Application Containerization Steps:

For this assignment, I containerized a static Next.js application using Docker. First, I ensured the application was built successfully locally with npm run build, which generates static files in the /out directory. I created a multi-stage Dockerfile to optimize image size and separate the build process from the final runtime environment.

In the first stage, I used node:22-alpine as the base image, installed dependencies using npm install --force, and built the app. In the second stage, I used the nginx:stable-alpine image to serve the static output. The compiled contents from /app/out/ were copied to the default nginx folder /usr/share/nginx/html/.

Volumes and Bind Mounts:

Though persistent storage is not critical for a static site, I used a bind mount during development to reflect local file changes without rebuilding the image every time. This was done using the docker run -v option (e.g., -v $(pwd)/out:/usr/share/nginx/html) to map the local build output to the nginx content folder. This helped test frontend updates quickly during development.

Challenges and Resolutions:

One challenge was to ensure the build output was compatible with nginx's static file serving structure. I initially used the Next.js server directly but switched to **npm run build** and **next export** to generate a static version. Another challenge was minimizing image size and avoiding unnecessary files, which I addressed by using a multi-stage build and .dockerignore to exclude node_modules.

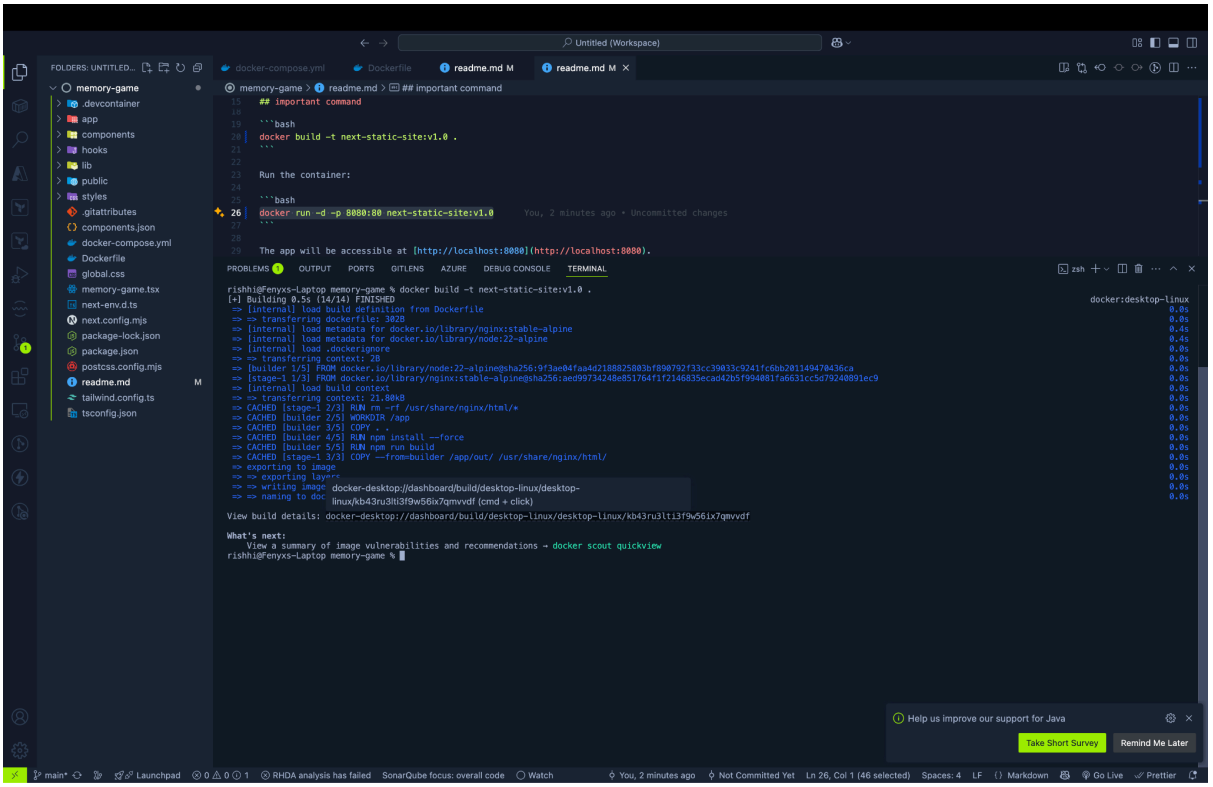
important command

Build the Docker image:

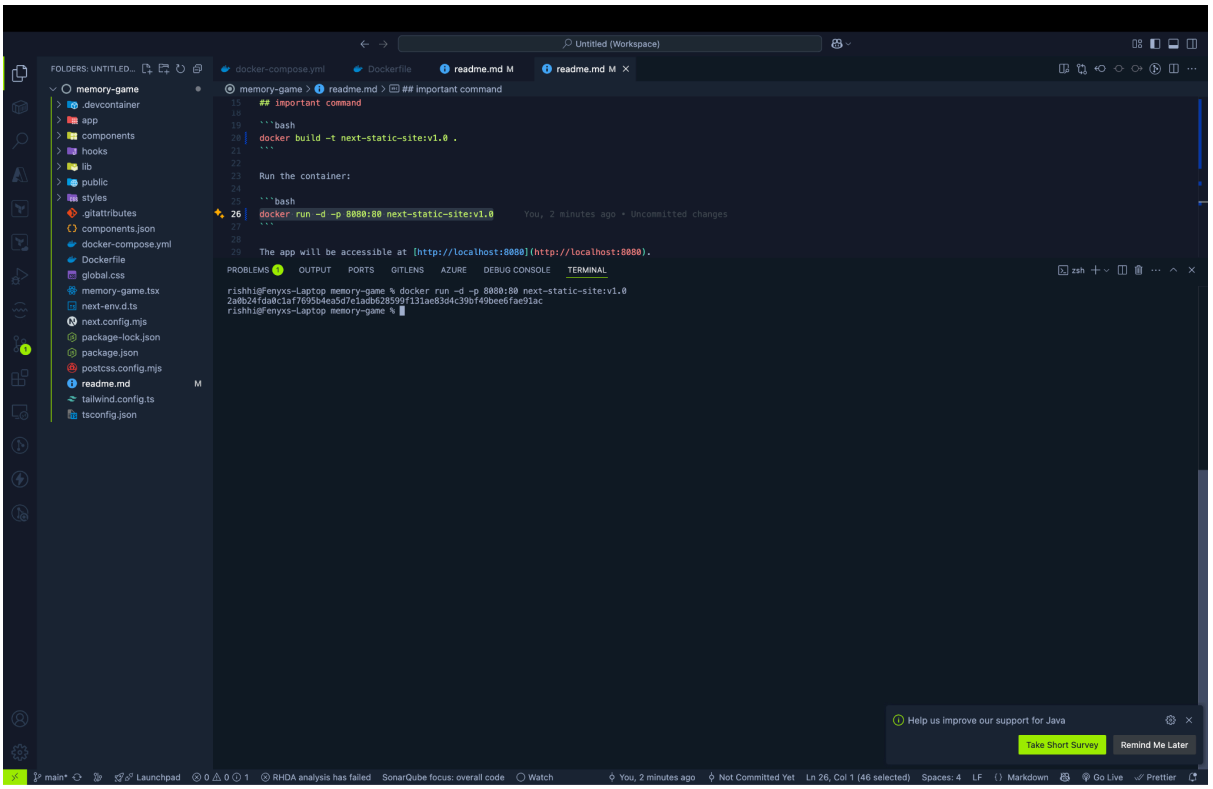
```
docker build -t next-static-site:v1.0 .
```

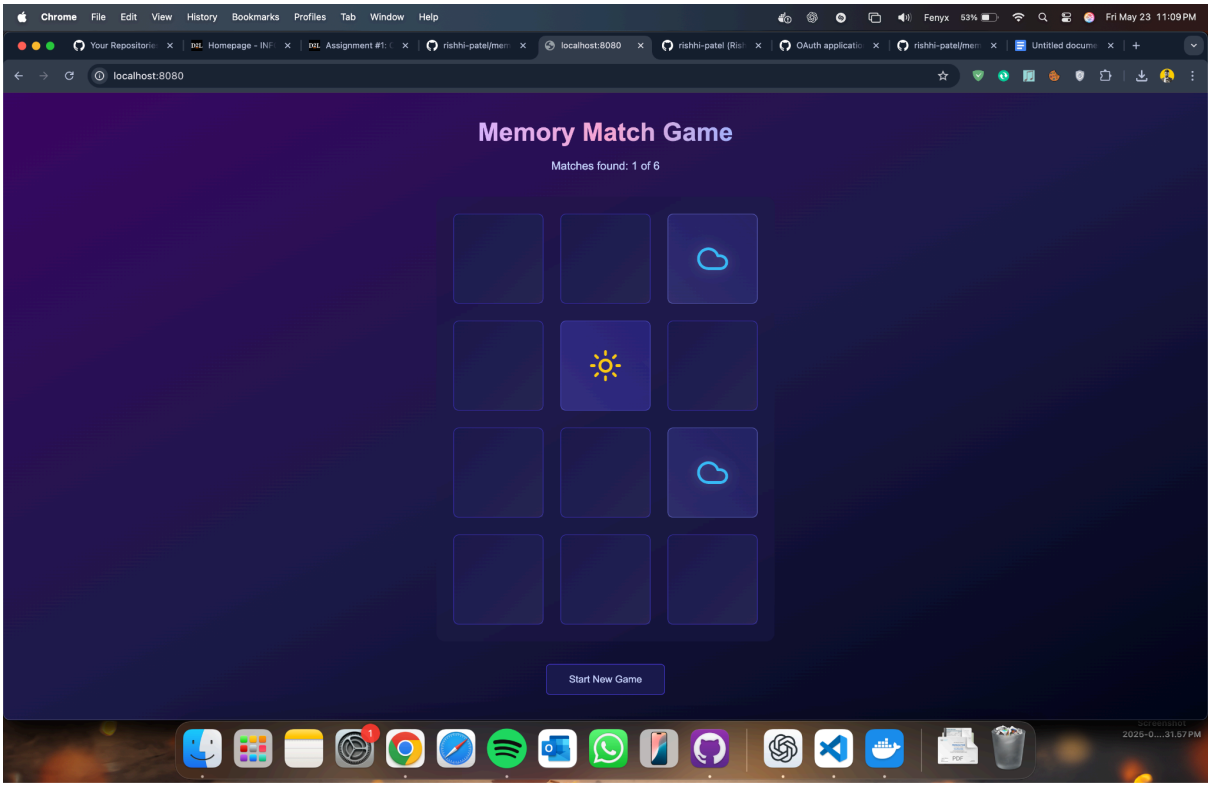
Run the container:

```
docker run -d -p 8080:80 next-static-site:v1.0
```


Untitled (Workspace)
memory-game
readme.md M
important command
docker build -t next-static-site:v1.0 .
Run the container:
docker run -d -p 8080:80 next-static-site:v1.0
You, 2 minutes ago • Uncommitted changes
The app will be accessible at http://localhost:8080.
PROBLEMS OUTPUT PORTS GITLENS AZURE DEBUG CONSOLE TERMINAL
rishhi@Fenyxs-Laptop memory-game % docker build -t next-static-site:v1.0 .
[+] Building 0.5s (14/14) FINISHED
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/kb43ru3lti3f9w56ix7qmvvdf
What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview
rishhi@Fenyxs-Laptop memory-game %
Help us improve our support for Java
Take Short Survey
Remind Me Later

Run command :


Untitled (Workspace)
memory-game
readme.md M
important command
docker build -t next-static-site:v1.0 .
Run the container:
docker run -d -p 8080:80 next-static-site:v1.0
You, 2 minutes ago • Uncommitted changes
The app will be accessible at http://localhost:8080.
PROBLEMS OUTPUT PORTS GITLENS AZURE DEBUG CONSOLE TERMINAL
rishhi@Fenyxs-Laptop memory-game % docker run -d -p 8080:80 next-static-site:v1.0
2a0b24fda0c1af7695b4ea5d7e1adb628599f131ae83d4c39bf49bee6fae91ac
rishhi@Fenyxs-Laptop memory-game %
Help us improve our support for Java
Take Short Survey
Remind Me Later


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Fri May 23 11:09 PM
localhost:8080
Memory Match Game
Matches found: 1 of 6
Start New Game