1. Clone github repository to local folder
2. Copy the link from github and open git bash in your local folder and write “git clone <link>”
3. To go into the folder type cd <name of the folder>(You can write the first few letters of the folder and press TAB to autocomplete it). Type “ls” to know the contents of the folder
4. Create folder structure in template.sh using bash script “mkdir” to create directories, “touch <folder>/<file\_name>”
5. In git bash type “sh template.sh” to create the folder structure
6. Python must be installed for pip to work
7. “pip list” to contents in each folder
8. Create an environment with “conda create -n medibot python=3.17.3 -y” and activate it with “conda activate medibot”.
9. Add necessary libraries in requirements.txt
10. In setup.py find\_packages() comes from **setuptools** and is used in setup.py to automatically discover your project’s Python packages (directories containing \_\_init\_\_.py). A **package = a folder with \_\_init\_\_.py + Python files**.  
    It helps you structure your project and make it reusable (like libraries you install with pip). “This folder is a package. You can import from it.”
11. “pip install -r requirements.txt” to install libraries
12. Copy the functions from trials.ipynb to helper.py and the prompt in prompt.py
13. To store the vectors in the knowledge base create a file “store\_index.py” in src
14. Run the store\_index.py in git bash using “python store\_index.py”
15. Create a flask app by creating chat.html and style.css for the web app and run app.py in git bash
16. Go to AWS Console and deploy the app
17. Go to IAM > Users >User name > Add policies(EC2 and ECR)
18. After creating IAM user go to the username > Security credential > access key > CLI > Create access key > download access key and secret access key as csv
19. Next create ECR instance from AWS homepage
20. Next create EC2 i(Virtual machine) nstance from AWS homepage > Launch instance > give a name > Operating system: Ubuntu > Machine configuration(t2 large) > Key pair login > Network settings(Select both HTTPS options) > Storage . after creating then click view all instances. Click on the instance id and then connect
21. Run the linux commands in the terminal
22. To configure the EC2 instance go to your github repo > settings > Actions>runners > new self hosted runner > select linux and run the linux commands in the aws terminal to connect our github to the aws and whenever there is a new code in github it will trigger the CICD > name of runner (self-hosted)
23. Add secrets like API keys in github left tab “Secrets and credentials” > Actions > Repository secrets(Add all secrets as mentioned in github repo)
24. Create a docker file In your project folder
25. Create .github folder and inside it create a workflows folder and inside it create a cicd.yaml file and inside it write the cicd github actions command
26. Commit to github and sync