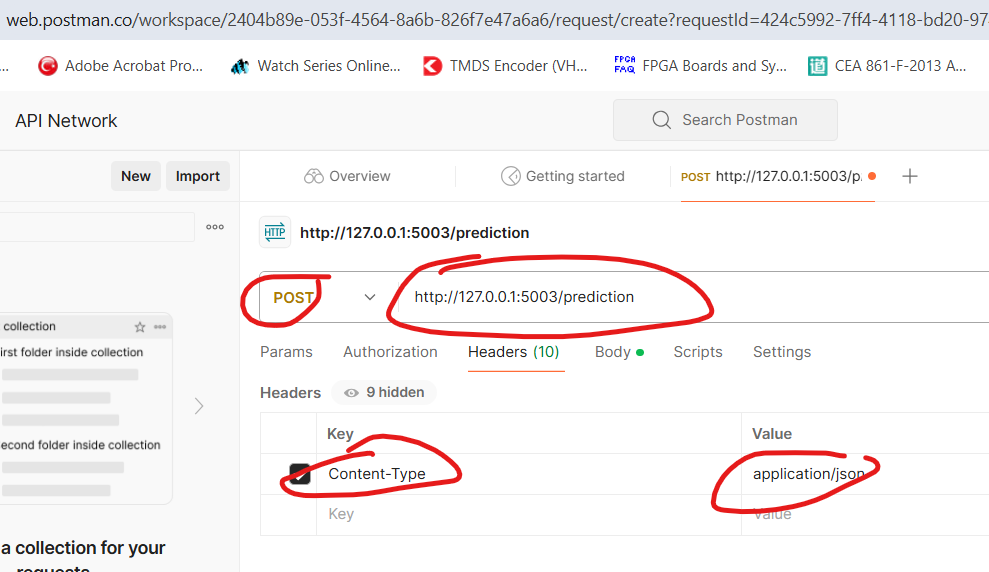
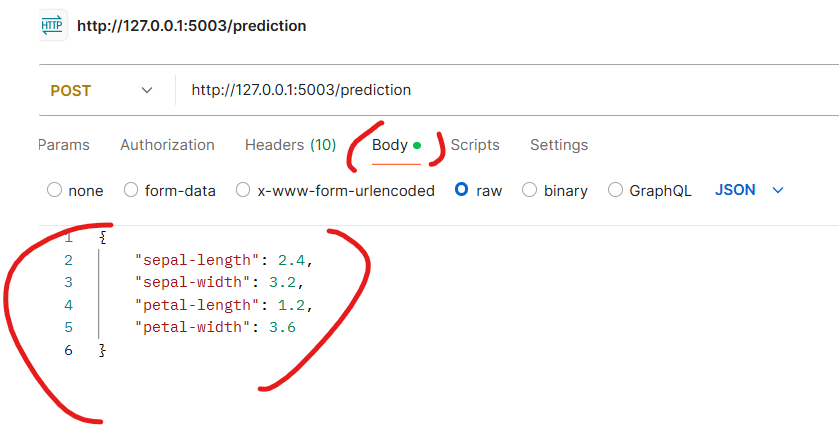
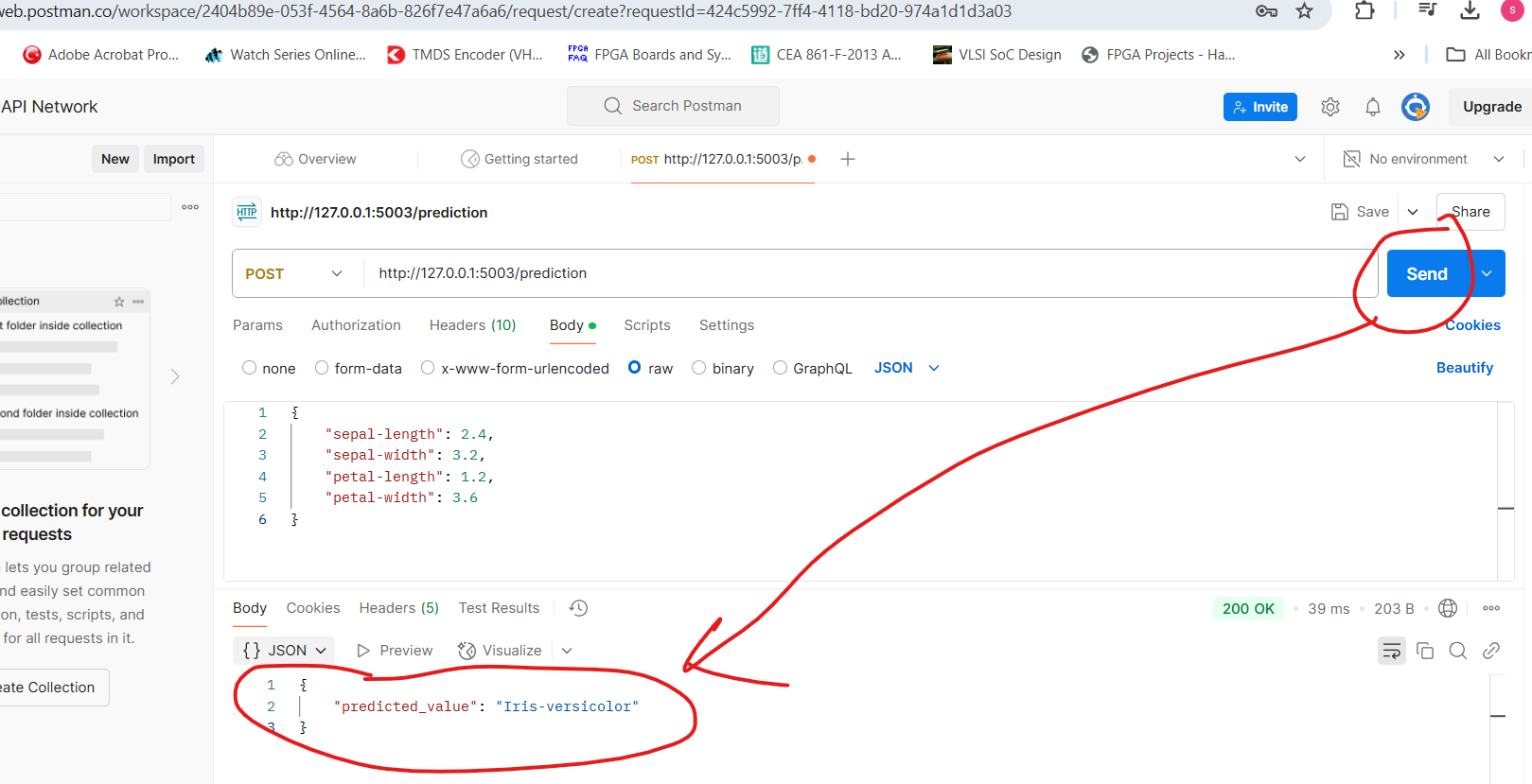
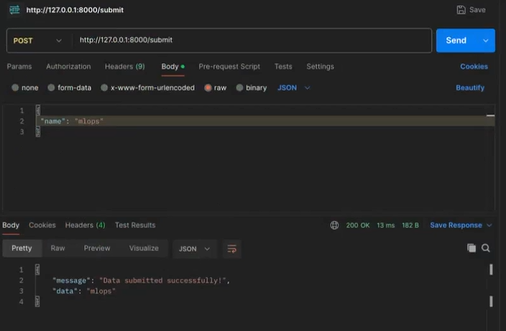
POSTMAN –

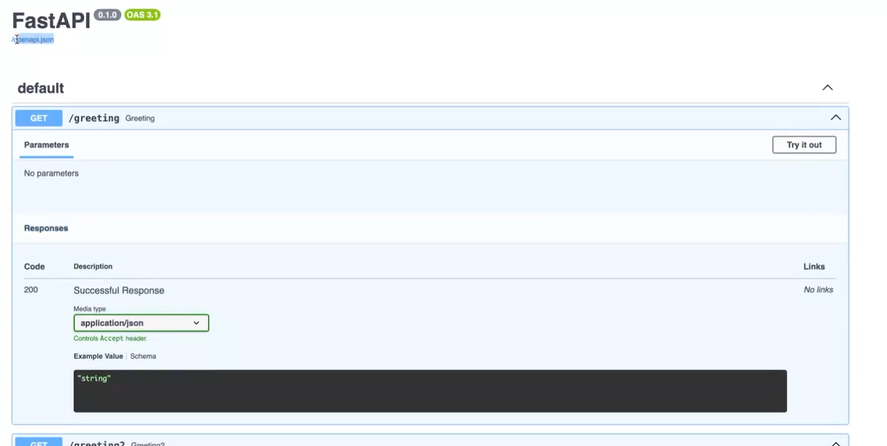


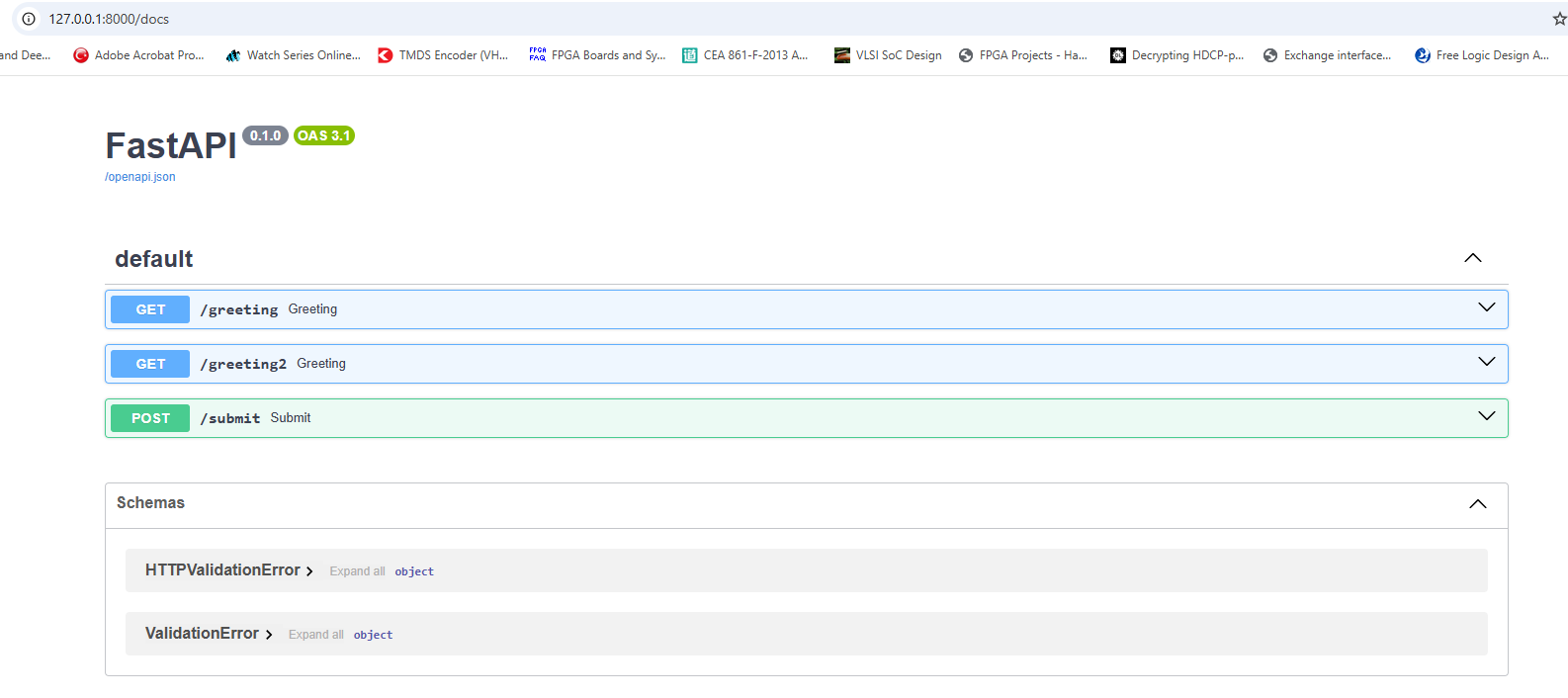


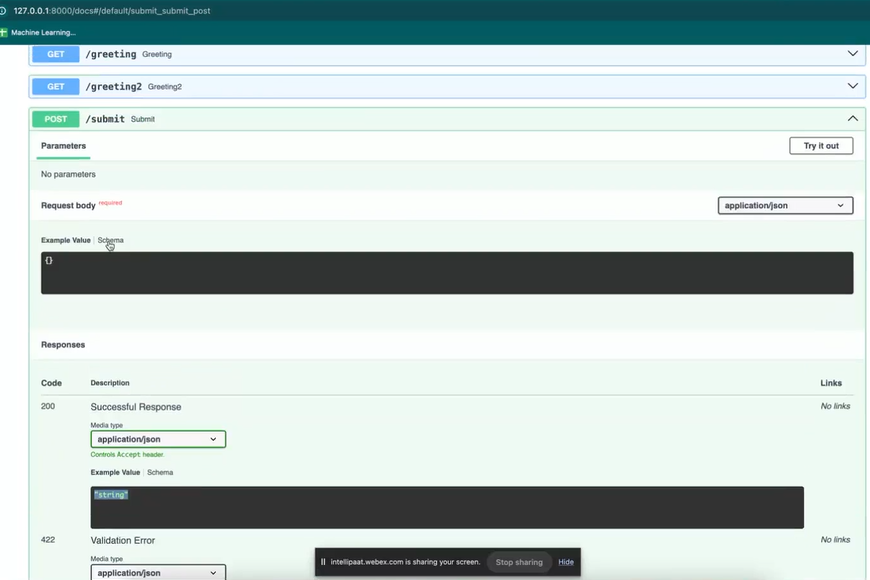


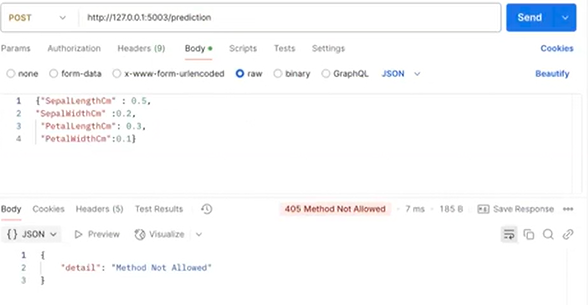
Using FAST API











DOCKER –

1. Terminal –
2. docker ps -a : will show all running containers
3. wsl –update : For updating to run linux cmd on windows Window subsystem for linux-wsl
4. Look for images – hello world or download from hub.docker.com - <https://hub.docker.com/r/aminehy/docker-streamlit-app>

Run app cmd in terminal of docker –

docker pull aminehy/docker-streamlit-app

docker run -ti --rm aminehy/docker-streamlit-app:latest

For compiling & running .py –

python -m app.py

1. Open Terminal –

**pip install pipreqs**

1. Run for creating requirements.txt file automatically –

**pipreqs . –force**

1. Building docker image, name of saurabh given to image –

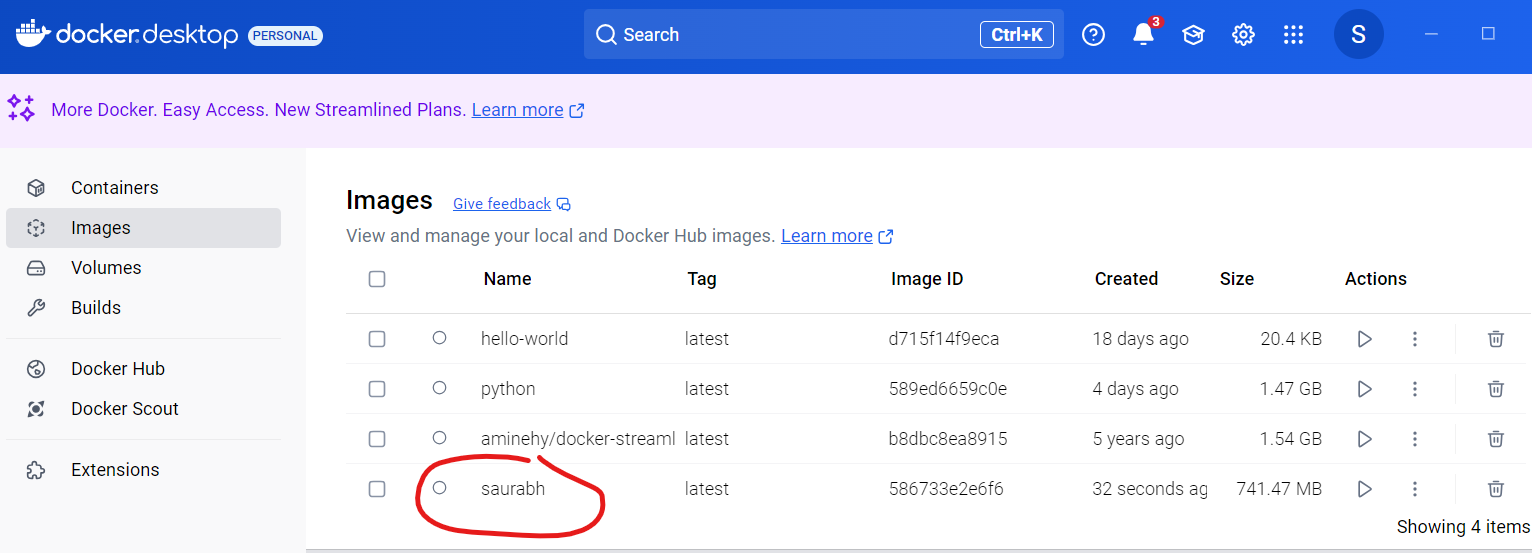
**docker build -t saurabh .**

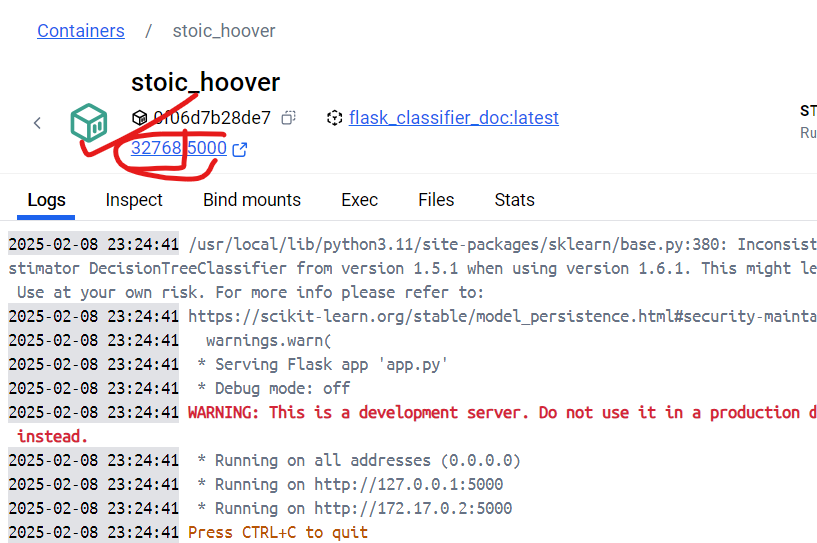
**docker run -p 5002:5002 flask\_classifier\_doc**

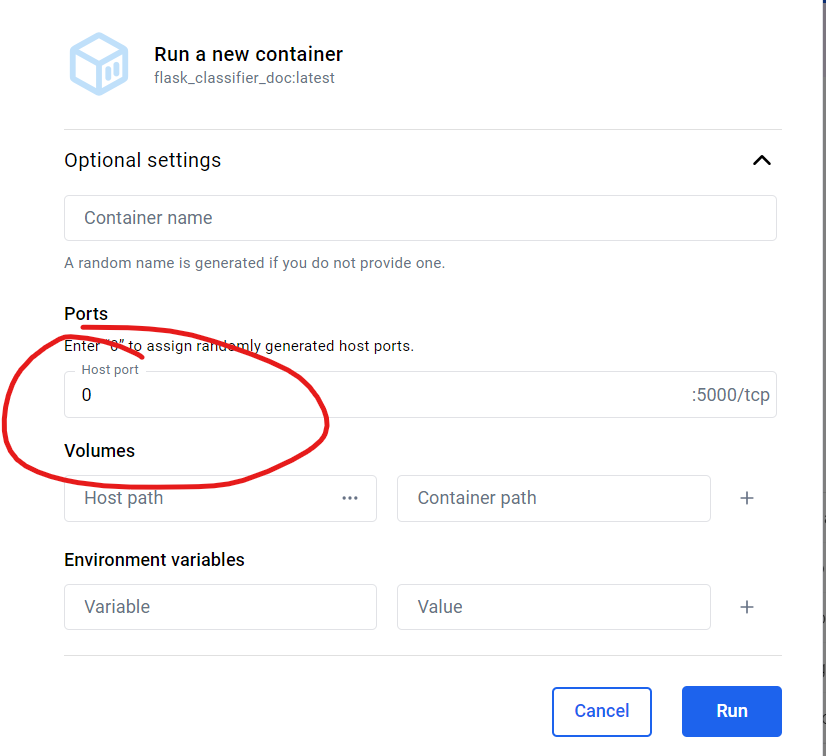
**run for random port & go to–**

[**http://localhost:32768/**](http://localhost:32768/)

**http://127.0.0.1:32768/**

****

****

****

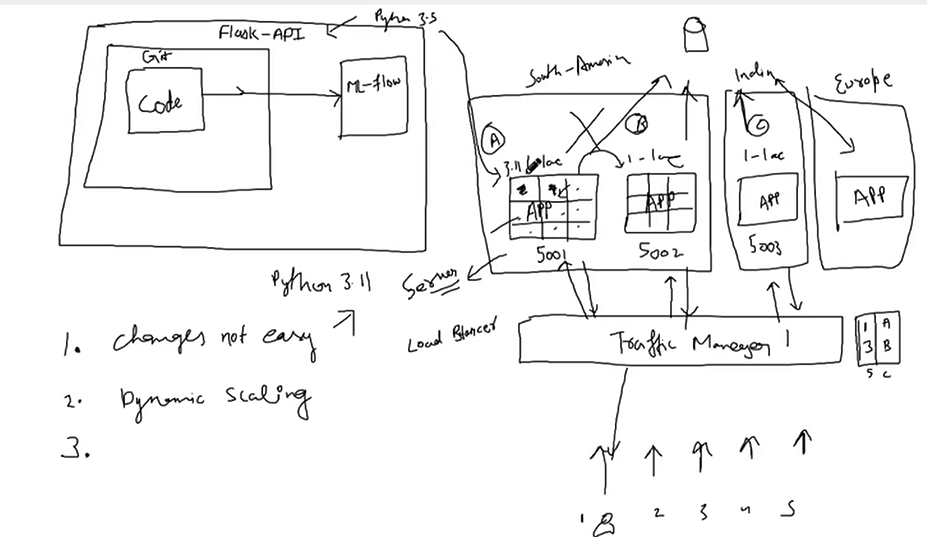
docker build -t mlops .

docker run -p 5004:5004 mlops

docker build -t iris\_classifier .

docker run -p 5004:5004 --name iris\_classifier -d iris\_classifier

docker-compose up --build -d



## Problem with Traffic Manager when having multiple apps running horizontally-

1. Changes not easy
2. Dynamic scaling

**Git CMDS –**

1. gh auth login – Login to account using github CLI
2. gh auth status – Check GitHub CLI login status
3. Create new model from cmdline -

gh repo create mlops-vnit-main\_0 --public --source=. --remote=origin

1. Any new repo initialize –

git init

git add .

git commit -m "Initial commit"

1. Push to github –

git branch -M main

git push -u origin main

1. Check the username details -

git config --global user.name

git config --global user.email

1. Verify the Remote Repository URL-

git remote -v

git remote set-url origin <https://github.com/your-username/mlops-vnit-main_0.git>

1. Check Files That Have Changed in the Last Commit-

git show --name-only

1. To see a list of files changed and who changed them, use:

git log --pretty=format:"%h %an %ad %s" --date=short --name-only --diff-filter=AM

 %h → Commit hash

 %an → Author name

 %ad → Date

 %s → Commit message

 --name-only → Shows changed files

 --diff-filter=AM → Only lists added/modified files (ignores deleted files)

1. If you want to see changes made by a specific user (e.g., "JohnDoe"):

git log --author="JohnDoe" --pretty=format:"%h %an %ad %s" --name-only

1. See File History and Who Modified It

git log --follow --pretty=format:"%h %an %ad %s" -- path/to/file

1. Find Who Last Modified a Specific Line in a File

git blame path/to/file

1. Check All Changes Not Yet Pushed-

git diff origin/main