NodeJS, Java, DotNET, Python Projects Compared

Project Type	Language	Build Tool	Output Artifact	Used For
✅ DotNET	C# (.NET)	dotnet CLI	.dll, .exe, .zip	Web App Deployment (Razor Pages)
✅ Java	Java	Maven (pom.xml)	.jar or .war	Spring Boot App (JAR/WAR deploy)
✅ NodeJS	JS/ESM	npm	source files + node_modules/	JavaScript Web App
✅ Python	Python	None or pip	source files only	Flask Web App

---

1. DotNET-Monitoring-Project (Razor Pages Web App)

🗂️ **Folder Breakdown:**

```
DotNET-Monitoring-Project/
├── src/                    # Source code
|   ├── dotnet-demoapp.csproj    # Project file
|   ├── Program.cs              # App entry point
|   ├── Pages/                  # Razor Pages (MVC-style)
|   ├── wwwroot/                # Static assets (CSS, JS, icons)
|   ├── appsettings.json        # App config
|   └── Properties/launchSettings.json  # Dev launch profile
├── tests/                  # xUnit Test Project
|   └── UnitTest.cs             # Unit test logic
```

🧱 **Build Tool:**

dotnet build

🎯 **Artifact(s):**

- ✅ dotnet-demoapp.dll (Intermediate Language)
- ✅ dotnet-demoapp.pdb, .deps.json, .runtimeconfig.json
- ✅ Publish output zipped to .zip for Nexus

📦 **Deployment:**

- Can be run via:

dotnet dotnet-demoapp.dll

- Or containerized via Dockerfile
- Or published as .exe for Windows

---

2. Java-Monitoring-Project (Spring Boot App)

🗂️ **Folder Breakdown:**

```
Java-Monitoring-Project/
├── pom.xml                 # Maven build file
├── src/
|   ├── main/java/demo/app/    # Main app logic & controllers
|   ├── main/resources/        # Static files, templates (Freemarker)
|   └── test/java/demo/app/    # JUnit Tests
```

🧱 **Build Tool:**

mvn clean package

🎯 **Artifact(s):**

- ✅ java-monitoring-app.jar or .war (for deployment)
- Includes everything: code + templates + static content

📦 **Deployment:**

- Via:

java -jar target/java-monitoring-app.jar

- Or deployed to app servers (e.g., Tomcat if .war)
- Easy to upload .jar to Nexus Maven Repo

---

3. NodeJS-Monitoring-Project (Express ESM Web App)

🗂️ **Folder Breakdown:**

```
NodeJS-Monitoring-Project/
├── src/
|   ├── public/             # Static content (icons, CSS, JS)
|   ├── routes/             # Express route handlers
|   ├── views/              # EJS Templates
|   ├── server.mjs          # App entry point
|   └── tests/              # REST API tests (.http)
├── package.json            # App metadata & dependencies
├── package-lock.json       # Locked versions
```

🧱 **Build Tool:**

npm install

🎯 **Artifact(s):**

- ⚠️ NodeJS doesn't "build" in classic sense
- Source + node_modules/ + package.json = deployable

📦 **Deployment:**

node server.mjs

Or:

npm start

💡 **Production Note:**

- Artifacts are zipped .tar.gz including:
    - server.mjs
    - routes/
    - views/
    - package.json + node_modules/
- Nexus upload: .tar.gz to raw repo or npm proxy

---

4. Python-Monitoring-Project (Flask App)

🗂️ **Folder Breakdown:**

```
Python-Monitoring-Project/
├── src/app/
│   ├── apis.py, views.py      # Route logic
│   ├── static/                # CSS, JS, images
│   ├── templates/             # HTML templates (Jinja2)
│   └── tests/                 # pytest-based unit tests
├── requirements.txt           # Python dependency list
├── run.py                     # App entry point
```

🧱 **Build Tool:**

pip install -r requirements.txt

🎯 **Artifact(s):**

- ✅ Raw source code zipped or containerized
- ⚠️ No binary artifact like .jar or .dll

- .whl or .tar.gz can be created for Python packages (if modularized)

📦 **Deployment:**

python run.py

Or via:

gunicorn app:app

Nexus:

- Upload project.tar.gz to raw repo
- Or host .whl files in Nexus Python repo (for libraries)

🧠 **Ultra Summary Table (DevOps Artifact View)**

Language	Build Tool	Entry Point	Final Artifact	Upload to Nexus As	Deploy With
.NET	dotnet	Program.cs	.dll, .zip, optional .exe	.zip to raw, or .nupkg to NuGet	dotnet, Docker, systemd
Java	mvn	JavaDemoapp.java	.jar or .war	.jar/.war to Maven repo	java -jar, Tomcat, Docker
NodeJS	npm	server.mjs	No artifact (source + deps)	.tar.gz to raw or npm repo	node, PM2, Docker
Python	pip	run.py	No binary artifact	.tar.gz or .whl to raw/PyPi	python, gunicorn, Docker

---

🚀 **Recommendations for Publishing to Nexus**

Tech	Nexus Format	Artifact Type
.NET	Raw / NuGet	dotnet publish → .zip or .nupkg
Java	Maven	mvn package → .jar/.war
NodeJS	Raw / npm	tar.gz (source + deps)
Python	Raw / PyPi	tar.gz or .whl