Master Cheat Sheet — Chatbot Starter Project

Full Project Cheat Sheet

Chatbot Starter Project — Cheat Sheet

Project structure
```
chatbot-starter/
app.py
.env # secrets (local only; NOT in git)
.gitignore
Procfile # for Render
requirements.txt
runtime.txt # optional; pins Python for Render
static/ # your frontend files (index.html, js, css)
venv/ # local Python virtual environment
```

Python & virtualenv
```bash
python3 -m venv venv # create venv (run once)
source venv/bin/activate # activate venv
deactivate # leave venv
pip install -r requirements.txt
python app.py # run locally (http://127.0.0.1:5000)
```

Updating code & redeploying (typical loop)
1. Edit files locally (e.g., app.py)
2. Test locally with `python app.py`
3. Stage & commit changes:
```bash
git add .
git commit -m "feat: describe your change"
```
4. Push to GitHub:
```bash
git push
```
5. Render auto-deploys on push (or use "Manual Deploy → Deploy latest commit")

If environment vars changed on Render → click **Manual Deploy → Clear build cache & deploy**.

Environment variables
- Local: in `.env` (ignored by git)
- Render: set via dashboard → Environment

Git basics
```bash
git status
git add .
git commit -m "feat: your message"
```

git push
```

## Render deployment
- **Procfile**:
```
web: gunicorn app:app --bind 0.0.0.0:$PORT --timeout 120
```
- **Manual Deploy** options:
- Deploy latest commit → after `git push`
- Clear build cache & deploy → if deps/env changed

## Google Sheets CSV sanity check
```bash
curl -sL "https://docs.google.com/spreadsheets/d/e/.../pub?output=csv" | head -n 10
```
Header row should be:
```
question_en,answer_en,question_es,answer_es
```

Secrets safety
```bash
printf "# secrets\n.env\n" >> .gitignore
git rm --cached .env
git commit -m "ignore .env"
git push
```

SSH for GitHub
```bash
ssh-keygen -t ed25519 -C "you@example.com"
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
cat ~/.ssh/id_ed25519.pub # Add to GitHub → Settings → SSH keys
```

Handy Linux commands
```bash
pwd # current dir
ls -l # list files
cd # change dir
rm # remove file
cp # copy
mv # move/rename
cat # print file
curl # fetch URL
```

Nano shortcuts
- Save: Ctrl+O → Enter
- Exit: Ctrl+X
- Search: Ctrl+W
- Go to line: Ctrl+_
```

Daily Update & Deploy — Quick Reference

Daily Update & Deploy — Quick Reference

1. Activate virtual environment
```bash
cd ~/chatbot/chatbot-starter
source venv/bin/activate
```

2. Run locally (optional test)
```bash
python app.py
# Visit http://127.0.0.1:5000 and /debug/csv to test
```

3. Stage & commit changes
```bash
git status
git add .
git commit -m "feat: describe your change"
```

4. Push to GitHub (Render auto-deploys)
```bash
git push
```

5. On Render
- **Deploy latest commit** after code changes
- **Clear build cache & deploy** if dependencies or environment variables changed

6. Environment Variables (if changed)
Render → Service → Environment → Add/Edit → Save → **Manual Deploy**

7. Local cleanup commands (optional)
```bash
deactivate # leave venv
rm -r directory # delete a directory
rm file # delete a file
```

8. Quick sanity checks
```bash
git log --oneline -- .env # should be empty if secrets ignored
curl -sL "$CSV_URL" | head -n 10 # preview CSV rows
```

Common Linux Commands

Action	Command	Example / Notes
Print working dir	pwd	Shows current directory
List files	ls -l	Detailed list
Change dir	cd path/to/dir	cd .. goes up one dir
Make directory	mkdir dirname	mkdir new_project
Remove empty dir	rmdir dirname	Fails if dir not empty
Remove dir & contents	rm -r dirname	rm -rf dirname (force, dangerous)
Remove file	rm filename	rm old.txt
Copy file	cp src dst	cp a.txt b.txt
Move/Rename file	mv src dst	mv old.txt new.txt
View file contents	cat filename	cat readme.txt
Edit file (CLI)	nano filename	Ctrl+O save, Ctrl+X exit
Download URL	curl -O URL	curl -O file.zip
Search text	grep 'text' file	grep 'main' app.py
Find files	find . -name '*.py'	Finds all Python files
Show disk usage	du -sh	Shows size of current dir
Show free space	df -h	Shows free space on drives