

Exercise 3: Freestyle Job - Flask Application Deployment

1. Objective: Automate the deployment of a Flask application.

2. Steps:

o Create a Freestyle Job.

o Configure the job to:

- Pull the Flask application repository from GitHub.

- Set up a virtual environment using Python:

```
python -m venv venv
```

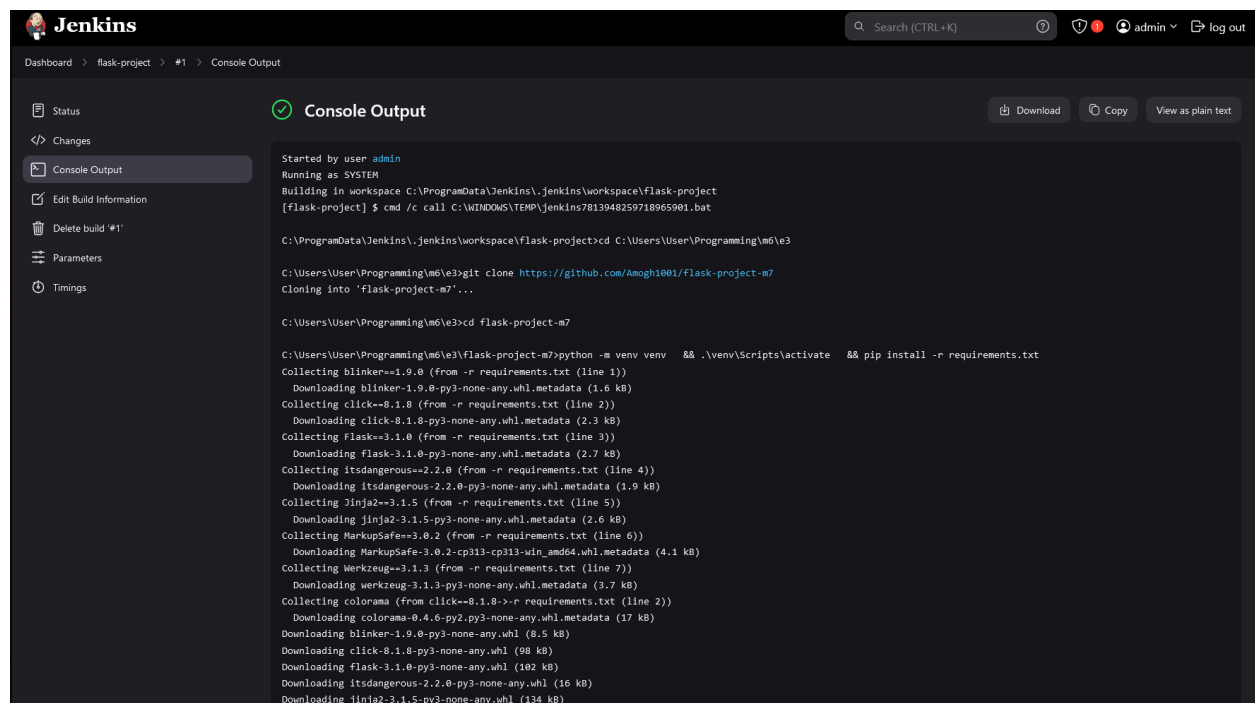
```
.\venv\Scripts\activate
```

```
pip install -r requirements.txt
```

- Start the Flask development server.

o Add a post-build action to verify the server is running (e.g., using curl or a similar tool to hit a test endpoint).

3. Task: Test the application deployment by accessing it via localhost.



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information (admin). The breadcrumb trail indicates the current view is 'Dashboard > flask-project > #1 > Console Output'. On the left sidebar, the 'Console Output' tab is selected. The main area displays the console output for a job run by user 'admin'. The output shows the following steps:

- Started by user admin
- Running as SYSTEM
- Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\flask-project
- Executing a shell command: `[flask-project] $ cmd /c call C:\WINDOWS\TEMP\Jenkins7813948259718965901.bat`
- Changing directory to `C:\ProgramData\Jenkins\jenkins\workspace\flask-project\cd C:\Users\User\Programming\m6\ve3`
- Cloning the repository: `C:\Users\User\Programming\m6\ve3>git clone https://github.com/Amogh1001/flask-project-m7`
- Cloning into 'flask-project-m7'...
- Changing directory to `C:\Users\User\Programming\m6\ve3>cd flask-project-m7`
- Setting up the virtual environment: `C:\Users\User\Programming\m6\ve3\flask-project-m7>python -m venv venv && .\venv\Scripts\activate && pip install -r requirements.txt`
- Installing dependencies from requirements.txt:

```
Collecting blinker==1.9.0 (from -r requirements.txt (line 1))
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click==8.1.8 (from -r requirements.txt (line 2))
  Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
Collecting flask==3.1.0 (from -r requirements.txt (line 3))
  Downloading flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting itsdangerous==2.2.0 (from -r requirements.txt (line 4))
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting Jinja2==3.1.5 (from -r requirements.txt (line 5))
  Downloading Jinja2-3.1.5-py3-none-any.whl.metadata (2.6 kB)
Collecting MarkupSafe==3.0.2 (from -r requirements.txt (line 6))
  Downloading MarkupSafe-3.0.2-cp311-cp311-win_amd64.whl.metadata (4.1 kB)
Collecting Werkzeug==3.1.3 (from -r requirements.txt (line 7))
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting colorama (from click==8.1.8->-r requirements.txt (line 2))
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Downloaded blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloaded click-8.1.8-py3-none-any.whl (98 kB)
Downloaded flask-3.1.0-py3-none-any.whl (102 kB)
Downloaded itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloaded Jinja2-3.1.5-py3-none-any.whl (134 kB)
```