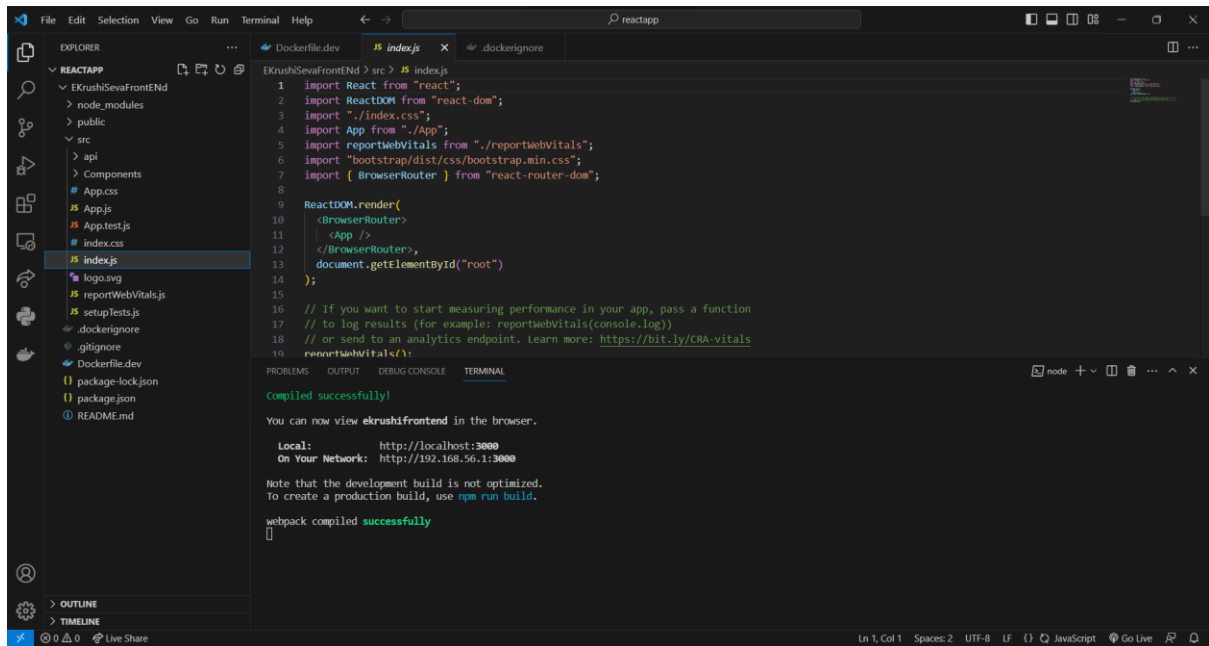


Name : Madhavi Suratkar

Topic : Project deployment using kubernetes

Date : 28-04-2023



The screenshot shows the Visual Studio Code editor with a project named 'reactapp'. The Explorer panel on the left shows the file structure, with 'index.js' selected under the 'src' directory. The main editor displays the content of 'index.js', which is a React application using ReactDOM and BrowserRouter. The terminal at the bottom shows the output of running 'npm run build', indicating a successful webpack build and providing local and network URLs to view the application in a browser.

```
1 import React from "react";
2 import ReactDOM from "react-dom";
3 import "../index.css";
4 import App from "./App";
5 import reportWebVitals from "../reportWebVitals";
6 import "bootstrap/dist/css/bootstrap.min.css";
7 import { BrowserRouter } from "react-router-dom";
8
9 ReactDOM.render(
10   <BrowserRouter>
11     <App />
12   </BrowserRouter>,
13   document.getElementById("root")
14 );
15
16 // If you want to start measuring performance in your app, pass a function
17 // to log results (for example: reportWebVitals(console.log))
18 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
19 reportWebVitals</>
```

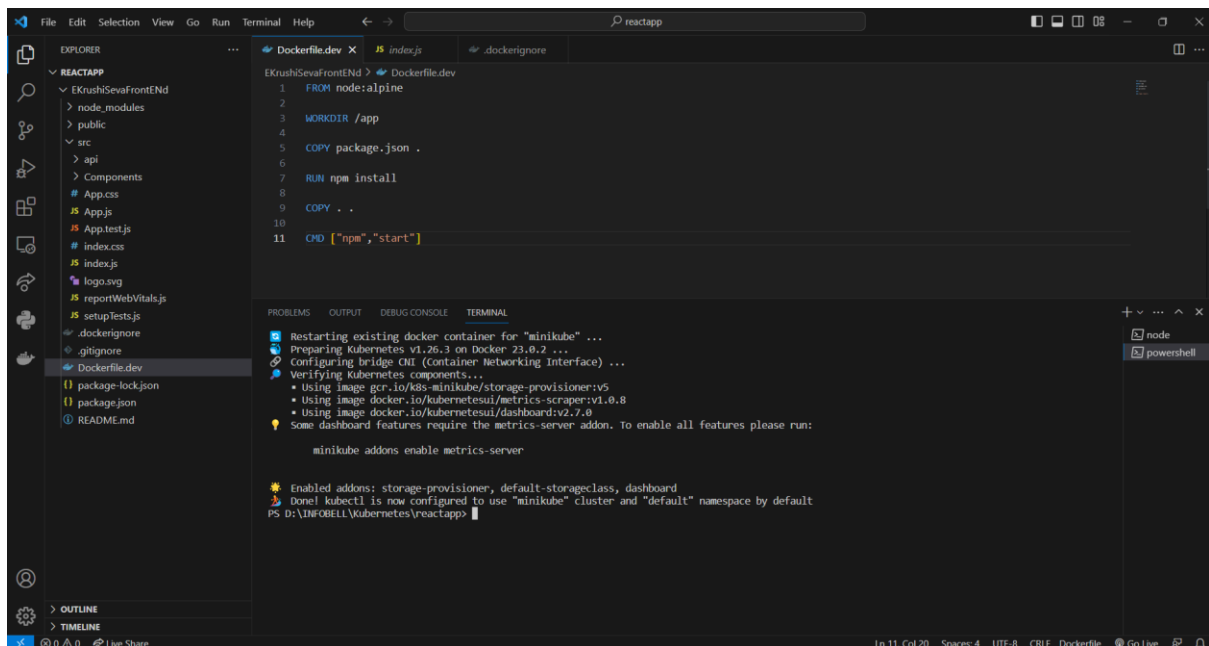
Compiled successfully!

You can now view **ekrushifrontend** in the browser.

Local: http://localhost:3000  
On Your Network: http://192.168.56.1:3000

Note that the development build is not optimized.  
To create a production build, use **npm run build**.

webpack compiled **successfully**



The screenshot shows the Visual Studio Code editor with the same project. The Explorer panel shows 'Dockerfile.dev' selected. The main editor displays the content of 'Dockerfile.dev', which is a Dockerfile for building and running the application in a container. The terminal at the bottom shows the output of running 'docker build -f Dockerfile.dev .', indicating a successful build and providing instructions on how to run the container.

```
1 FROM node:alpine
2
3 WORKDIR /app
4
5 COPY package.json .
6
7 RUN npm install
8
9 COPY . .
10
11 CMD ["npm", "start"]
```

Restarting existing docker container for "minikube" ...  
Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...  
Configuring bridge CNI (Container Networking Interface) ...  
Verifying Kubernetes components...  
• Using image gcr.io/k8s-minikube/storage-provisioner:v5  
• Using image docker.io/kubernetes/metrics-scraper:v1.0.8  
• Using image docker.io/kubernetes/dashboard:v2.7.0  
Some dashboard features require the metrics-server addon. To enable all features please run:  
  
minikube addons enable metrics-server

Enabled addons: storage-provisioner, default-storageclass, dashboard  
Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default  
PS D:\INFOBELL\kubernetes\reactapp>

