

Module : 06

Jenkins Exercise

Name : Anubhav Ranjan

Registration No : 774

College : IIIT kalyani

Roll No : ECE/21114

Exercise 2: Freestyle Job - Build and Deploy a JavaScript Project

1. Objective: Build and deploy a JavaScript-based web project.

2. Steps:

- o Create a Freestyle Job.
- o Configure the job to:
 - Clone a JavaScript repository (e.g., a React project) from GitHub.
 - Run npm install to install dependencies.
 - Build the project using npm run build.
 - Copy the build folder to a deployment directory using a Windows batch command or PowerShell.
- o Archive the build artifacts for future reference.

3. Task: Verify the deployment by accessing the web application from the deployment directory.

Let's complete this exercise step by step to build and deploy a JavaScript-based web project using Jenkins.

Step 1: Prerequisites

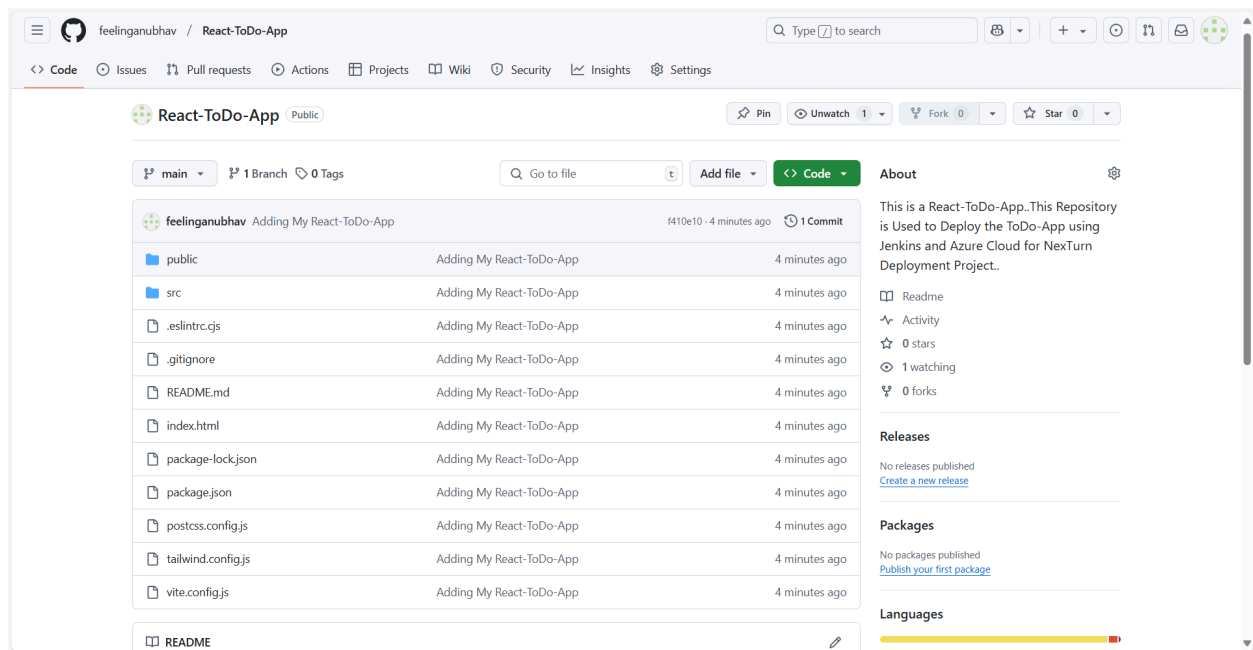
Before proceeding, ensure the following:

1. **Jenkins is installed and running.**
2. **Node.js and npm are installed** on the Jenkins server. Verify by running:

```
node -v & npm -v
```

```
(base) PS D:\jenkins_New_Home> npm -v
10.8.2
(base) PS D:\jenkins_New_Home> node -v
v20.17.0
```

3. A **JavaScript project repository** (e.g., a React app) is available on GitHub.

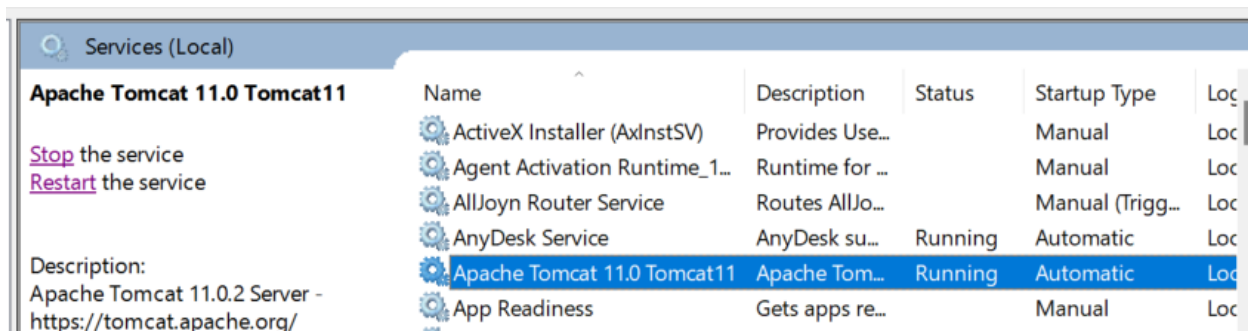


4. The server has access to the deployment directory.

Step 2: Create a Freestyle Job in Jenkins

1. Open Jenkins:

- Start the Apache Tomcat Server from Services on your Windows



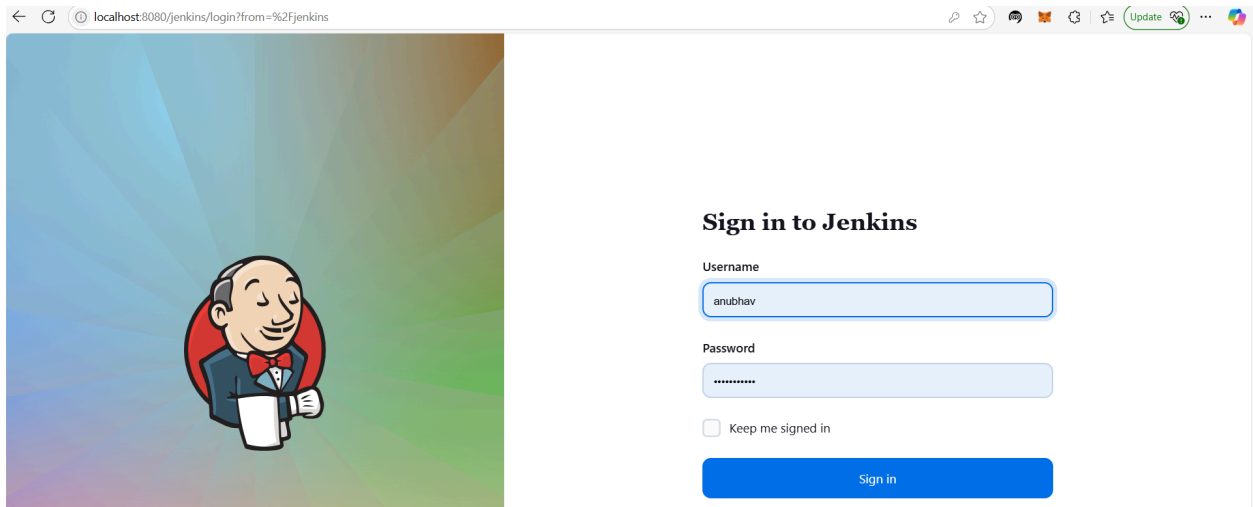
- Or Open the Terminal..Navigate to the directory where jenkins.war file is present and Run the command:

```
java -jar .\jenkins.war
```

```
(base) PS D:\packages> java -jar .\jenkins.war
Running from: D:\packages\jenkins.war
webroot: D:\jenkins_New_Home\war
2025-01-25 04:21:33.952+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2025-01-25 04:21:34.081+0000 [id=1] WARNING o.e.j.ee9.nested.ContextHandler#setContextPath: Empty contextPath
2025-01-25 04:21:34.193+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-12.0.13; built: 2024-09-0
3T03:04:05.240Z; git: 816018a420329c1cacd4116799cda8c8c60a57cd; jvm 21.0.5+9-LTS-239
2025-01-25 04:21:35.110+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /
, did not find org.eclipse.jetty.ee9.jsp.JettyJspServlet
2025-01-25 04:21:35.178+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2025-01-25 04:21:35.854+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: D:\jenkins
_New_Home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
```

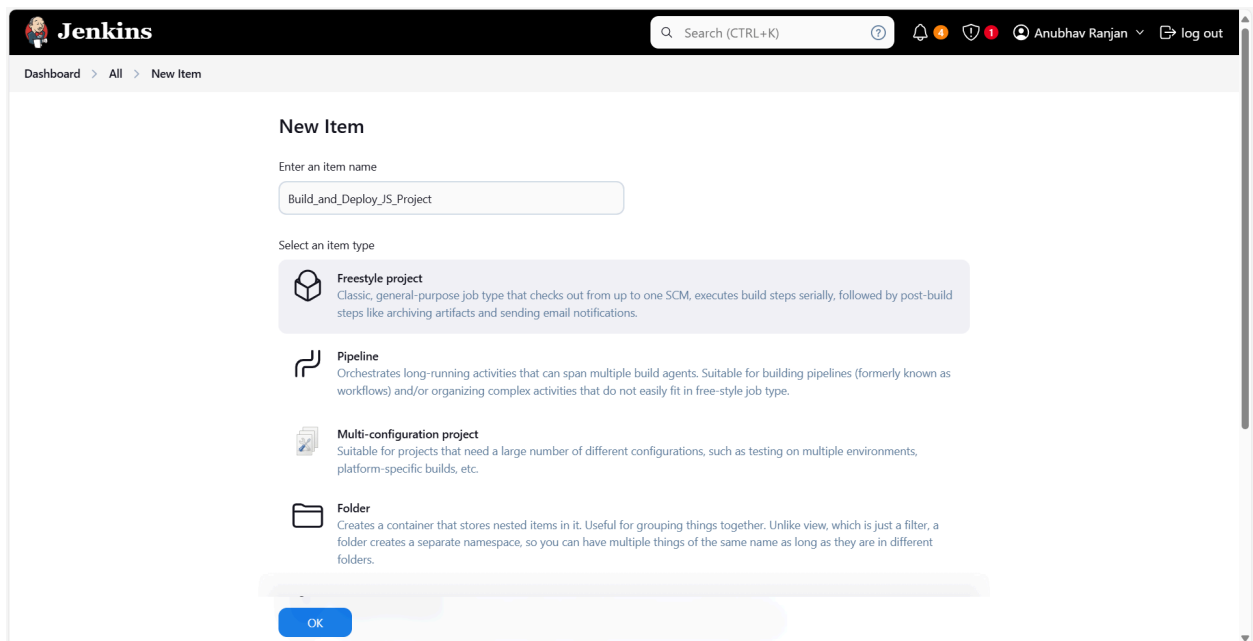
```
2025-01-25 04:21:36.304+0000 [id=47] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2025-01-25 04:21:36.504+0000 [id=56] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2025-01-25 04:21:40.459+0000 [id=54] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2025-01-25 04:21:40.482+0000 [id=46] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2025-01-25 04:21:40.485+0000 [id=49] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2025-01-25 04:21:40.847+0000 [id=47] INFO h.p.b.g.GlobalTimeoutConfiguration#load: global timeout not set
2025-01-25 04:21:41.824+0000 [id=56] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2025-01-25 04:21:41.828+0000 [id=47] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-01-25 04:21:41.883+0000 [id=56] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-01-25 04:21:41.897+0000 [id=51] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updat
ed
2025-01-25 04:21:41.948+0000 [id=57] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-25 04:21:41.993+0000 [id=39] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
```

- Launch Jenkins in your browser (usually <http://localhost:8080> or <http://localhost:8080/jenkins>).
- Log in with your credentials.



2. Create a New Freestyle Job:

- Click on “**New Item**” from the Jenkins dashboard.
- Enter a name for the job, i.e.
Build_and_Deploy_JS_Project
- Select “**Freestyle project**” and click **OK**.



3. Add a Description :

- On the job configuration page, add a brief description such as:

Description

This Jenkins job automates the process of building and deploying a JavaScript-based web application. It performs the following steps:

Clones the repository: Retrieves the source code of the JavaScript project (e.g., React, Vue.js) from a GitHub repository.

Installs dependencies: Runs npm install to download and set up all required project dependencies.

Builds the project: Executes npm run build to generate optimized production-ready files in the build directory.

Deploys the project: Copies the contents of the build directory to a specified deployment path on the server using xcopy.

Archives the build artifacts: Saves the build files in Jenkins for future reference or redeployment.

Verification:

After deployment, the application can be accessed from the deployment directory via a simple HTTP server or hosted environment.

Step 3: Configure the Job

Step 3.1: Clone the GitHub Repository

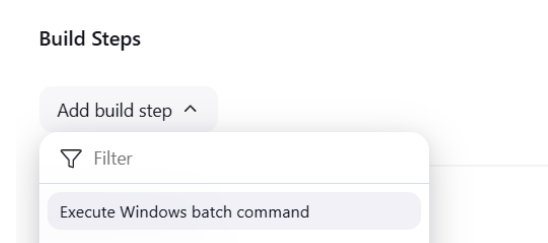
1. In the job configuration page, go to the **Source Code Management** section.
2. Select **Git** and enter the repository URL (i.e., <https://github.com/feelinganubhav/React-ToDo-App.git>).
3. (Optional) If the repository is private, add your credentials.

The screenshot shows the Jenkins job configuration page for a job named 'Build_and_Deploy_JS_Project'. The 'Configuration' tab is selected. On the left, the 'Configure' sidebar shows 'Source Code Management' as the active section. The main area is titled 'Source Code Management' and has two radio buttons: 'None' and 'Git'. The 'Git' option is selected. Below this, there is a 'Repositories' section with a dashed border. It contains a 'Repository URL' field with the value 'https://github.com/feelinganubhav/React-ToDo-App.git'. Below the URL field is a 'Credentials' dropdown menu currently set to '- none -'. There are '+ Add' and 'Advanced' buttons below the credentials. At the bottom of the 'Repositories' section is an 'Add Repository' button.

Step 3.2: Install Dependencies

1. Scroll to the “Build” Section:

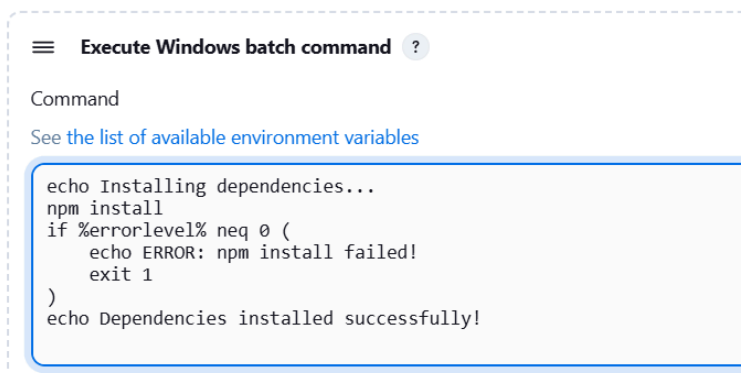
- Click “Add Build Step” → Select “Execute Windows batch command”.



2. Add the following commands to install dependencies:

```
echo Installing dependencies...  
  
npm install  
  
if %errorlevel% neq 0 (  
    echo ERROR: npm install failed!  
    exit 1  
)  
  
echo Dependencies installed successfully!
```

Build Steps



Step 3.3: Build the Project

1. Click **Add build step** > **Execute Windows batch command**.
2. Add the following commands to build the project:

```
echo Building the project...

npm run build

if %errorlevel% neq 0 (

    echo ERROR: Build failed!

    exit 1

)

echo Build completed successfully!
```

≡ **Execute Windows batch command** ?

Command

See [the list of available environment variables](#)

```
echo Building the project...
npm run build
if %errorlevel% neq 0 (
    echo ERROR: Build failed!
    exit 1
)
echo Build completed successfully!
|
```

Advanced ▾

Step 3.4: Deploy the Project

1. Click **Add build step** > **Execute Windows batch command**.
2. Add the following commands to copy the build folder to the deployment directory:

```
echo Deploying the build...
```

```
xcopy "%WORKSPACE%\build" "D:\React-ToDo-App-Deployment" /E /I /H /Y
```

```
if %errorlevel% neq 0 (
```

```
    echo ERROR: Deployment failed!
```

```
    exit 1
```

```
)
```

```
echo Deployment completed successfully!
```

≡ Execute Windows batch command ?

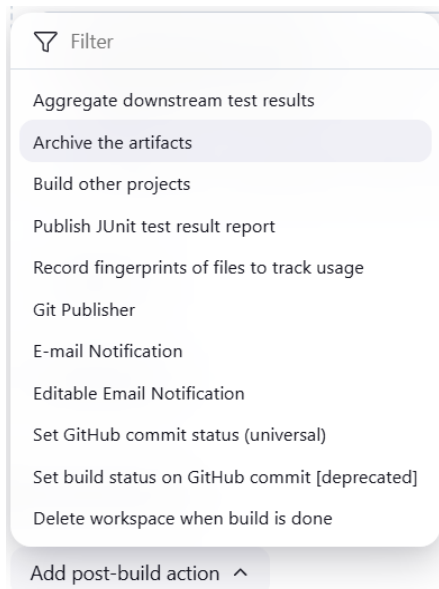
Command

See [the list of available environment variables](#)

```
echo Deploying the build...
xcopy "%WORKSPACE%\build" "D:\React-ToDo-App-Deployment" /E /I /H /Y
if %errorlevel% neq 0 (
    echo ERROR: Deployment failed!
    exit 1
)
echo Deployment completed successfully!
```

Step 3.5: Archive Build Artifacts

1. Go to the **Post-build Actions** section.
2. Click **Add post-build action** > **Archive the artifacts**.



3. In the **Files to archive** field, enter:

`build/**`

Post-build Actions



4. **Save Your Job:**

- Scroll to the bottom and click “Save”.

Step 5: Run and Test the Job

1. Run the Job:

- Go to the job's dashboard.
- Click “Build Now” to execute the file transfer task.

2. Verify the Output:

- Check the Console Output in Jenkins for the success message:

✓ Console Output

[Download](#)[Copy](#)[View as plain text](#)

```
Started by user Anubhav Ranjan
Running as SYSTEM
Building in workspace D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/feelinganubhav/React-ToDo-App.git # timeout=10
Fetching upstream changes from https://github.com/feelinganubhav/React-ToDo-App.git
> git.exe --version # timeout=10
> git --version # 'git version 2.38.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/feelinganubhav/React-ToDo-App.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision f410e106c1ab7f219fbd0c70b9c812d9e0549c04 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f410e106c1ab7f219fbd0c70b9c812d9e0549c04 # timeout=10
Commit message: "Adding My React-ToDo-App"
> git.exe rev-list --no-walk f410e106c1ab7f219fbd0c70b9c812d9e0549c04 # timeout=10
[Build_and_Deploy_JS_Project] $ cmd /c call C:\Users\ANUBHA~1\AppData\Local\Temp\jenkins10806346563219259011.bat

D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>echo Installing dependencies...
Installing dependencies...

D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>npm install

up to date, audited 363 packages in 5s
```

```
125 packages are looking for funding
  run `npm fund` for details

5 vulnerabilities (3 moderate, 2 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
[Build_and_Deploy_JS_Project] $ cmd /c call C:\Users\ANUBHA~1\AppData\Local\Temp\jenkins5840406530483943286.bat

D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>echo Building the project...
Building the project...

D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>npm run build

> @9todo@0.0.0 build
> vite build

[36mvite v5.3.4 [32mbuilding for production...[36m[39m
transforming...
Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
[32m[39m 36 modules transformed.
rendering chunks...
computing gzip size...
[2mdist/[22m[32mindex.html [39m[1m[2m 0.47 kB[22m[1m[22m[2m | gzip: 0.30 kB[22m
[2mdist/[22m[35massets/index-qX10TC9a.css [39m[1m[2m 7.95 kB[22m[1m[22m[2m | gzip: 2.28 kB[22m
[2mdist/[22m[36massets/index-BehmD-c_.js [39m[1m[2m146.16 kB[22m[1m[22m[2m | gzip: 47.08 kB[22m
[32mbuilt in 1.86s[39m
[Build and Deploy JS Project] $ cmd /c call C:\Users\ANUBHA~1\AppData\Local\Temp\jenkins11686296670645456913.bat
```

```
[Build_and_Deploy_JS_Project] $ cmd /c call C:\Users\ANUBHA~1\AppData\Local\Temp\jenkins11686296670645456913.bat

D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>echo Deploying the build...
Deploying the build...

D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>xcopy "D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project\dist" "D:\React-
ToDo-App-Deployment" /E /I /H /Y
D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project\dist\index.html
D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project\dist\vite.svg
D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project\dist\assets\index-BehmD-c_.js
D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project\dist\assets\index-qX10TC9a.css
4 File(s) copied

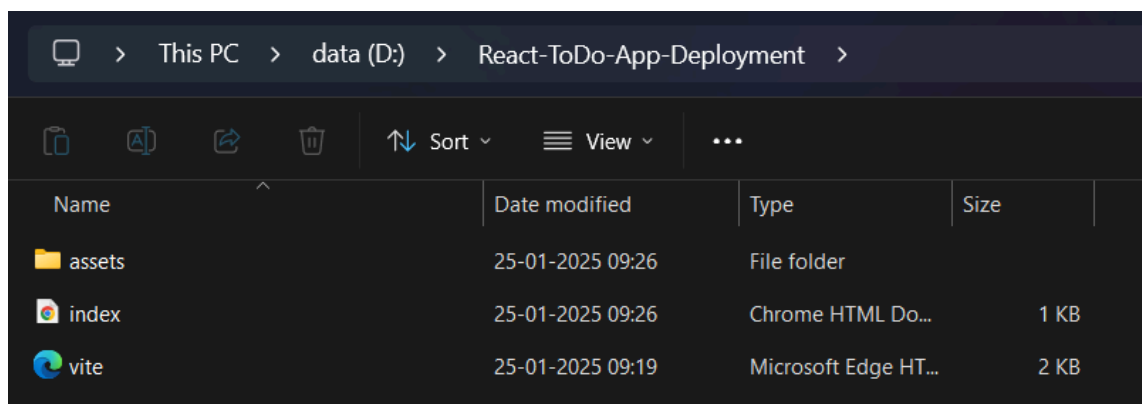
D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>if 0 NEQ 0 (
echo ERROR: Deployment failed!
exit 1
)

D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>echo Deployment completed successfully!
Deployment completed successfully!

D:\jenkins_New_Home\workspace\Build_and_Deploy_JS_Project>exit 0
Archiving artifacts
Finished: SUCCESS
```

Step 6: Verify the Deployment

1. Navigate to the deployment directory:
"D:\React-ToDo-App-Deployment"
2. Check if the React-ToDo-App-Deployment folder contains the files.



3. Host the React-ToDo-App-Deployment folder using a simple HTTP server :

npx serve -s React-ToDo-App-Deployment

```
(base) PS D:\> npx serve -s .\React-ToDo-App-Deployment\  
Need to install the following packages:  
serve@14.2.4  
Ok to proceed? (y) y
```

Serving!

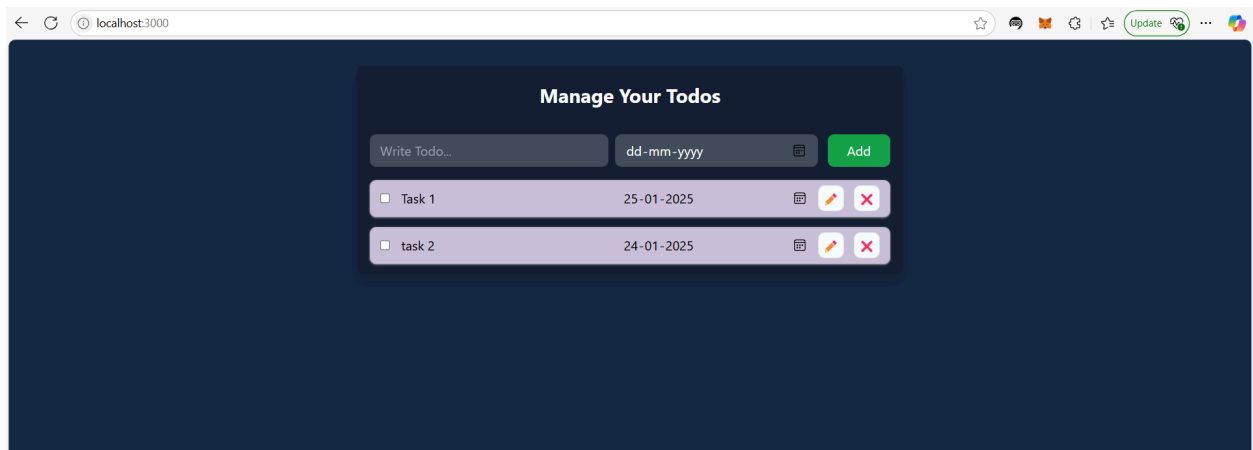
- Local: http://localhost:3000
- Network: http://192.168.51.56:3000

Copied local address to clipboard!

```
HTTP 25/1/2025 9:37:52 am ::1 GET /  
HTTP 25/1/2025 9:37:52 am ::1 Returned 200 in 136 ms  
HTTP 25/1/2025 9:37:52 am ::1 GET /assets/index-BehmD-c_.js  
HTTP 25/1/2025 9:37:52 am ::1 GET /assets/index-qX10TC9a.css  
HTTP 25/1/2025 9:37:52 am ::1 Returned 200 in 24 ms  
HTTP 25/1/2025 9:37:52 am ::1 Returned 200 in 129 ms  
HTTP 25/1/2025 9:37:52 am ::1 GET /vite.svg  
HTTP 25/1/2025 9:37:52 am ::1 Returned 200 in 15 ms  
HTTP 25/1/2025 9:37:52 am ::1 GET /vite.svg  
HTTP 25/1/2025 9:37:52 am ::1 Returned 304 in 1 ms
```

4. Access the web application in your browser at:

<http://<server-ip>:<port>>



Git Build Data

Revision: f410e106c1ab7f219fbd0c70b9c812d9e0549c04

Repository: <https://github.com/feelinganubhav/React-ToDo-App.git>

- refs/remotes/origin/main

Built Branches

- refs/remotes/origin/main: Build #11 of Revision f410e106c1ab7f219fbd0c70b9c812d9e0549c04 (refs/remotes/orig

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

Rename

Builds

Filter

Today

✓ #11 9:26 AM

✓ Build_and_Deploy_JS_Project

This Jenkins job automates the process of building and deploying a JavaScript-based web application. It performs the following steps:

Clones the repository: Retrieves the source code of the JavaScript project (e.g., React, Vue.js) from a GitHub repository.

Installs dependencies: Runs npm install to download and set up all required project dependencies.

Builds the project: Executes npm run build to generate optimized production-ready files in the build directory.

Deploys the project: Copies the contents of the build directory to a specified deployment path on the server using xcopy.

Archives the build artifacts: Saves the build files in Jenkins for future reference or redeployment.

Verification:

After deployment, the application can be accessed from the deployment directory via a simple HTTP server or hosted environment.

Last Successful Artifacts

index-BehmD-c_js	142.74 KiB	view
index-qX10TC9a.css	7.76 KiB	view
index.html	472 B	view
vite.svg	1.46 KiB	view

Permalinks

Artifacts are Archived Sucessfully..