

```

version: '3.8'
services:
  newfrontend:
    build: ./RECKON_TP/backend
    ports:
      - 8080:8080

```

```

Dockerfile backend
1 # Use Node.js LTS version
2 FROM node:18
3
4 # Set working directory inside the container
5 WORKDIR /app
6
7 # Copy package.json and package-lock.json
8 COPY package*.json ./ 
9
10 # Install dependencies
11 RUN npm install
12
13 # Copy the rest of the app
14 COPY . .
15
16 # Expose port (make sure it's same as server uses)
17 EXPOSE 8080
18
19 # Start the server
20 CMD ["node", "server.js"]
21

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Compiled successfully!

You can now view [newfrontend](#) in the browser.

Local: <http://localhost:3000>
On Your Network: <http://192.168.137.28:3000>

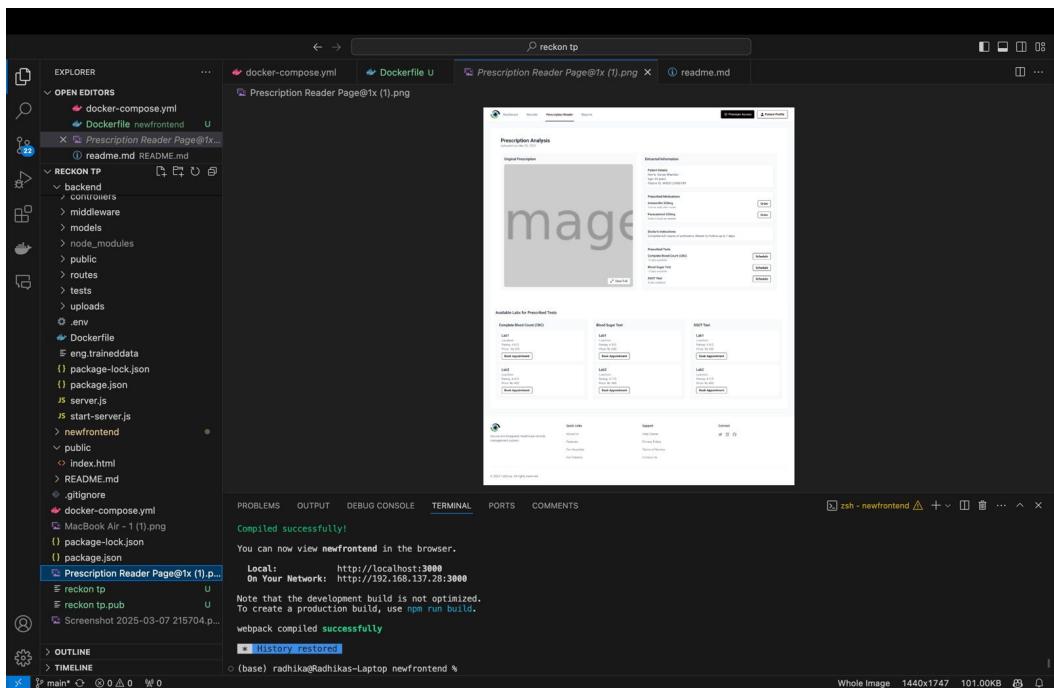
Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

* History restored

(base) radhika@Radhikas-Laptop newfrontend %

DOCKER FILE



PRESSCRIPTION READER PAGE SCREENSHOT

The screenshot shows the TabCura home page. At the top, there is a navigation bar with links for Features, How it Works, Testimonials, Contact, Login, Sign Up, and Doctor Portal. Below the navigation is a header with the TabCura logo and a search bar. The main content area features a large section titled "Your Health Records, Simplified" with a subtext: "Securely store, track, and share your medical history and health records all in one place." Below this are three statistics: "10k+" Users, "100+" Healthcare Providers, and "98%" Satisfaction Rate. To the right is a "Patient Dashboard" section showing connected hospitals (2), total records (2), active prescriptions (2), recent medical records, and a prescription reader interface. A "Medical Timeline" graph is also present. At the bottom of the page are quick links for About Us, Support, and Connect.

LIVE WEBSITE

The screenshot shows the TabCura login page. The URL in the browser is 20.82.88.51:3000/login. The page has a "Welcome Back" heading and a subtext: "Login to access your TabCura health records". It features two input fields: "Email Address" and "Password", both with placeholder text "Enter your email" and "Enter your password". Below these is a blue "Login" button. At the bottom of the form, there is a link "Don't have an account? Sign Up" and a "Back to Home" link.

LOGIN PAGE

```

1 # Use Maven to build the app
2 FROM maven:3.9.4-eclipse-temurin-17 AS build
3 WORKDIR /app
4 COPY . .
5 RUN mvn clean package -skipTests
6
7 # Use JOK to run the app
8 FROM eclipse-temurin:17
9 WORKDIR /app
10 COPY --from=build /app/target/*.jar app.jar
11
12 # Run the jar
13 ENTRYPOINT ["java", "-jar", "app.jar"]
14

```

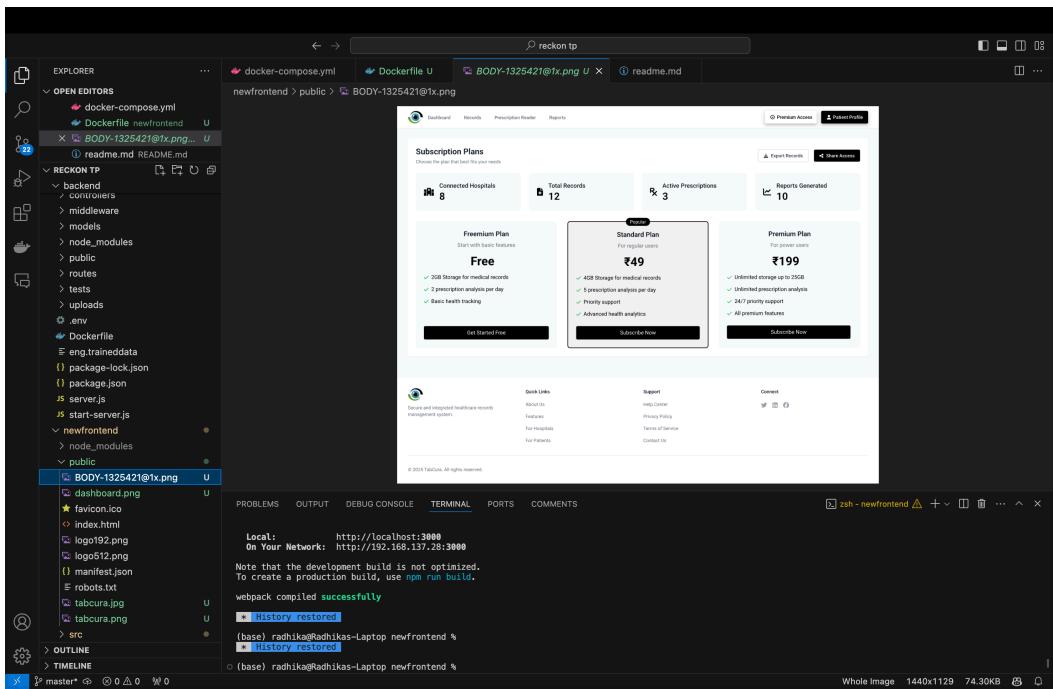
BACKEND DOCKER OF SPRINGBOOT

```

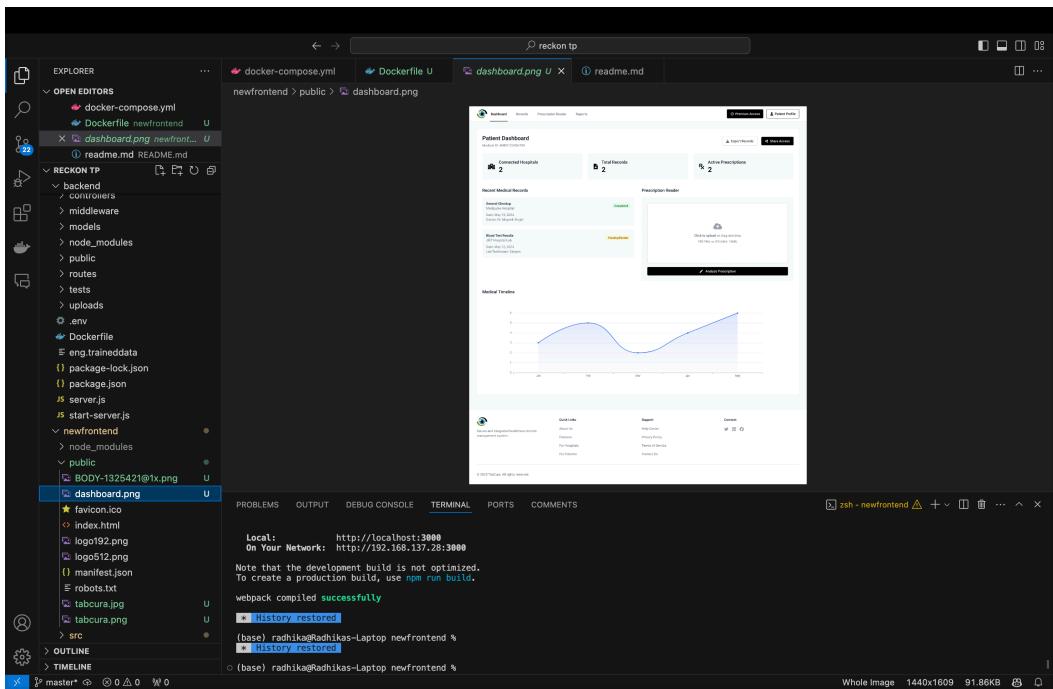
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>3.5.3</version>
9     <relativePath/>
10  </parent>
11  <groupId>com.radhika</groupId>
12  <artifactId>TabCurabackend</artifactId>
13  <version>0.0.1-SNAPSHOT</version>
14  <name>TabCurabackend</name>
15  <description>Demo project for Spring Boot</description>
16  <url>http://www.example.com</url>
17  <licenses>
18    <license>
19      <name>Apache 2.0</name>
20      <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
21    </license>
22  </licenses>
23  <developers>
24    <developer>
25      <connection>http://github.com/radhika</connection>
26      <developerConnection>http://github.com/radhika</developerConnection>
27      <tag>radhika</tag>
28    </developer>
29  </developers>
30  <scm>
31    <connection>http://github.com/radhika/TabCurabackend</connection>
32    <developerConnection>http://github.com/radhika/TabCurabackend</developerConnection>
33    <tag>v1.0.0</tag>
34  </scm>
35  <properties>
36    <java.version>17</java.version>
37  </properties>
38  <repositories>
39  </repositories>
40  <dependencyManagement>
41  </dependencyManagement>
42  <dependencies>
43    <dependency>
44      <groupId>org.springframework.boot</groupId>
45      <artifactId>spring-boot-starter-web</artifactId>
46      <version>3.5.3</version>
47    </dependency>
48  </dependencies>
49  <build>
50    <plugins>
51      <plugin>
52        <groupId>org.springframework.boot</groupId>
53        <artifactId>spring-boot-maven-plugin</artifactId>
54        <version>3.5.3</version>
55        <configuration>
56          <imageName>com.radhika/TabCurabackend</imageName>
57        </configuration>
58      </plugin>
59    </plugins>
60  </build>
61  <profiles>
62    <profile>
63      <id>prod</id>
64      <activation>
65        <activeByDefault>true</activeByDefault>
66      </activation>
67      <resources>
68        <resource>
69          <directory>src/main/resources/prod</directory>
70        </resource>
71      </resources>
72    </profile>
73  </profiles>
74  <ciManagement>
75    <ciSystem>Jenkins</ciSystem>
76  </ciManagement>
77  <distributionManagement>
78    <repository>
79      <id>central</id>
80      <name>Central Repository</name>
81      <url>http://repo.maven.apache.org/maven2</url>
82    </repository>
83  </distributionManagement>
84  <parentInformation>
85    <parentVersion>3.5.3</parentVersion>
86  </parentInformation>
87  <dependencyManagement>
88    <parent>
89      <groupId>org.springframework.boot</groupId>
90      <artifactId>spring-boot-dependencies</artifactId>
91      <version>3.5.3</version>
92    </parent>
93    <dependency>
94      <groupId>org.springframework.boot</groupId>
95      <artifactId>spring-boot-starter-parent</artifactId>
96      <version>3.5.3</version>
97    </dependency>
98  </dependencyManagement>
99  <dependencies>
100 </dependencies>
101 </project>

```

POM.XML OF TABCURA BACKEND USSING SPRINGBOOT



SUBSCRIPTION PLAN



DASHBOARD FOR DOCTORS

The screenshot shows a terminal window with several tabs open. The current tab, titled 'reckon tp', displays the contents of a Dockerfile. The Dockerfile includes instructions for building a React application using Create React App. It specifies mounting the 'public' directory from the host into the container at '/app/public'. It also installs dependencies and runs the application using 'npm start'. Below the Dockerfile, there is a note about running the build command to produce a production-ready bundle.

```
FROM node:12
WORKDIR /app
COPY package.json .
COPY package-lock.json .
RUN npm install
EXPOSE 3000
CMD ["npm", "start"]
```

Below the Dockerfile, the terminal shows the output of the 'npm run build' command, which successfully compiles the webpack code. The output indicates that the build was restored from history and was successful.

```
Local: http://localhost:3000
On Your Network: http://192.168.137.28:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully
  * History restored
(base) radhika@Radhikas-Laptop newFrontend %
  * History restored
(base) radhika@Radhikas-Laptop newFrontend %
```

USING REACT

CONNECTED FRONTEND WITH BACKEND