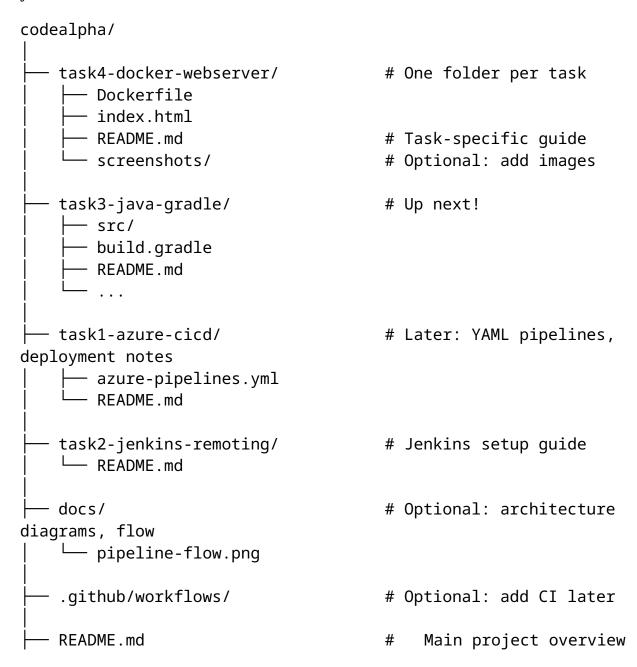
Great question — and excellent thinking! Yes, since this is for a **CodeAlpha internship**, you **absolutely should** create a professional GitHub repository (e.g., codealpha) to showcase your work.

Even though the tasks don't involve thousands of lines of code, **DevOps is about process**, automation, and documentation — and your reposhould reflect that.

What to Include in Your codealpha GitHub Repo

Here's a **professional structure** that will impress reviewers and show you're serious:



What to Add (Even With "No Code")

You don't need complex code — focus on clarity, learning, and reproducibility.

For Each Task, Include:

| File | Purpose |
|---------------|--|
| README.md | Explain what you did, how to run it, and what you learned |
| Config files | Dockerfile, build.gradle, Jenkinsfile, azure-pipelines.yml |
| Scripts | Any shell scripts used |
| Screenshots | Prove it works (e.g., browser showing "Hello from Docker") |
| Commands used | Document key docker, gradle, or az commands |

Example: task4-docker-webserver/README.md

```
# Task 4: Web Server using Docker

Completed on: 2025-04-05
Objective: Deploy a web server inside a Docker container.

## What I Did
- Created a custom `index.html` page.
- Built a Docker image using Nginx.
- Ran the container and accessed it at `http://localhost:8080`.

## Files
- `Dockerfile`: Defines the image with Nginx and custom HTML.
```

```
- `index.html`: Custom homepage.
- `screenshots/running.png`: Proof of running container.
## ► How to Run
1. Build the image:
   ```bash
 docker build -t my-webserver .
```

1. Run the container:

```
docker run -d -p 8080:80 --name web-container my-webserver
```

2. Open in browser:

```
http://localhost:8080
```

#### Screenshot

Docker Web Server Running Docker Web Server Running

#### What I Learned

- Container lifecycle: create, run, stop, remove.
- Importance of CMD ["nginx", "-g", "daemon off;"].
- Debugging with docker logs and docker exec.
- Best practices: health checks, non-root users, small images.

```
Main `README.md` (Root of Repo)
###
```markdown
# CodeAlpha DevOps Internship
  Hands-on DevOps projects using Docker, Jenkins, Azure, and
Gradle.
##
    Tasks Completed
| Task | Title | Status | Link |
|-----|-----|
   Task 4 | Web Server using Docker | Done | [View](task4-
docker-webserver/) |
```

```
Task 3 | Java Application using Gradle | In Progress | [View]
(task3-java-gradle/) |
    Task 1 | CI/CD Pipeline using Azure | Not Started | ... |
    Task 2 | Jenkins Remoting | Not Started | ... |
##
     Skills Demonstrated
- Docker containerization

    CI/CD fundamentals

- Automation with Gradle & Jenkins

    Cloud deployment (coming soon)

     Repository Structure
Each task has its own folder with:
- Code/configs
- Step-by-step instructions
- Screenshots (where applicable)
##
     Learning Resources
- [Docker Docs](https://docs.docker.com/)
- [Azure Pipelines](https://learn.microsoft.com/en-us/azure/
devops/pipelines/)
- [Gradle User Guide](https://docs.gradle.org/)
> Maintained by: [Your Name]
> Email: you@example.com
> LinkedIn: [your-profile]
```

Pro Tip: Add Screenshots!

Even simple ones make a **huge difference**:

- Terminal showing docker ps
- Browser showing your "Hello from Docker" page
- docker logs output
- docker stats running

Save them in screenshots/ and link in README.

Example:

```
task4-docker-webserver/

— Dockerfile
— index.html
— README.md
— screenshots/
```

All Dockerfiles and configs committed

Final Checklist Before Pushing

Each task has a README.md
Screenshots added (optional but recommended)
Main README.md explains the whole repo
No secrets (like passwords, Azure keys) in code
.gitignore added (ignore node_modules, logs, etc.)

Example .gitignore:

Ignore IDE files
.vscode/
.idea/
*.swp

Ignore logs
*.log

Ignore OS files
.DS_Store
Thumbs.db

Bonus: Make It Stand Out

- Add a GitHub Profile Readme (e.g., https://github.com/ yourusername/yourusername)
- Use **GitHub Issues** to track your task progress
- Enable **GitHub Pages** to host a simple project site
- Add **badges** to your main README:

Final Thoughts

You **don't need a lot of code** — you need to show: - You understand the **concepts** - You can **document and reproduce** your work - You follow **best practices** - You're **learning and growing**

This kind of repo will **stand out** in your internship application.

When you're ready, let me know: - If you want me to generate the full folder structure - Or help you write the README.md files - Or move to **Task 3: Java + Gradle**

You're doing awesome — keep going!