Experiment 01

Aim:

To implement Version control for different files/directories using GIT.

Theory:

In today's world of collaborative and distributed software development, **Version Control Systems (VCS)** play a crucial role. They enable teams to work on the same codebase simultaneously, manage code changes over time, and prevent accidental overwriting of each other's work. One of the most widely used VCS tools is **Git**, and platforms like **GitHub** have become industry standards for hosting and sharing repositories online.

What is Git?

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Developed by **Linus Torvalds** in 2005 (also the creator of Linux), Git allows every developer to have their own local repository, complete with full history and version tracking.

Key Features of Git:

- Distributed architecture (no central dependency)
- Fast branching and merging
- Lightweight commits
- Full traceability of changes (via commit logs)
- Offline-first operations

In Git, the complete history of the codebase is stored in the .git folder, allowing developers to switch between versions, revert to previous states, and compare changes with ease.

What is GitHub?

GitHub is a cloud-based platform built on top of Git, providing an intuitive interface for managing Git repositories online. It simplifies collaboration and enables developers to share their code publicly or privately with others. It supports additional features such as:

- Forking and cloning of projects
- Pull requests and code reviews
- Actions for automation
- Issue tracking and project boards

- Seamless integration with deployment platforms like **Vercel**

GitHub is especially useful for hosting open-source projects and collaborating with developers across the globe.

What is Cloning in Git?

Cloning means downloading the complete code repository, along with all branches and commit history, from GitHub to your local system. Once cloned, the project can be edited, versioned, or pushed to a different remote repository.

Command used:

```
>> git clone <repository-url>
```

What is Vercel?

Vercel is a modern hosting platform for static websites and front-end frameworks such as React, Next.js, and Vue. With a GitHub integration, Vercel automatically builds and deploys your site every time you push changes to the connected repository.

Vercel provides:

- One-click deployment
- Continuous deployment from GitHub
- Global CDN for fast performance
- Live preview links for every branch and PR
- Easy custom domain setup

Objective of This Experiment:

In this experiment, we focus on **understanding the basic Git workflow** with a practical exercise:

- Clone any public GitHub project (no need to make changes)
- Push the project as-is to a new repository under your GitHub account
- Deploy the project on **Vercel**
- Capture and attach screenshots as proof of each step

This process reflects real-world scenarios like:

- Testing a codebase before modifying it
- Forking and showcasing an open-source project

- Practicing Git fundamentals and deployment skills

Procedure:

Step 1: Git Installation (If not already installed)

Download and install Git from:
https://git-scm.com/downloads

To verify:

```
>> git --version
```

Step 2: Git Account Setup

Configure your name and email for Git to track your commits:

```
>> git config --global user.name "Khan Humayun"
```

```
>> git config --global user.email "humayunk.pvt@gmail.com"
```

(Optional: View your config)

```
>> git config --list
```

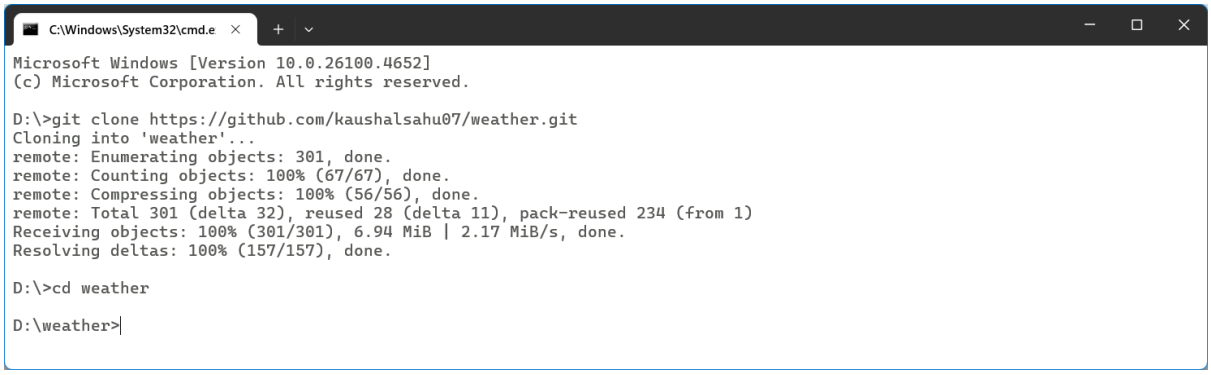
Step 3: Select a Public GitHub Repository

- Go to GitHub
- Search for any simple project (e.g., portfolio, weather app)
- Copy the **clone URL** (HTTPS)

Step 4: Clone the Repository Locally

```
>> git clone https://github.com/kaushalsahu07/weather.git
```

```
>> cd weather
```



Step 5: Create a New GitHub Repository

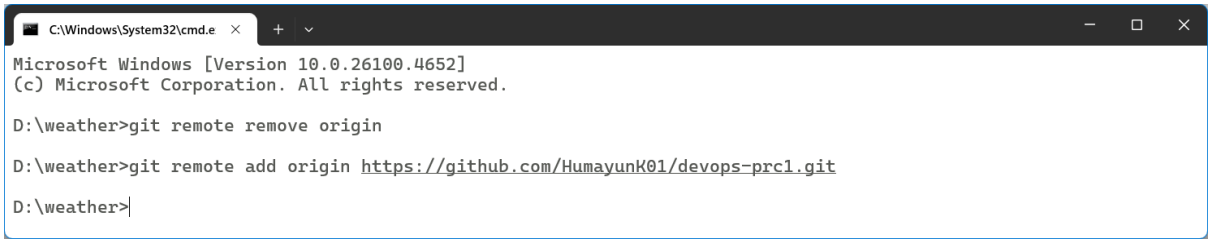
- Go to your GitHub account
- Click on **New Repository**
- Name it (e.g., devops-prc1)
- Do not initialize with README

Step 6: Connect Your Local Repo to Your GitHub Repo

Remove the old remote and add your own:

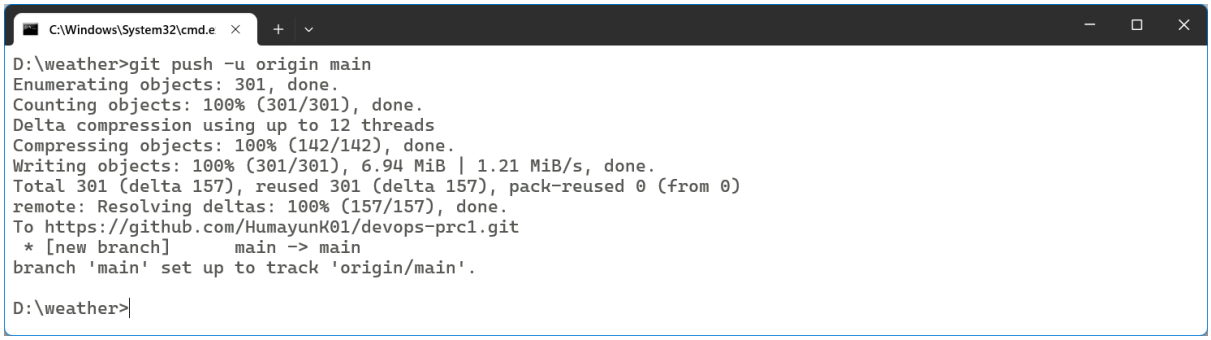
```
>> git remote remove origin
```

```
>> git remote add origin https://github.com/HumayunK01/devops-prc1.git
```



Step 7: Push to Your GitHub Repository

```
>> git push -u origin main
```



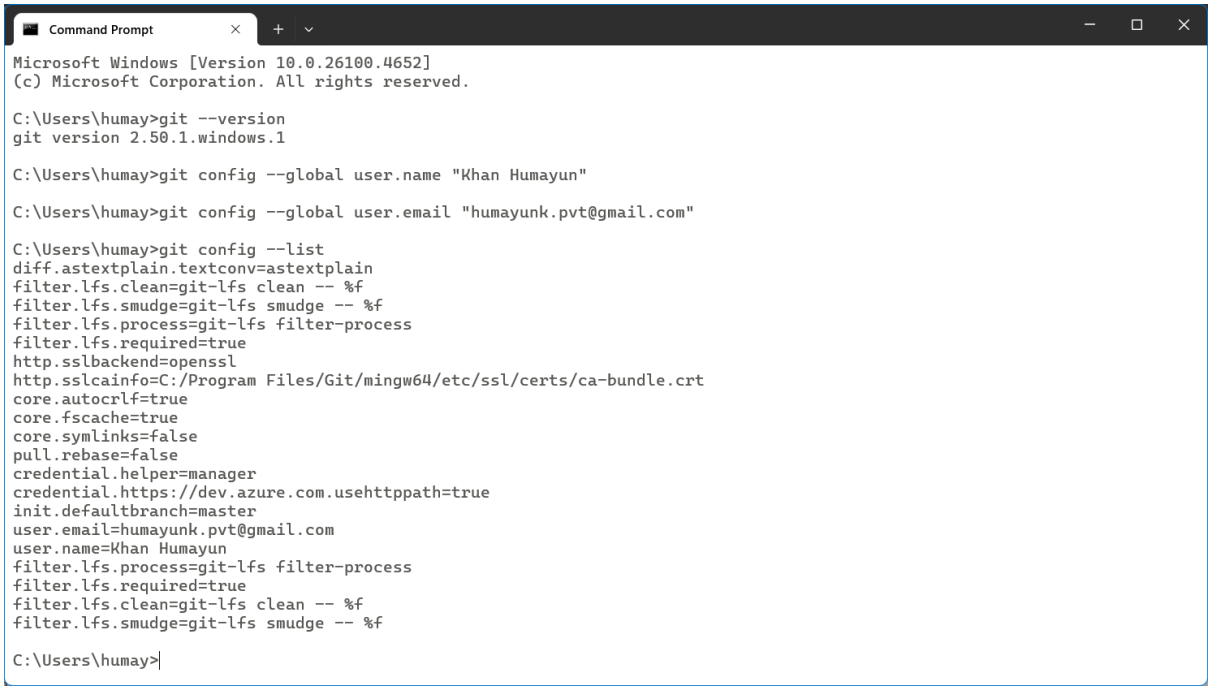
Step 8: Deploy on Vercel

- Visit https://vercel.com
- Log in with GitHub

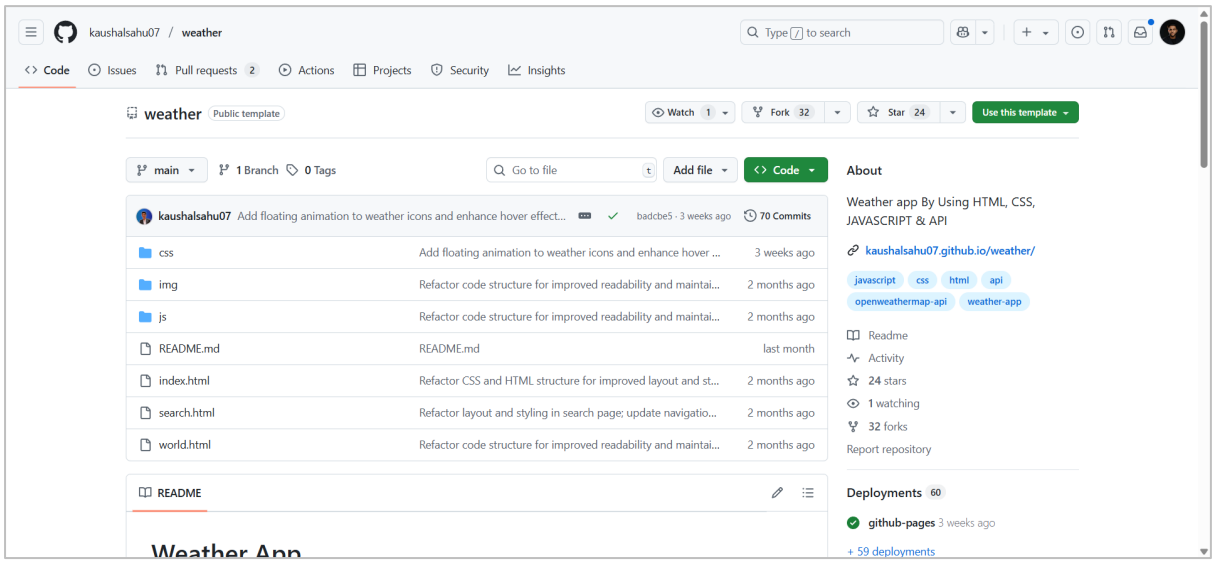
- Click **New Project**
- Select your devops-prc1
- Click **Deploy**

Outputs:

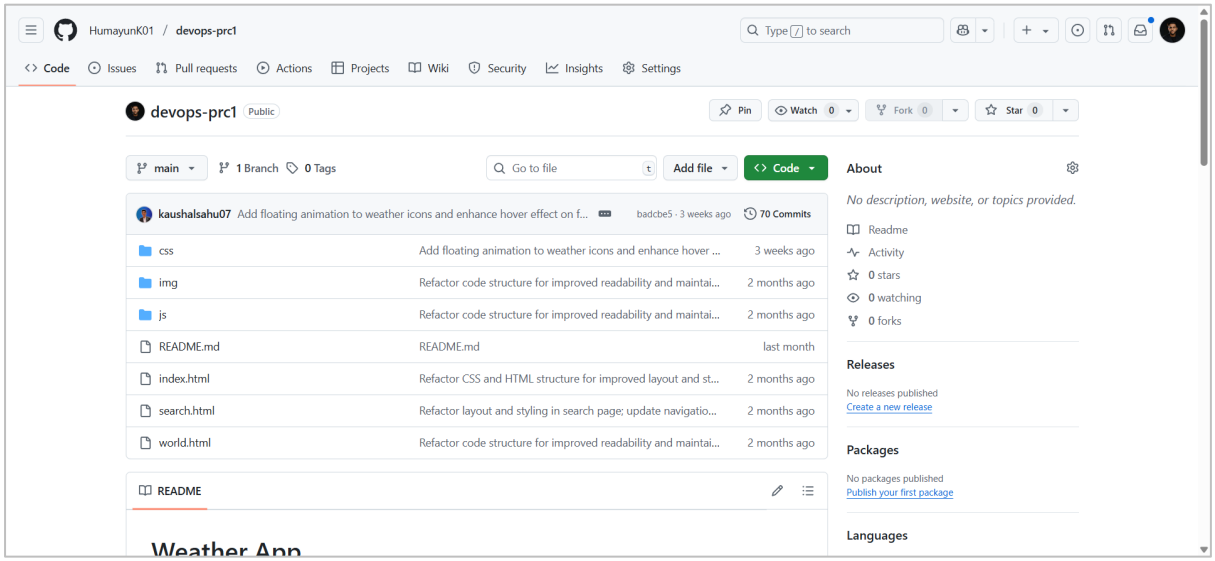
1. Git installation and config setup in terminal



2. Cloned project folder shown in Github

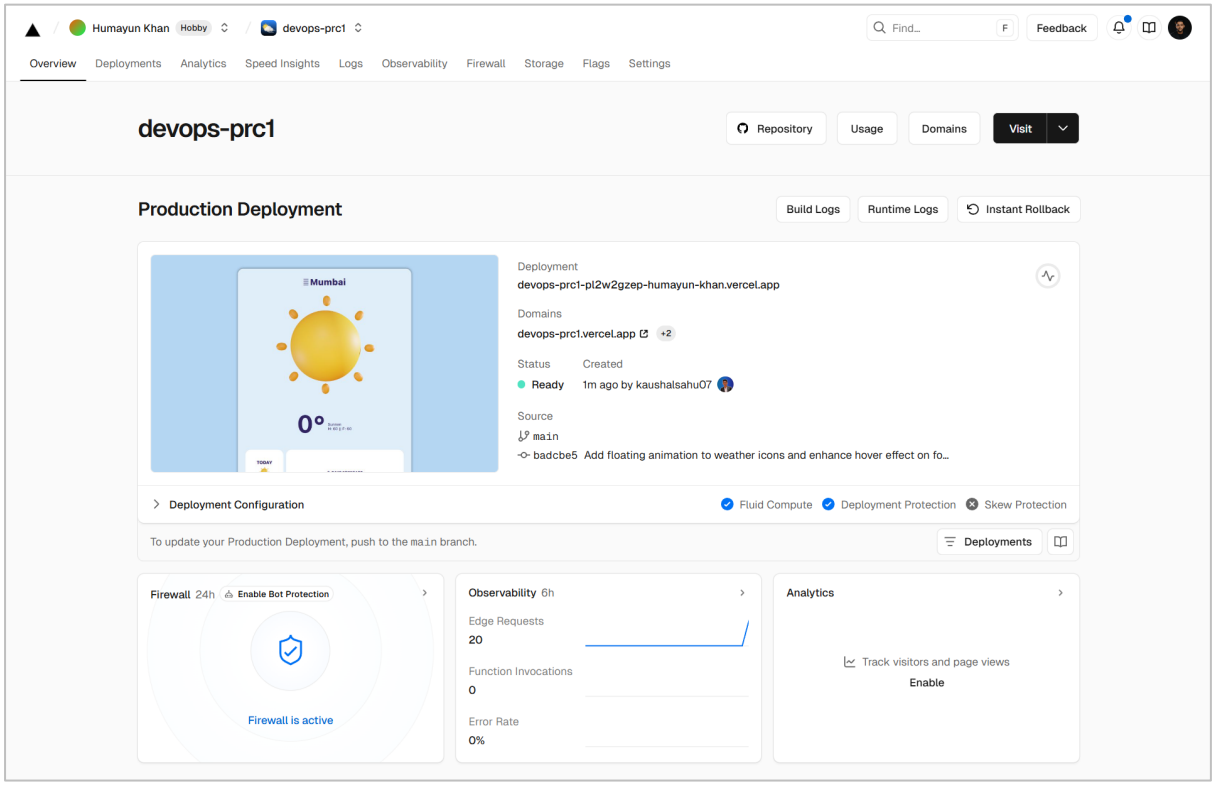


3. GitHub repo showing your own pushed project



4. Vercel dashboard with live deployment URL:

https://devops-prc1.vercel.app/



Conclusion:

The experiment was successfully completed. We cloned a public GitHub repository, pushed it to our own GitHub account without making changes, and deployed it on Vercel for live preview and access.