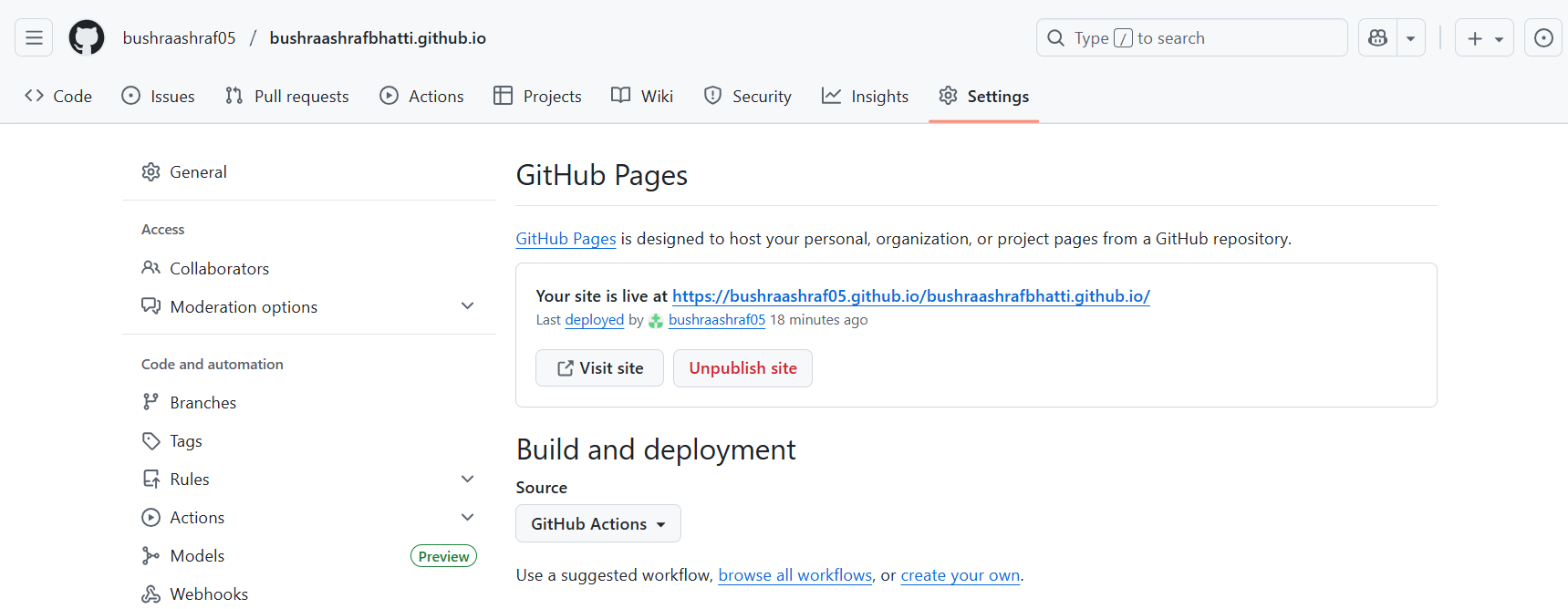




Step 2: Push the files to the repository through codespace:



Step 3: Enable GitHub Pages in your repository settings if not automatically enabled. Publish your site and share the link.



Step 4:

Link: [**https://bushraashraf05.github.io/bushraashrafbhatti.github.io/**](https://bushraashraf05.github.io/bushraashrafbhatti.github.io/)

