



National Textile University

Department of Computer Science

Subject:

Operating system

Submitted To:

Sir Nasir

Submitted By:

Noor ul Ain

Registration No:

23-NTU-CS-1221

Lab no:

13

Semester:

5th

Directory structure:

The screenshot shows the Visual Studio Code interface with a Docker extension. On the left, the Explorer sidebar shows a project structure with a 'public' folder containing 'index.html'. The 'Dockerfile' is also visible. In the center, the code editor displays the 'index.html' file, which contains basic HTML structure. At the bottom, the terminal shows commands related to Docker, and the status bar indicates the Docker extension is active.

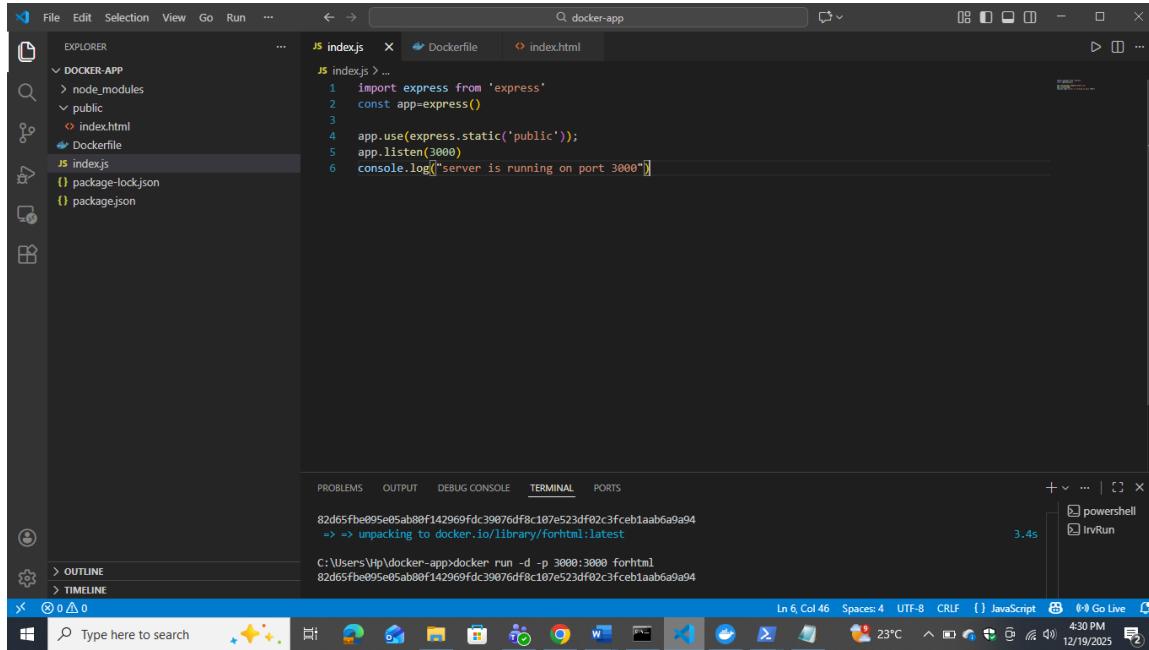
Docker file:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows a project structure for "DOCKER-APP" containing "node_modules", "public", and "index.html". The "Dockerfile" is selected.
- Code Editor:** The main area displays the "Dockerfile" content:

```
FROM node:lts-alpine3.23
COPY index.js .
COPY package.json .
COPY public ./public
RUN npm i
CMD ["node","index.js"]
```
- Terminal:** At the bottom, the terminal shows the command being run: `82d65fbe095e05ab88f142969fdc39076df8c107e523df02c3fce1aab6a9a94 => unpacking to docker.io/library/forhtml:latest`. It also shows the path: `C:\Users\HP\docker-app\docker run -d -p 3000:3000 forhtml`.
- Status Bar:** The status bar at the bottom right shows the date and time as "12/19/2025 4:29 PM".

Javascript file:

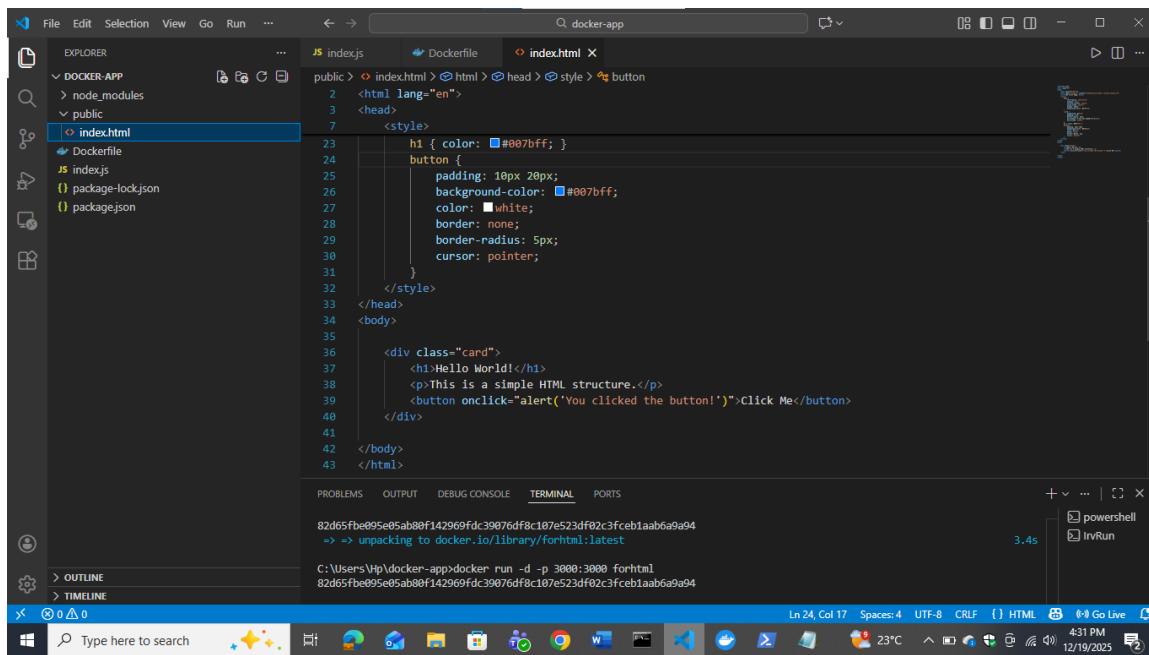


A screenshot of the Visual Studio Code interface. The title bar says "File Edit Selection View Go Run ... docker-app". The Explorer sidebar shows a folder named "DOCKER-APP" containing "node_modules", "public", "Dockerfile", "index.js", "package-lock.json", and "package.json". The "index.js" tab is active, displaying the following code:

```
1 import express from 'express'
2 const app=express()
3
4 app.use(express.static('public'))
5 app.listen(3000)
6 console.log("server is running on port 3000")
```

The bottom status bar shows "Ln 6, Col 46 Spaces: 4 UTF-8 CRLF {} JavaScript 4:30 PM 23°C 12/19/2025".

Html file:



A screenshot of the Visual Studio Code interface. The title bar says "File Edit Selection View Go Run ... docker-app". The Explorer sidebar shows a folder named "DOCKER-APP" containing "node_modules", "public", and "index.html". The "index.html" tab is active, displaying the following HTML code:

```
1 <html lang="en">
2   <head>
3     <style>
4       h1 { color: #007bff; }
5       button {
6         padding: 10px 20px;
7         background-color: #007bff;
8         color: white;
9         border: none;
10        border-radius: 5px;
11        cursor: pointer;
12      }
13    </style>
14  </head>
15  <body>
16    <div class="card">
17      <h1>Hello World!</h1>
18      <p>This is a simple HTML structure.</p>
19      <button onclick="alert('You clicked the button!')">Click Me</button>
20    </div>
21  </body>
22 </html>
```

The bottom status bar shows "Ln 24, Col 17 Spaces: 4 UTF-8 CRLF {} HTML 4:31 PM 23°C 12/19/2025".

Image creation:

```
C:\Users\Hp\docker-app>docker build -t forhtml .
[+] Building 28.2s (10/10) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 159B
  => [internal] load metadata for docker.io/library/node:lts-alpine3.23
  => [internal] load .dockerrcignore
  => => transferring context: 2B
  => CACHED [1/5] FROM docker.io/library/node:lts-alpine3.23@sha256:c921b97d4b74f51744057454b306b418cf693865e73b810055918960
  => => resolve docker.io/library/node:lts-alpine3.23@sha256:c921b97d4b74f51744057454b306b418cf693865e73b810055918960
  => [internal] load build context
  => => transferring context: 1.43kB
  => [2/5] COPY index.js .
  => [3/5] COPY package.json .
  => [4/5] COPY public ./public
  => [5/5] RUN npm i
  => exporting to image
  => => exporting layers
  => => exporting manifest sha256:ee6babd1335dad0a13ba5048688a47c6e62606bc7bc2171358dd32b30eeee9d51
  => => exporting config sha256:d548e72ca7ee1becff9340a2affdb3579eb0e02dec20771de5a3a9938998d8a5
  => => exporting attestation manifest sha256:73db44e25d06e0f99a786a5c289057606052adf206e73ecd92564558de0ccb20
  => => exporting manifest list sha256:41f6f794e96cb202a900b8efd67f6f86cc491fd466f949c3e367af5b91f39d37
  => => naming to docker.io/library/forhtml:latest
  => [5/5] RUN npm i
  => exporting to image
  => => exporting layers
  => => exporting manifest sha256:ee6babd1335dad0a13ba5048688a47c6e62606bc7bc2171358dd32b30eeee9d51
  => => exporting config sha256:d548e72ca7ee1becff9340a2affdb3579eb0e02dec20771de5a3a9938998d8a5
  => => exporting attestation manifest sha256:73db44e25d06e0f99a786a5c289057606052adf206e73ecd92564558de0ccb20
```

```
C:\Users\Hp\docker-app>docker run -d -p 3000:3000 forhtml
82d65fbe095e05ab80f142969fdc39076df8c107e523df02c3fcceb1aab6a9a94

C:\Users\Hp\docker-app>[]
```

Output:

