This documentation will guide you through setting up a To-Do List web application using Docker Compose on Ubuntu 22.04. Docker Compose will help in managing the application's services and dependencies in a containerized environment.

INSTALL DOCKER AND DOCKER COMPOSE

First, we need to install Docker on our system.

```
sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"

apt-cache policy docker-ce

sudo apt install docker-ce

sudo systemctl status docker

sudo systemctl enable docker

sudo usermod -aG docker ${USER}

su - ${USER}

sudo usermod -aG docker username
```

Next, we will install Docker Compose.

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

mkdir ~/compose-demo

cd ~/compose-demo
```

```
mkdir app
```

```
nano app/index.html
```

```
<!doctype html>
<html lang="en">
<head>
   <meta charset="utf-8">
   <title>Docker Compose Demo</title>
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/kognise/water.css@latest/dist/dark.min.css">
</head>
<body>

   <h1>This is a Docker Compose Demo Page.</h1>

</body>
</html>
```

```
nano docker-compose.yml
```

Install the content you'd like based on what you will be running.

```
docker-compose up -d
```

CREATE THE DOCKER COMPOSE FILE

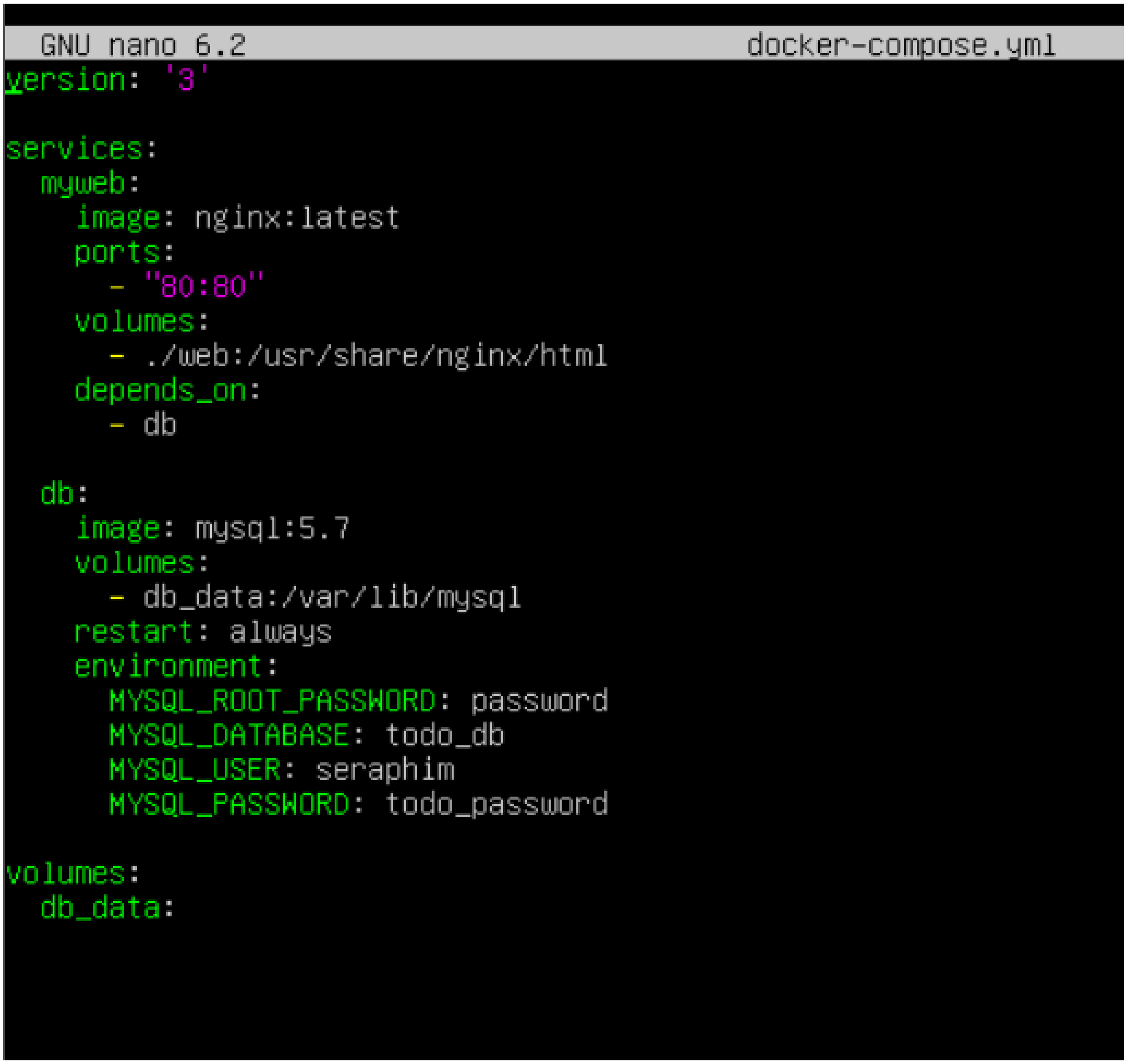
Create a directory for your project and navigate into it:

```
mkdir todo_list
```

```
cd todo_list
```

Now, create a docker-compose.yml file.

```
nano docker-compose.yml
```



```
  GNU nano 6.2                                   docker-compose.yml
version: '3'

services:
  myweb:
    image: nginx:latest
    ports:
      - "80:80"
    volumes:
      - ./web:/usr/share/nginx/html
    depends_on:
      - db

  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: password
      MYSQL_DATABASE: todo_db
      MYSQL_USER: seraphim
      MYSQL_PASSWORD: todo_password

volumes:
  db_data:
```

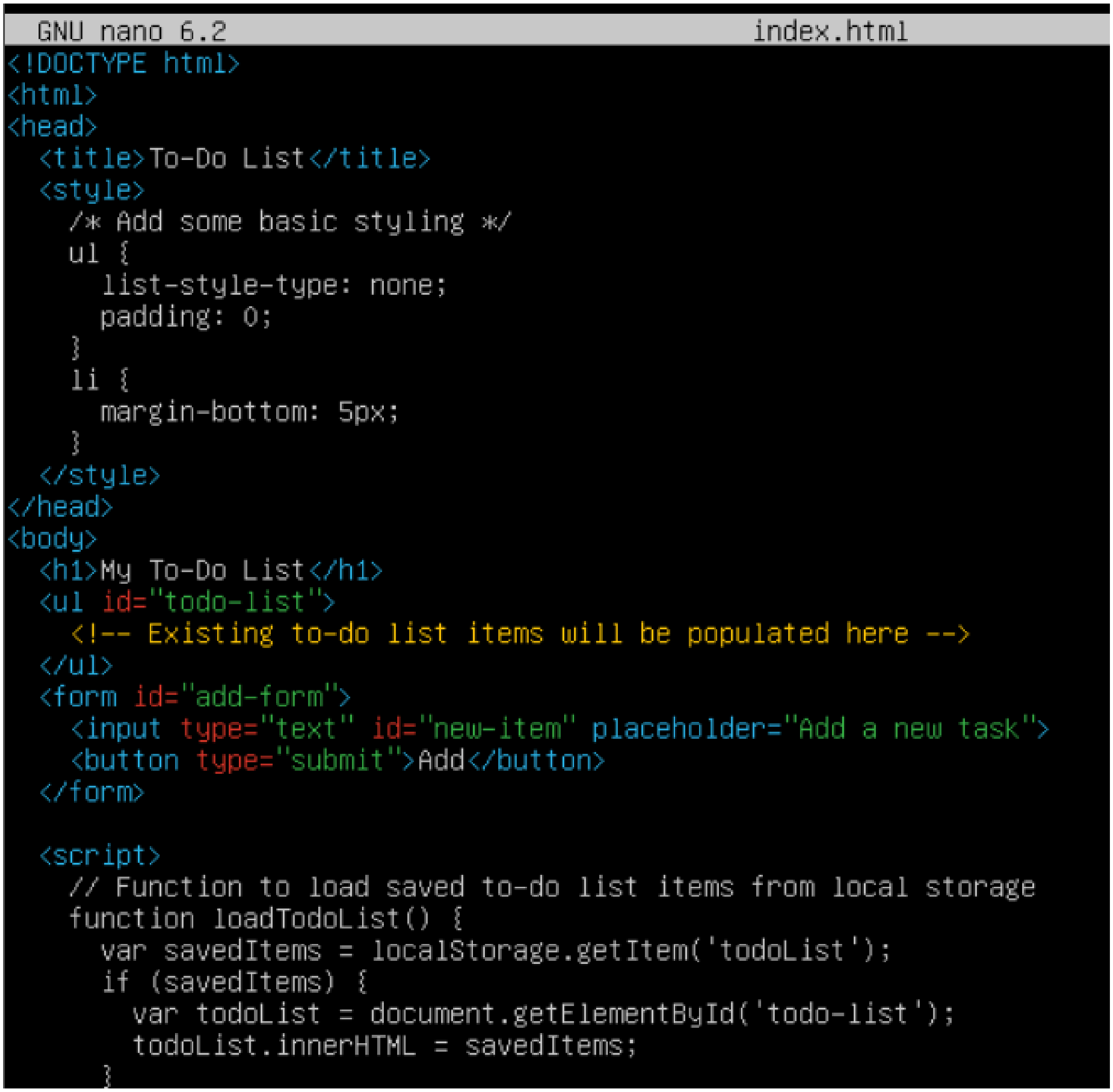
CREATE THE WEB APPLICATION

Create a web directory in your project and create an index.html file inside of it.

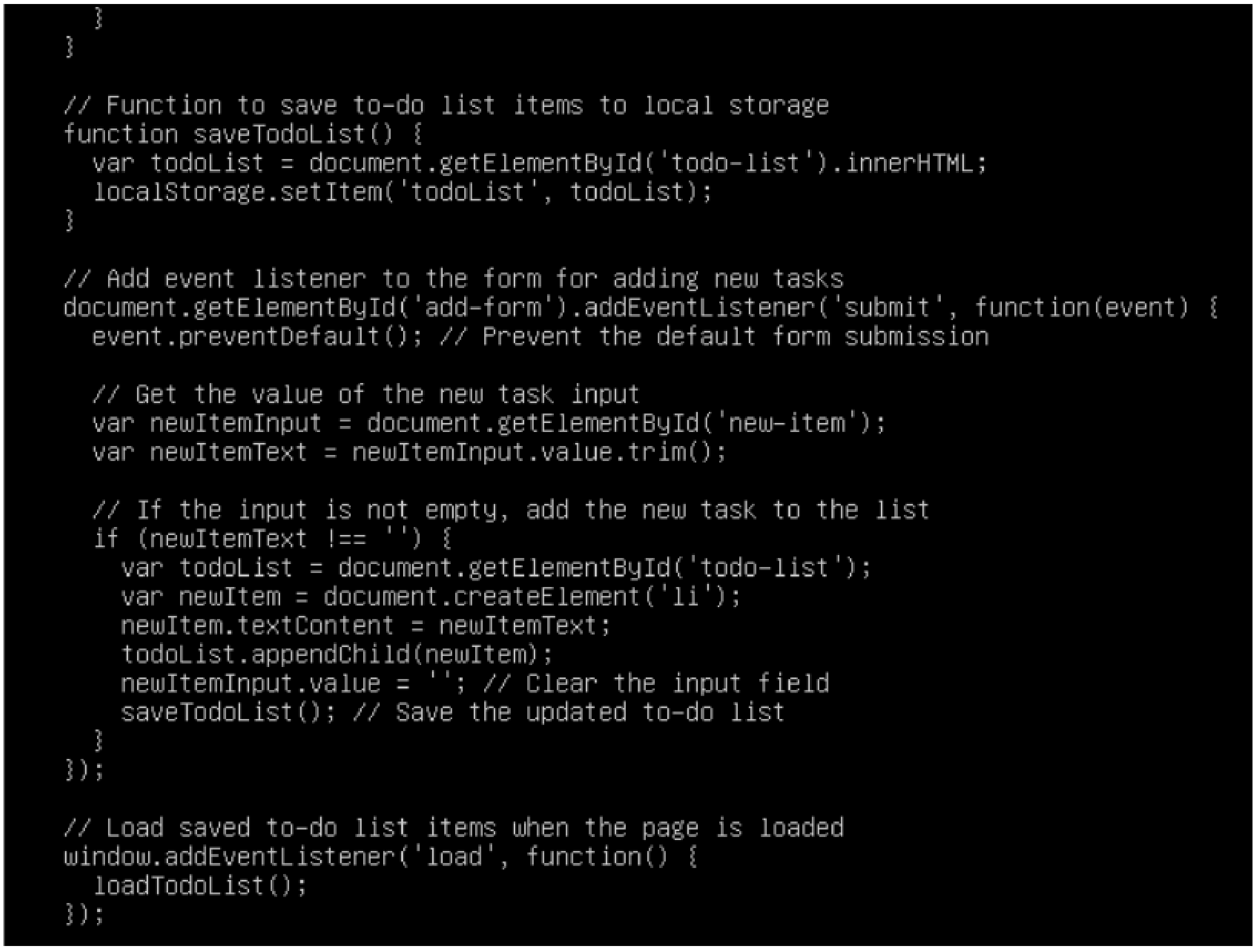
mkdir web

cd web

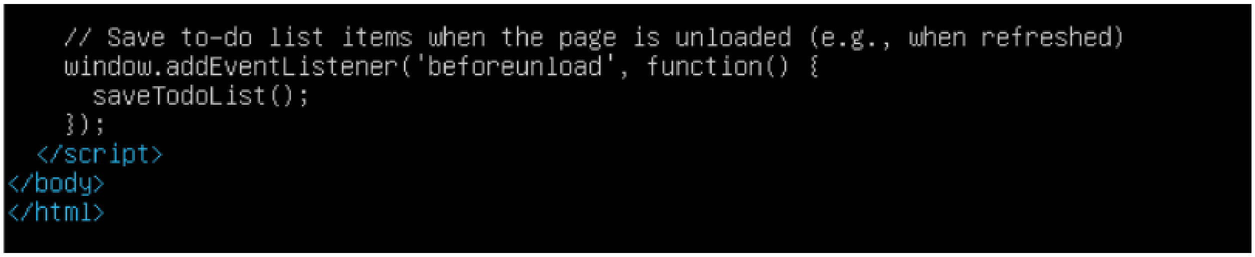
nano index.html



```
<!DOCTYPE html>
<html>
<head>
  <title>To-Do List</title>
  <style>
    /* Add some basic styling */
    ul {
      list-style-type: none;
      padding: 0;
    }
    li {
      margin-bottom: 5px;
    }
  </style>
</head>
<body>
  <h1>My To-Do List</h1>
  <ul id="todo-list">
    <!-- Existing to-do list items will be populated here -->
  </ul>
  <form id="add-form">
    <input type="text" id="new-item" placeholder="Add a new task">
    <button type="submit">Add</button>
  </form>

  <script>
    // Function to load saved to-do list items from local storage
    function loadTodoList() {
      var savedItems = localStorage.getItem('todoList');
      if (savedItems) {
        var todoList = document.getElementById('todo-list');
        todoList.innerHTML = savedItems;
      }
```



```
      }
    }

    // Function to save to-do list items to local storage
    function saveTodoList() {
      var todoList = document.getElementById('todo-list').innerHTML;
      localStorage.setItem('todoList', todoList);
    }

    // Add event listener to the form for adding new tasks
    document.getElementById('add-form').addEventListener('submit', function(event) {
      event.preventDefault(); // Prevent the default form submission

      // Get the value of the new task input
      var newItemInput = document.getElementById('new-item');
      var newItemText = newItemInput.value.trim();

      // If the input is not empty, add the new task to the list
      if (newItemText !== '') {
        var todoList = document.getElementById('todo-list');
        var newItem = document.createElement('li');
        newItem.textContent = newItemText;
        todoList.appendChild(newItem);
        newItemInput.value = ''; // Clear the input field
        saveTodoList(); // Save the updated to-do list
      }
    });

    // Load saved to-do list items when the page is loaded
    window.addEventListener('load', function() {
      loadTodoList();
    });
```



```
    // Save to-do list items when the page is unloaded (e.g., when refreshed)
    window.addEventListener('beforeunload', function() {
      saveTodoList();
    });
  </script>
</body>
</html>
```

START DOCKER COMPOSE

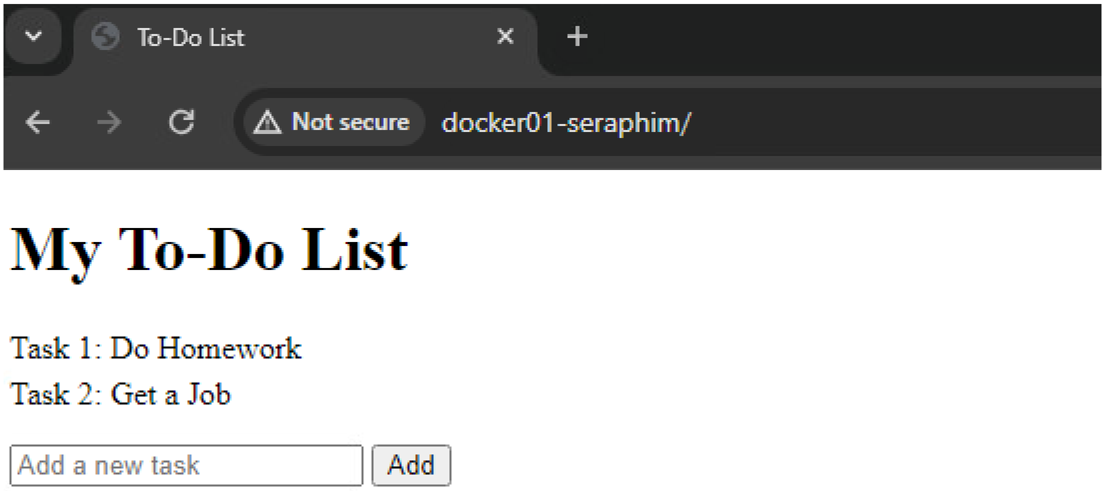
Now, navigate back to the root of the project and start Docker Compose.

cd ..

docker-compose up -d

ACCESS YOUR TO-DO LIST SITE

Once Docker Compose has started the containers, you can access your To-Do List site by opening a web browser and navigating to ‘http://[UBUNTU IP]:80.’



My To-Do List

Task 1: Do Homework

Task 2: Get a Job

Works Cited

Hogan, Brian. “How to Install and Use Docker on Ubuntu 20.04.” DigitalOcean, DigitalOcean, 29 Sept. 2021, www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04.

Tran, Tony, and Erika Heidi. “How to Install and Use Docker Compose on Ubuntu 20.04.” DigitalOcean, DigitalOcean, 28 Apr. 2022, www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-20-04.