Simulation Transcript
Create a Docker container and install an app

Clara, the senior engineer, welcomes you to the project. You've got VS code open and you're ready to start. Clara points out the areas you'll be using. Notice the following three sections and hover over the **i** icons. Then, select the **Next arrow** to continue.

1. Clara explains that the first step is to create a file called a package.json. This file holds the information that Docker uses to create and run the web server. Clara creates the file and writes the first few lines of code. Take a moment to review the code that sets up the app name, version, and more. Select the **Next arrow** to continue.

2. The Code pad provides all the code you need to complete your coding work. You will refer to it throughout these steps. Select each field under Clara's code and then type each line of code exactly as you see it in the Code pad. For example, in the field at line 7, you would type, **"scripts":{**. It's time to try it out! Select the **Next arrow** to continue.

3. You'll complete the package.json file by writing the instructions that tell Node to start the web server. Below the last line in the package.json file, type the following code. Type one line of code in each field exactly as it displays in the Code pad. Select each field to start typing a line. When you finish typing all of the code press **Enter**.

   Code pad code:

   ```
   "scripts": {
       "start": "node server.js"
     },
     "dependencies": {
       "express": "^4.16.1"
     }
   }
   ```

4. Clara compliments you on your work. Well done! The senior engineer explains that the next step is to create a server.js file. This is a script that Node.js uses to provide instructions for how the web server will operate. Clara creates the file and writes the first few lines of code. Take a moment to review her code. Select the **Next arrow** to continue.

5. Clara asks you to complete this file by writing code to open communication between the browser and the web server using the settings she specified in her

earlier code. Below the last line of code in the server.js file, type the following code. Then, press **Enter**.

Code pad code:

```
app.listen(PORT, HOST, () => {
  console.log('Running on http://${HOST}:${PORT}');
});
```

6. The final file you'll need to create is the Dockerfile. This is the specific file used by Docker that tells it which software and dependencies to use in the container. Select the **Next arrow** to continue.

7. Clara creates the empty Dockerfile and asks you to write the instructions. Notice the commands include the RUN command to install npm into the image and the COPY command to copy all the files. Finally, the CMD command runs the node.js file you previously wrote which tells Docker to start the web server. Select the first line in the file and type the following instructions. Then, press **Enter**.

   Code pad code:

   ```
   FROM node:18-alpine
   WORKDIR /app
   COPY package*.json ./
   RUN npm install
   COPY . .
   EXPOSE 8080
   CMD ["node", "server.js"]
   ```

8. Clara creates a file to tell Docker to ignore all the existing Node.js files in the laptop's folder because the instructions you wrote tell Docker to install them. Without this file, the Docker image might include unnecessary files in the image increasing its size. Select the **Next arrow** to continue.

9. Now you're ready to build the container! Clara explains that you'll be doing this in a terminal window using the command line. Visual Studio Code has a terminal built into it. Clara opens this window so you can write the command to build the container. Select the **Next arrow** to continue.

10. Clara tells you to use Docker's "build" command to install the to-do app and build the container. Select the first line in the file and type the following instructions. Then, press **Enter**.

    Code pad code:
    ```
    docker build . -t todo-app
    ```

11. Clara asks you to notice that the terminal window confirms that the container has been built successfully. Nice work! Now that the container is built, you're ready to test it. Clara explains that you'll run a Docker command to launch the container and load the image which contains the web app. Select the **Next arrow** to continue.

12. To run the Docker container, place your cursor in the terminal window prompt and type the following command. Then, press **Enter**.

    Code pad:
    docker run -p 8080 -d todo-app

13. To verify the app is running, Clara opens the Docker app and highlights the running container with the to-do app image loaded. Now you can browse to the website that is running inside the container using Node.js to deliver the web pages. Select the **Next arrow** to continue.

14. Because it's a web app, you can test the application using a browser. The Docker app provides a link you can use to browse to the site. When you select the link, the to-do app opens in the browser. Select the highlighted **active port number** for the running container that has the web app running inside it.

15. Clara congratulates you on a successful build. Notice that the Node.js application launches when you run the image through the Docker engine. This means you correctly packaged the Docker image and the container is ready to be deployed to the cloud so employees all over the world can use it. Select the **Next arrow** to continue.

You successfully packaged a web application into a container image, created a Docker container and loaded the image into it, and ran the application inside the container. You also tested the application to ensure it ran from the container as expected.