WEB TECHNOLOGY PRACTICAL NO. 1

Create a responsive web page which shows the ecommerce/college/exam admin dashboard with sidebar and statistics in cards using HTML, CSS and Bootstrap.

SOURCE CODE:

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Urban Cart</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css"
rel="stylesheet">
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container-fluid">
    <div class="row">
      <nav id="sidebar" class="col-md-3 col-lg-2 d-md-block bg-dark sidebar collapse">
        <div class="d-flex flex-column vh-100">
          <div class="text-center p-3">
            <img src="shopping.png" alt="Logo" class="img-fluid" style="max-width: 80px;">
            <h4 class="text-white mt-2">Urban Cart</h4>
          </div>
          <ul class="nav flex-column p-3">
            <li class="nav-item">
              <a class="nav-link text-white" href="#">
                <i class="bi bi-house-door"></i>
                Dashboard
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="#">
                <i class="bi bi-cart"></i>
                Orders
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="#">
                <i class="bi bi-people"></i>
                Customers
              </a>
            </li>
            <li class="nav-item">
```

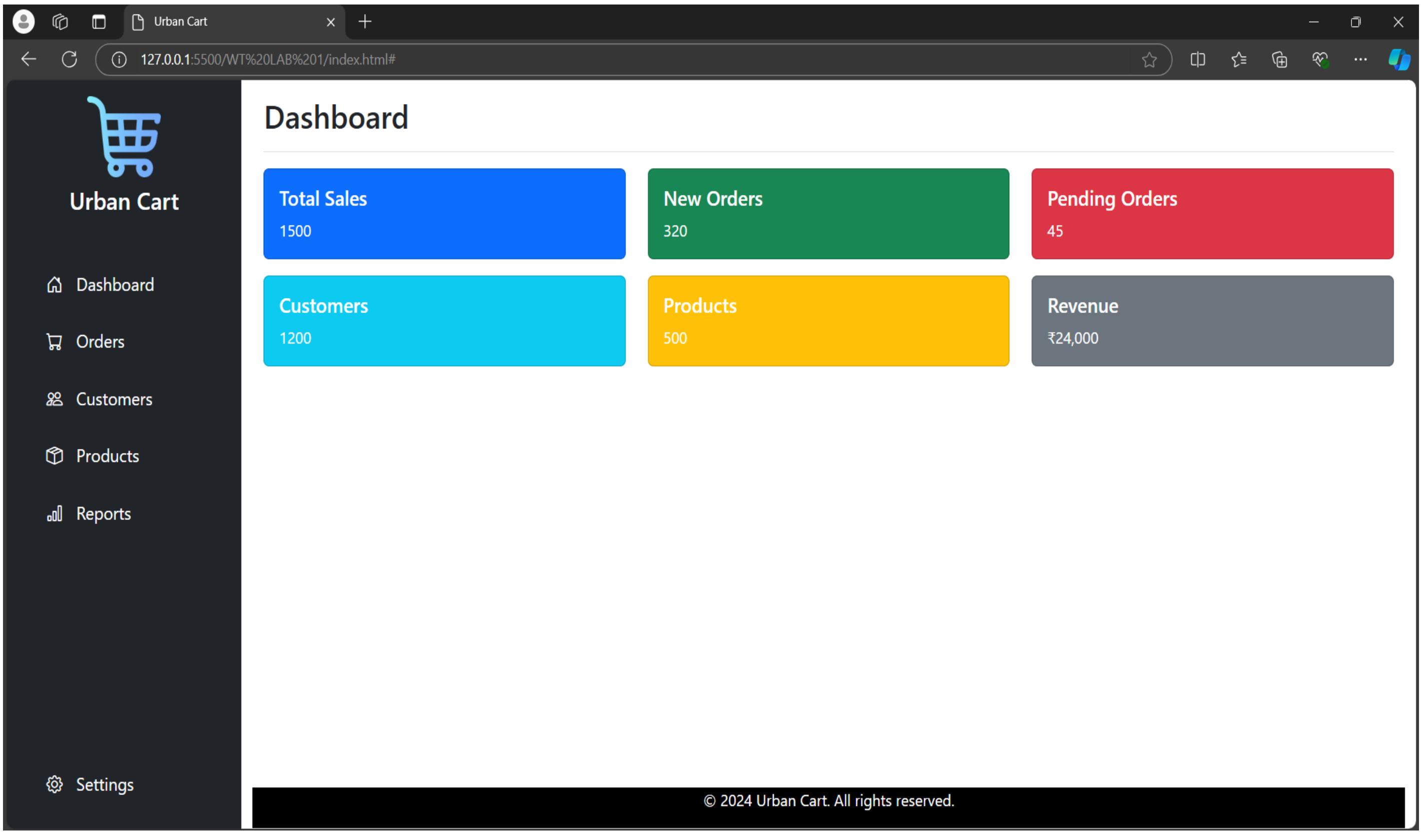
```
              <a class="nav-link text-white" href="#">
                <i class="bi bi-box-seam"></i>
                Products
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="#">
                <i class="bi bi-bar-chart"></i>
                Reports
              </a>
            </li>
          </ul>
          <div class="mt-auto p-3">
            <a class="nav-link text-white" href="#">
              <i class="bi bi-gear"></i> Settings
            </a>
          </div>
        </div>
      </nav>
      <main class="col-md-9 ms-sm-auto col-lg-10 px-md-4">
        <div class="d-flex justify-content-between flex-wrap flex-md-nowrap align-items-center
pt-3 pb-2 mb-3 border-bottom">
          <h1 class="h2">Dashboard</h1>
        </div>
        <div class="row">
          <div class="col-md-4">
            <div class="card text-white bg-primary mb-3">
              <div class="card-body">
                <h5 class="card-title">Total Sales</h5>
                <p class="card-text">1500</p>
              </div>
            </div>
          </div>

          <div class="col-md-4">
            <div class="card text-white bg-success mb-3">
              <div class="card-body">
                <h5 class="card-title">New Orders</h5>
                <p class="card-text">320</p>
              </div>
            </div>
          </div>

          <div class="col-md-4">
            <div class="card text-white bg-danger mb-3">
              <div class="card-body">
                <h5 class="card-title">Pending Orders</h5>
                <p class="card-text">45</p>
```

```
                            </div>
                        </div>
                    </div>
                </div>

                <div class="row">
                    <div class="col-md-4">
                        <div class="card text-white bg-info mb-3">
                            <div class="card-body">
                                <h5 class="card-title">Customers</h5>
                                <p class="card-text">1200</p>
                            </div>
                        </div>
                    </div>

                    <div class="col-md-4">
                        <div class="card text-white bg-warning mb-3">
                            <div class="card-body">
                                <h5 class="card-title">Products</h5>
                                <p class="card-text">500</p>
                            </div>
                        </div>
                    </div>

                    <div class="col-md-4">
                        <div class="card text-white bg-secondary mb-3">
                            <div class="card-body">
                                <h5 class="card-title">Revenue</h5>
                                <p class="card-text">₹24,000</p>
                            </div>
                        </div>
                    </div>
                    <div class="footer">
                        <p>&copy; 2024 Urban Cart. All rights reserved.</p>
                    </div>
                </div>
            </main>
        </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

- Styles.css

```
*{

      margin: 0;
      padding: 0;
    }
    #sidebar {
      height: 100vh;
      position: fixed;
      background-color: #343a40;
    }

    .sidebar .nav-link {
      font-size: 1.1rem;
      padding: 15px;
      color: #ffffff;
    }

    .sidebar .nav-link:hover {
      background-color: #495057;
      color: #ffffff;
    }

    .sidebar .nav-link i {
      margin-right: 10px;
    }

    .d-flex.flex-column {
      height: 100vh;
    }

    .main-content {
      padding: 2rem;
    }
    .footer{
      width: 100%;
      height: 40px;
      background-color: rgb(0, 0, 0);
      text-align: center;
      color: #ffffff;
      margin-top: 399px;

}
```

OUTPUT:



Urban Cart

- Dashboard
- Orders
- Customers
- Products
- Reports
- Settings

Dashboard

Total Sales
1500

New Orders
320

Pending Orders
45

Customers
1200

Products
500

Revenue
₹24,000

© 2024 Urban Cart. All rights reserved.

WEB TECHNOLOGY PRACTICAL NO. 2

Write a JavaScript Program to get the user registration data and push to array/local storage with AJAX POST method and data list in new page.

SOURCE CODE:

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Registration Form</title>
   <link rel="stylesheet" href="styles.css">
</head>
<body>
   <div class="form-container">
      <h2>Register Here</h2>
      <form id="registrationForm">
         <label for="name">Name:</label>
         <input type="text" id="name" name="name" required><br>

         <label for="email">Email:</label>
         <input type="email" id="email" name="email" required><br>

         <label for="username">Username:</label>
         <input type="text" id="username" name="username" required><br>

         <button type="submit">Submit</button>
      </form>
```

```
    <button id="viewUsersButton" onclick="location.href='users.html'">View Registered Users</button>

    <div id="responseContainer"></div>

  </div>


  <script src="script.js"></script>

</body>

</html>
```

- script.js

```
document.getElementById('registrationForm').addEventListener('submit', function (e) {

  e.preventDefault();


  let name = document.getElementById('name').value;

  let email = document.getElementById('email').value;

  let username = document.getElementById('username').value;


  let userData = {

    name: name,

    email: email,

    username: username

  };


  let xhr = new XMLHttpRequest();

  xhr.open('POST', 'https://jsonplaceholder.typicode.com/users', true);

  xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');


  xhr.onreadystatechange = function () {

    if (xhr.readyState === 4 && xhr.status === 201) {

      let response = JSON.parse(xhr.responseText);

      let users = JSON.parse(localStorage.getItem('registeredUsers')) || [];

      users.push(response);
```

```
            localStorage.setItem('registeredUsers', JSON.stringify(users));
            let responseContainer = document.getElementById('responseContainer');
            responseContainer.innerHTML = `
                <h3>Registered User Data:</h3>
                <p><strong>Name:</strong> ${response.name}</p>
                <p><strong>Email:</strong> ${response.email}</p>
                <p><strong>Username:</strong> ${response.username}</p>
                <p><strong>ID:</strong> ${response.id}</p>
            `;
        }
    };

    xhr.send(JSON.stringify(userData));
});
```

- styles.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
}

.form-container {
    width: 300px;
    margin: 100px auto;
    padding: 20px;
    background-color: white;
    border-radius: 5px;
    box-shadow: 0px 0px 10px 0px #aaa;
}

h2 {
    text-align: center;
```

```
}

label {
   display: block;
   margin-top: 10px;
}

input {
   width: 100%;
   padding: 8px;
   margin-top: 5px;
   box-sizing: border-box;
}

button {
   width: 100%;
   padding: 10px;
   margin-top: 20px;
   background-color: #28a745;
   color: white;
   border: none;
   cursor: pointer;
}

button:hover {
   background-color: #218838;
}

#responseContainer {
   margin-top: 20px;
   color: #333;
```

```
}
```

users.html

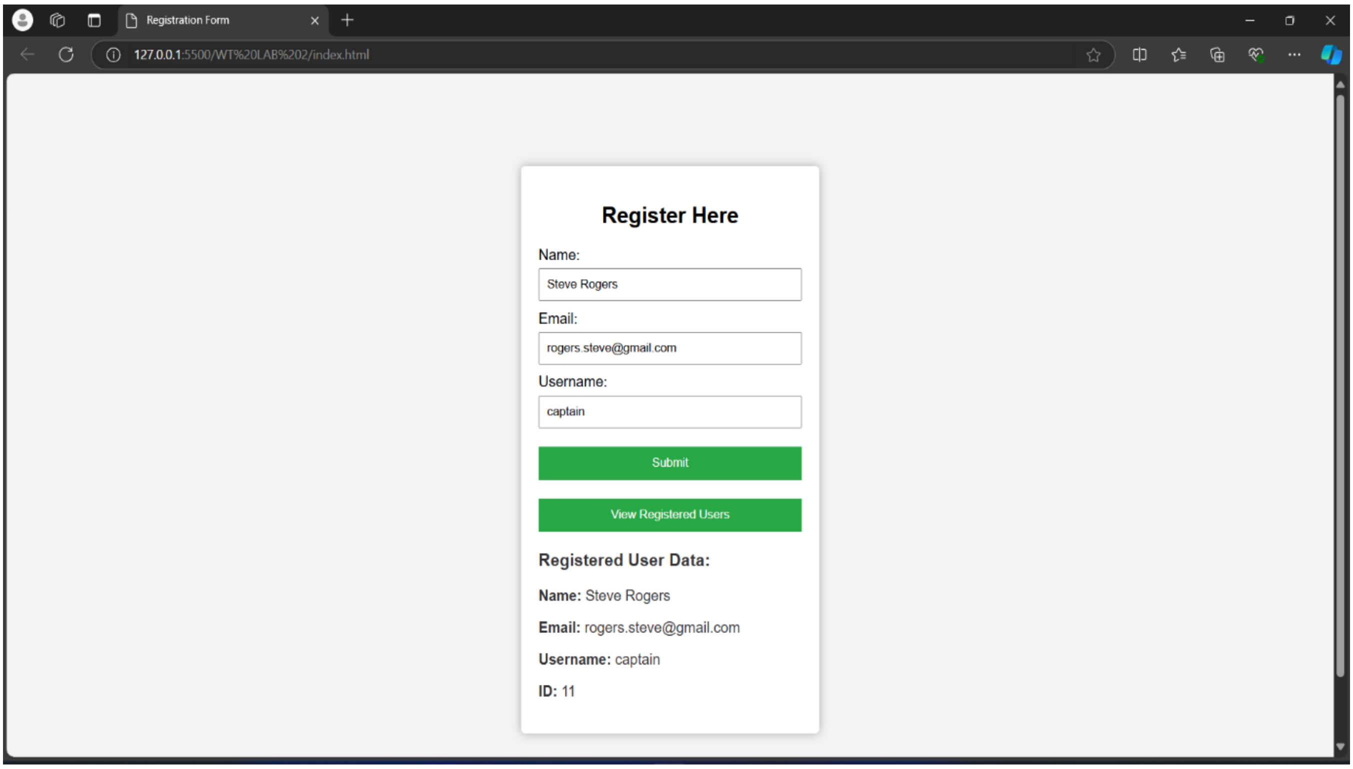
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registered Users</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="users-container">
    <h2>Registered Users</h2>
    <ul id="usersList"></ul>
    <button onclick="location.href='index.html'">Back to Registration</button>
  </div>

  <script src="users.js"></script>
</body>
</html>
```

script.js

```
window.onload = function () {
  let usersList = document.getElementById('usersList');
  let users = JSON.parse(localStorage.getItem('registeredUsers')) || [];
  if (users.length === 0) {
    usersList.innerHTML = '<p>No registered users found.</p>';
  } else {
    users.forEach(function (user) {
      let listItem = document.createElement('li');
      listItem.innerHTML = `<strong>Name:</strong> ${user.name}, <strong>Email:</strong> ${user.email}, <strong>Username:</strong> ${user.username}`;
```

```
            usersList.appendChild(listItem);
          });
        }
    };
```

OUTPUT:



Register Here

Name:

Steve Rogers

Email:

rogers.steve@gmail.com

Username:

captain

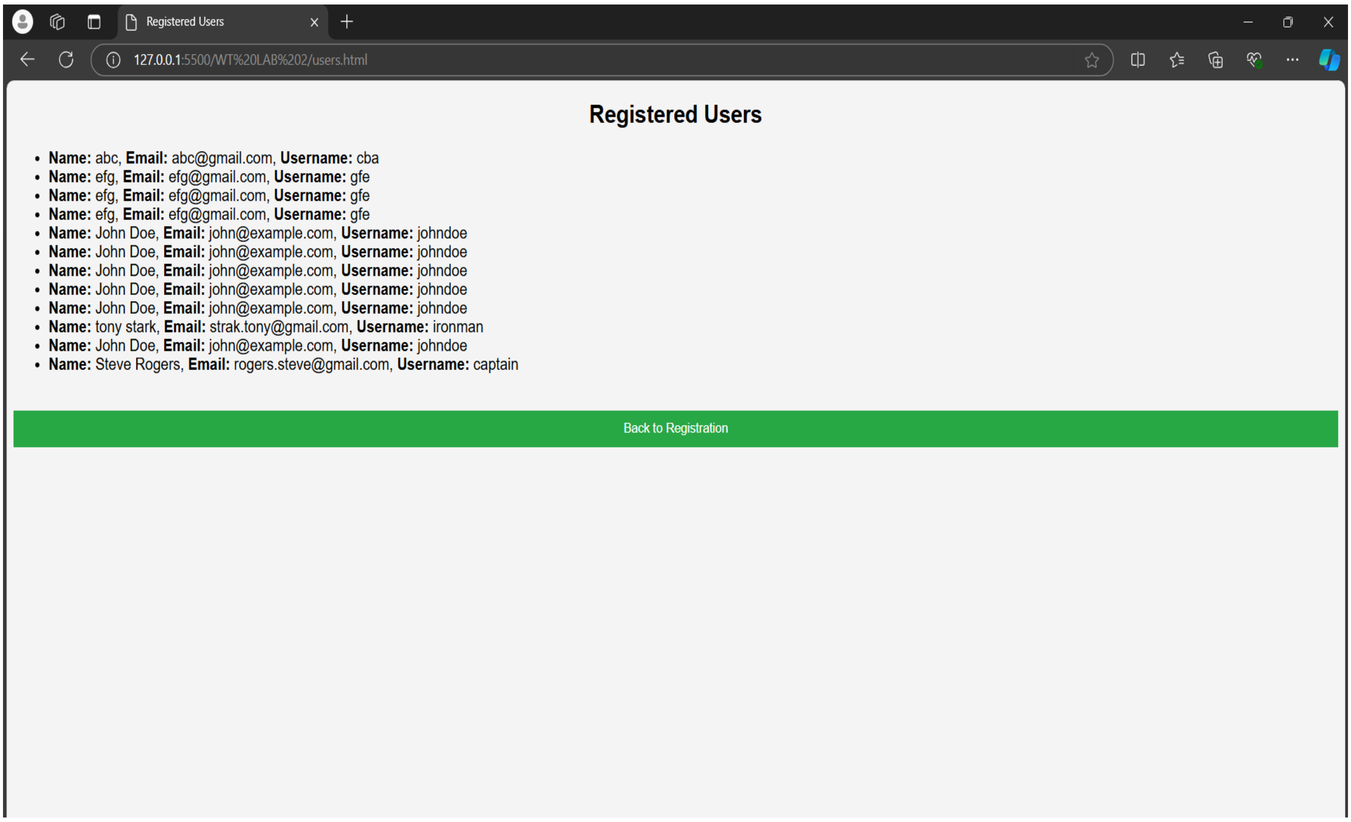
Registered User Data:

Name: Steve Rogers

Email: rogers.steve@gmail.com

Username: captain

ID: 11



Registered Users

- **Name:** abc, **Email:** abc@gmail.com, **Username:** cba
- **Name:** efg, **Email:** efg@gmail.com, **Username:** gfe
- **Name:** efg, **Email:** efg@gmail.com, **Username:** gfe
- **Name:** efg, **Email:** efg@gmail.com, **Username:** gfe
- **Name:** John Doe, **Email:** john@example.com, **Username:** johndoe
- **Name:** John Doe, **Email:** john@example.com, **Username:** johndoe
- **Name:** John Doe, **Email:** john@example.com, **Username:** johndoe
- **Name:** John Doe, **Email:** john@example.com, **Username:** johndoe
- **Name:** John Doe, **Email:** john@example.com, **Username:** johndoe
- **Name:** tony stark, **Email:** strak.tony@gmail.com, **Username:** ironman
- **Name:** John Doe, **Email:** john@example.com, **Username:** johndoe
- **Name:** Steve Rogers, **Email:** rogers.steve@gmail.com, **Username:** captain

WEB TECHNOLOGY PRACTICAL NO. 3 A

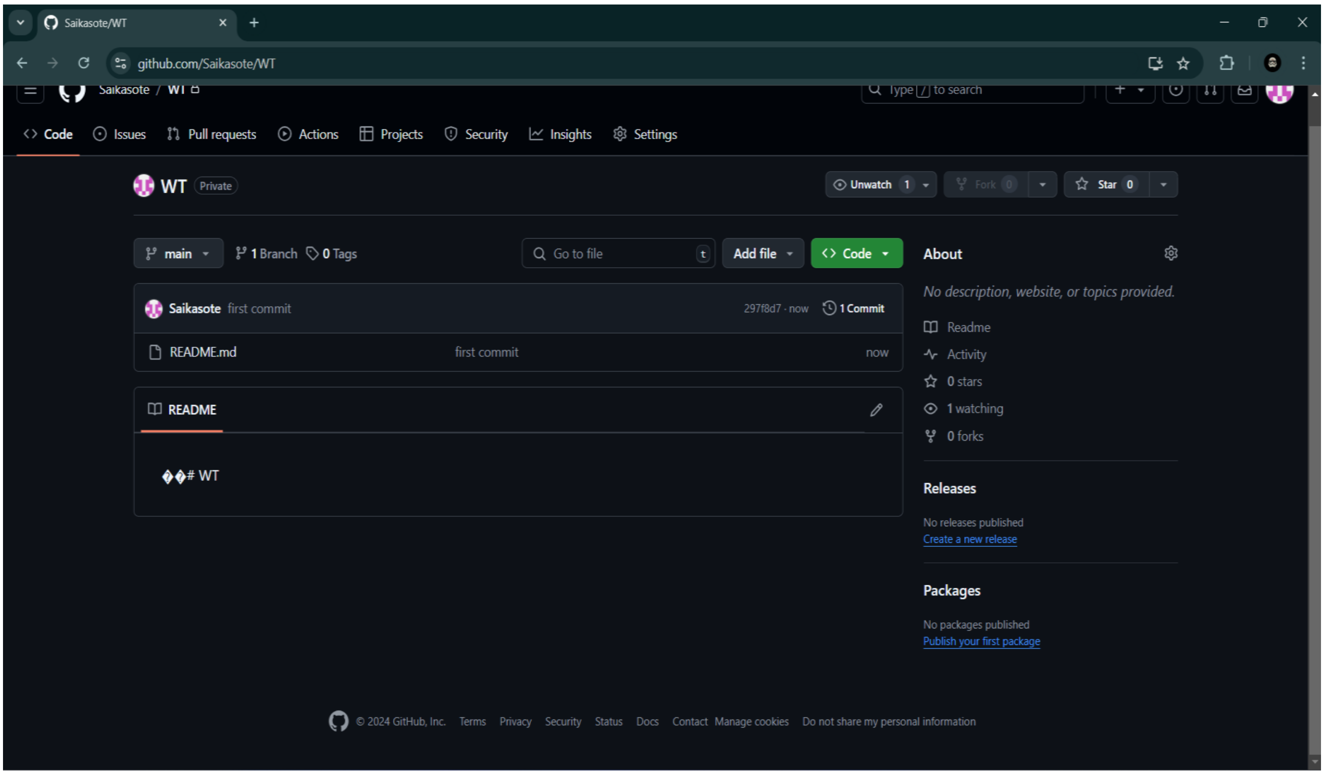
Create version control account on GitHub and using Git commands to create repository and push your code to GitHub.

COMMANDS:

// install git bash and create account on GitHub also create a repository over there.

```
echo "# WT" >> README.md
 git init
 git add README.md
 git commit -m "first commit"
 git branch -M main
 git remote add origin https://github.com/Saikasote/WT.git
 git push -u origin main
```

OUTPUT:



github.com/Saikasote/WT

Saikasote / WT

WT Private

main 1 Branch 0 Tags

Saikasote first commit 297f8d7 · now 1 Commit

README.md first commit now

README

��# WT

About

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

WEB TECHNOLOGY PRACTICAL NO. 3 B

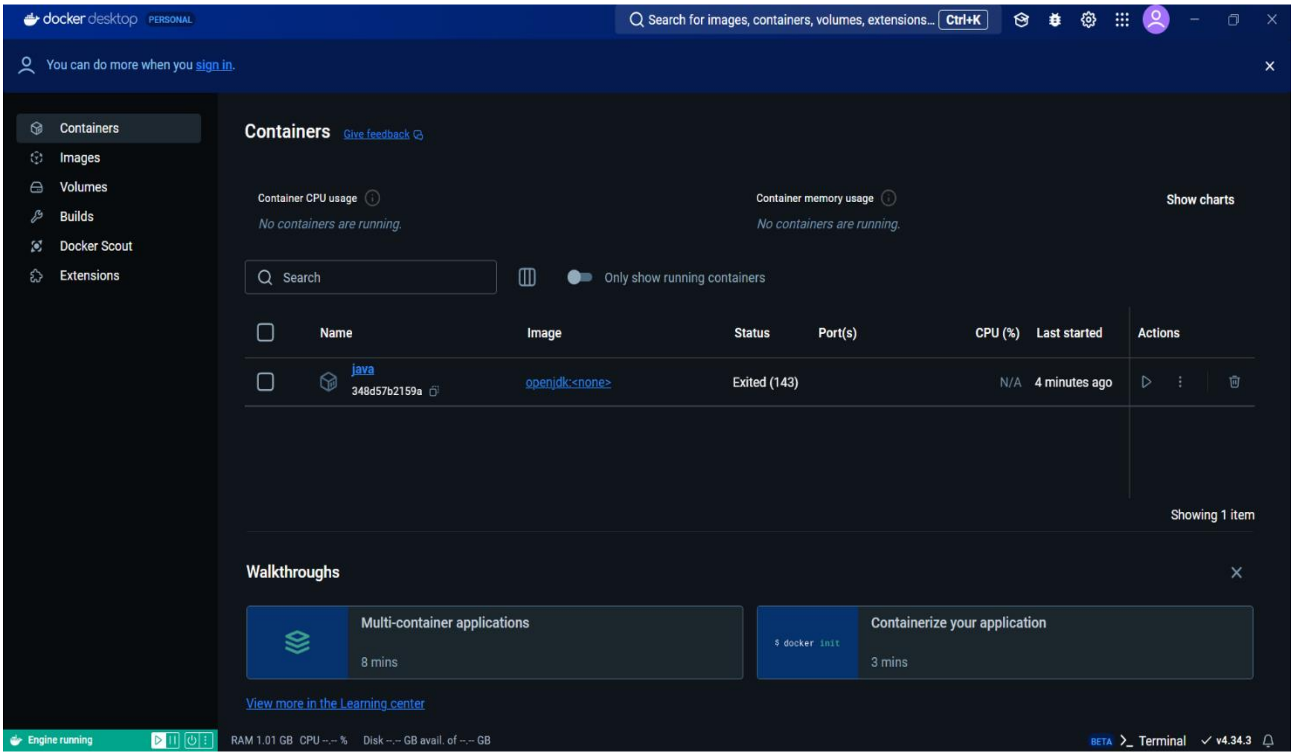
Create Docker Container Environment (NVIDEIA Docker or any other).

COMMANDS:

// Install docker.

```
docker –version

docker pull openjdk

docker images

docker run –name <container_name> -it -d openjdk

docker ps

docker exec -it <container_name>  jshell

int a=10;

int b = 20;

System.out.println(a+b);

/exit

docker stop <container_id>
```

OUTPUT:



You can do more when you sign in.

Container CPU usage
No containers are running.

Container memory usage
No containers are running.

Name	Image	Status	Port(s)	CPU (%)	Last started
java 348d57b2159a	openjdk:<none>	Exited (143)		N/A	4 minutes ago

Showing 1 item

RAM 1.01 GB CPU --.-- % Disk --.-- GB avail. of --.-- GB

WEB TECHNOLOGY PRACTICAL NO. 3 C

Create an Angular application which will do following actions: Register User, Login User, Show User Data on Profile Component

SOURCE CODE:

- login.component.css

```
.container {
  background-color: #fff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  width: 400px;
  margin-top: 50px;
  margin-left: 500px;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
h2 {
  text-align: center;
  margin-bottom: 20px;
}
form {
  display: flex;
  flex-direction: column;
}
label {
  margin-bottom: 5px;
  font-weight: bold;
  margin: 20px;
}
input[type="email"], input[type="password"]{
  padding: 10px;
  margin-bottom: 15px;
border-radius: 4px;
border: 1px solid #ccc;
font-size: 14px;
  }
select {
  padding: 10px;
  margin-bottom: 15px;
  border-radius: 4px;
  border: 1px solid #ccc;
  font-size: 14px;
}
#submit-btn {
  padding: 10px;
  background-color: #28a745;
  border: none;
  border-radius: 4px;
  color: white;
  font-size: 16px;
  cursor: pointer;
}
#submit-btn:hover {
  background-color: #218838;
}
```

- login.component.html

```
<div class="container">
   <h2>Login</h2>
   <form (ngSubmit)="login(f)" #f="ngForm">
     <div class="form-group">
       <label for="email">Email:</label>
       <input
         type="email"
         id="email"
         class="form-control"
         [(ngModel)]="email"
         name="email"
         required
       />
     </div>
     <div class="form-group">
       <label for="password">Password:</label>
       <input
         type="password"
         id="password"
         class="form-control"
         [(ngModel)]="password"
         name="password"
         required
       />
     </div>
     <p class="text-danger">{{ loginError }}</p>
     <button type="submit" id="submit-btn" class="btn btn-primary">Login</button>
   </form>
  </div>
```

- login.component.ts

```
import { Component, NgModule } from '@angular/core';

import {RouterModule } from '@angular/router';

import { FormsModule } from '@angular/forms';

import { NgModel } from '@angular/forms';

import { Router } from '@angular/router';

import { NgForm } from '@angular/forms';


@Component({
  selector: 'app-login',
  standalone: true,
  imports: [RouterModule,FormsModule],
  templateUrl: './login.component.html',
  styleUrl: './login.component.css'
})
export class LoginComponent {
  email: string = '';
  password: string = '';
  loginError: string = '';
  constructor(private router: Router) {}
  login(form: NgForm) {
    if (form.valid) {
      const storedData = localStorage.getItem('userData');
      if (storedData) {
        const userData = JSON.parse(storedData);
        // Check if email and password match
        if (this.email === userData.email && this.password === userData.password) {
          this.router.navigate(['/profile']); // Navigate to profile on successful login
        } else {
          this.loginError = 'Invalid email or password'; // Show error if credentials don't match
        }
      } else {
        this.loginError = 'No user data found. Please register first.';
      }
    } else {
      this.loginError = 'Please fill in both fields.';
    }
  }
}
```

- profile.component.css

```
.container {
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 400px;
    margin-top: 50px;
    margin-left: 500px;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
```

- profile.component.html

```
<div class="container mt-5">
    <h2>User Profile</h2>
    <div *ngIf="user">
      <p><strong>Name:</strong> {{ user.name }}</p>
      <p><strong>Email:</strong> {{ user.email }}</p>
      <p><strong>Date of Birth:</strong> {{ user.dateOfBirth }}</p>
      <p><strong>Gender:</strong> {{ user.gender }}</p>
    </div>
    <div *ngIf="!user">
      <p>No user data available.</p>
    </div>
  </div>
```

- profile.component.ts

```
import { Component,OnInit , Inject, PLATFORM_ID } from '@angular/core';
import { CommonModule } from '@angular/common';
import { isPlatformBrowser } from '@angular/common';


@Component({
  selector: 'app-profile',
  standalone: true,
  imports: [CommonModule],
  templateUrl: './profile.component.html',
  styleUrl: './profile.component.css'
})
export class ProfileComponent {
  user: any;
  errorMessage :string="";
  constructor(@Inject(PLATFORM_ID) private platformId: Object) {}
  ngOnInit() {
    if (isPlatformBrowser(this.platformId)) {
      const storedData = localStorage.getItem('userData');
      if (storedData) {
        this.user = JSON.parse(storedData);
      } else {
        this.errorMessage = 'No user data found';
      }
    } else {
      this.errorMessage = 'localStorage is not available';
    }
  }
  }
```

- register.component.css

```
.container {
   background-color: #fff;
   padding: 20px;
   border-radius: 8px;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   width: 400px;
   margin-top: 50px;
   margin-left: 500px;
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

h2 {
   text-align: center;
   margin-bottom: 20px;
}

form {
   display: flex;
   flex-direction: column;
}

label {
   margin-bottom: 5px;
   font-weight: bold;
   margin: 20px;
}

input[type="text"], input[type="email"], input[type="password"], input[type="date"] {
   padding: 10px;
   margin-bottom: 15px;
   border-radius: 4px;
   border: 1px solid #ccc;
   font-size: 14px;
}

select {
   padding: 10px;
   margin-bottom: 15px;
   border-radius: 4px;
   border: 1px solid #ccc;
   font-size: 14px;
}

#submit-btn {
   padding: 10px;
   background-color: #28a745;
   border: none;
   border-radius: 4px;
   color: white;
   font-size: 16px;
   cursor: pointer;
}

#submit-btn:hover {
   background-color: #218838;
}
```

- register.component.html

```
<div class="container mt-5">
  <h2>Register</h2>
  <form (ngSubmit)="register(f)" #f="ngForm">
    <div class="form-group">
      <label for="name">Name:</label>
      <input
        type="text"
        id="name"
        class="form-control"
        [(ngModel)]="name"
        name="name"
        required
      />
    </div>
    <div class="form-group">
      <label for="email">Email:</label>
      <input
        type="email"
        id="email"
        class="form-control"
        [(ngModel)]="email"
        name="email"
        required
      />
    </div>
    <div class="form-group">
      <label for="dateOfBirth">Date of Birth:</label>
      <input
        type="date"
        id="dateOfBirth"
        class="form-control"
        [(ngModel)]="dateOfBirth"
        name="dateOfBirth"
        required
      />
    </div>
    <div class="form-group">
      <label for="password">Password:</label>
      <input
        type="password"
        id="password"
        class="form-control"
        [(ngModel)]="password"
        name="password"
        required
      />
    </div>
    <div class="form-group">
      <label for="gender">Gender:</label>
      <select
        id="gender"
        class="form-control"
        [(ngModel)]="gender"
        name="gender"
        required
      >
        <option value="" disabled selected>Select Gender</option>
        <option value="male">Male</option>
        <option value="female">Female</option>
        <option value="other">Other</option>
      </select>
    </div>
    <p class="text-danger">{{ registrationError }}</p>
    <button type="submit" id="submit-btn" class="btn btn-primary">Register</button>
  </form>
</div>
```

- register.component.ts

```
import { Component } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { NgForm } from '@angular/forms';

@Component({
  selector: 'app-register',
  standalone: true,
  imports: [FormsModule],
  templateUrl: './register.component.html',
  styleUrl: './register.component.css'
})
export class RegisterComponent {
  name: string = '';
  email: string = '';
  dateOfBirth: string = '';
  password: string = '';
  gender: string = '';
  registrationError: string = '';

  register(form: NgForm) {
    if (form.valid) {
      const userData = {
        name: this.name,
        email: this.email,
        dateOfBirth: this.dateOfBirth,
        password: this.password,
        gender: this.gender,
      };

      // Store user data in local storage
      localStorage.setItem('userData', JSON.stringify(userData));
      form.reset();
    } else {
      this.registrationError = 'Please fill in all required fields';
    }
  }
}
```

- <u>app.component.css</u>

```
   .container {
     width: 100%;
     height: 60px;
     background-color: rgb(0, 0, 0);
     margin: 0;
     padding: 0;

   }

   .btn {

     width: 150px;
     height: 40px;
     align-self: center;
     border-radius: 5px;
     margin-top: 10px;
     margin-left: 5px;
     background-image: linear-gradient(144deg,#AF40FF, #5B42F3 50%,#00DDEB);
     font-family: Phantomsans, sans-serif;
     font-size: 20px;
     justify-content: center;
     line-height: 1em;
     color: #FFFFFF;

}
```

- app.component.html

```
<div id="page"><div class="container mt-5 text-center">
  <button class="btn btn-primary m-2" (click)="navigateToRegister()">Go to Register</button>
  <button class="btn btn-secondary m-2" (click)="navigateToLogin()">Go to Login</button>
</div>
</div>

<router-outlet></router-outlet>
```

- app.component.ts

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { RegisterComponent } from './register/register.component';
import { LoginComponent } from './login/login.component';
import { Router } from '@angular/router';
import { ProfileComponent } from './profile/profile.component';
import { Routes } from '@angular/router';

@Component({
 selector: 'app-root',
 standalone: true,
 imports: [RouterOutlet,LoginComponent,RegisterComponent,ProfileComponent],
 templateUrl: './app.component.html',
 styleUrl: './app.component.css'
})
export class AppComponent {
 title = 'application';
 constructor(private router: Router) {}

 navigateToRegister() {
  this.router.navigate(['/register']);
 }

 navigateToLogin() {
  this.router.navigate(['/login']);
 }

}
```

- app.routes.ts

```
import { Routes } from '@angular/router';
import { LoginComponent } from './login/login.component';
import { RegisterComponent } from './register/register.component';
import { FormsModule } from '@angular/forms';
import { ProfileComponent } from './profile/profile.component';

export const routes: Routes = [
  {path:"login",component:LoginComponent},
  {path:"register",component:RegisterComponent},
  {path:"profile",component:ProfileComponent}
];
```

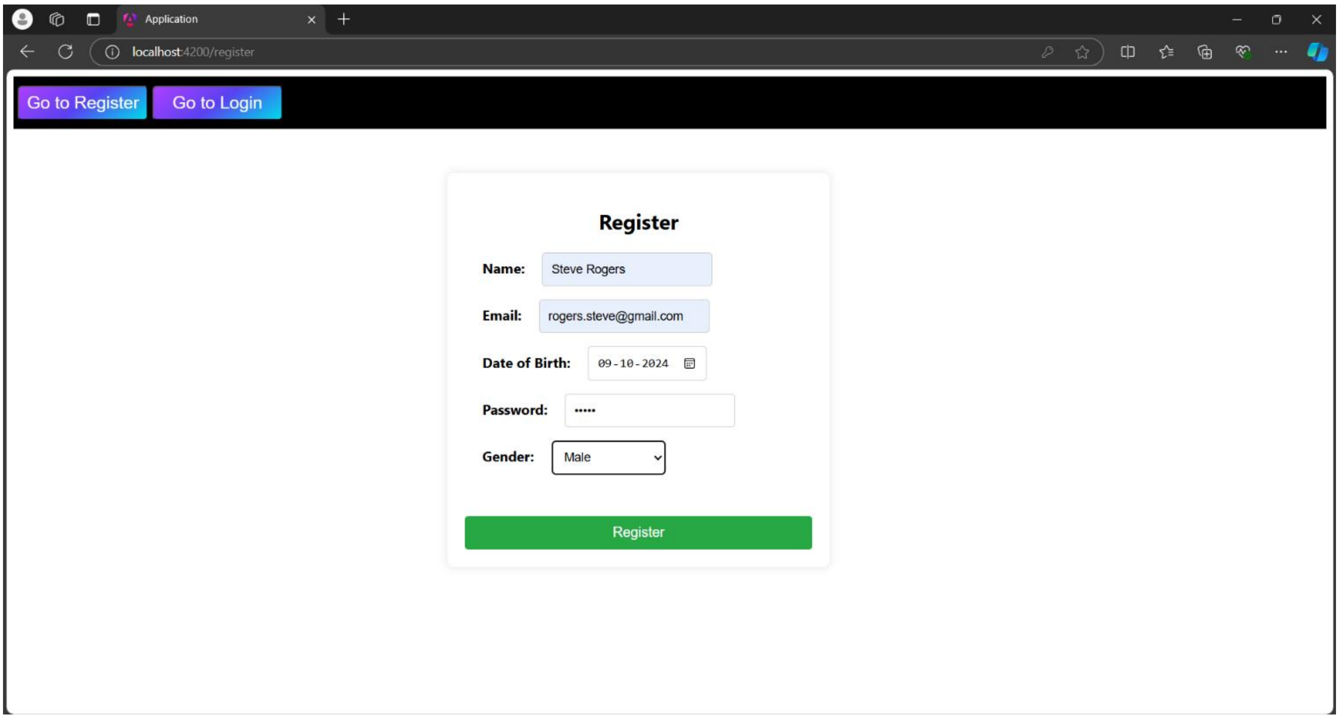
- main.ts

```
import { bootstrapApplication } from '@angular/platform-browser';
import { appConfig } from './app/app.config';
import { AppComponent } from './app/app.component';
import { provideRouter, Routes } from '@angular/router';
import { routes } from './app/app.routes';
import { FormsModule } from '@angular/forms';

bootstrapApplication(AppComponent,{providers:[provideRouter(routes)]})
  .catch((err) => console.error(err));
```

COMMANDS:

```
node --version

npm --version

npm install -g @angular/cli

ng --version

ng new <angular_project_name>

cd <angular_project_name>

ng g c <component_name>

ng serve
```

OUTPUT:



Register

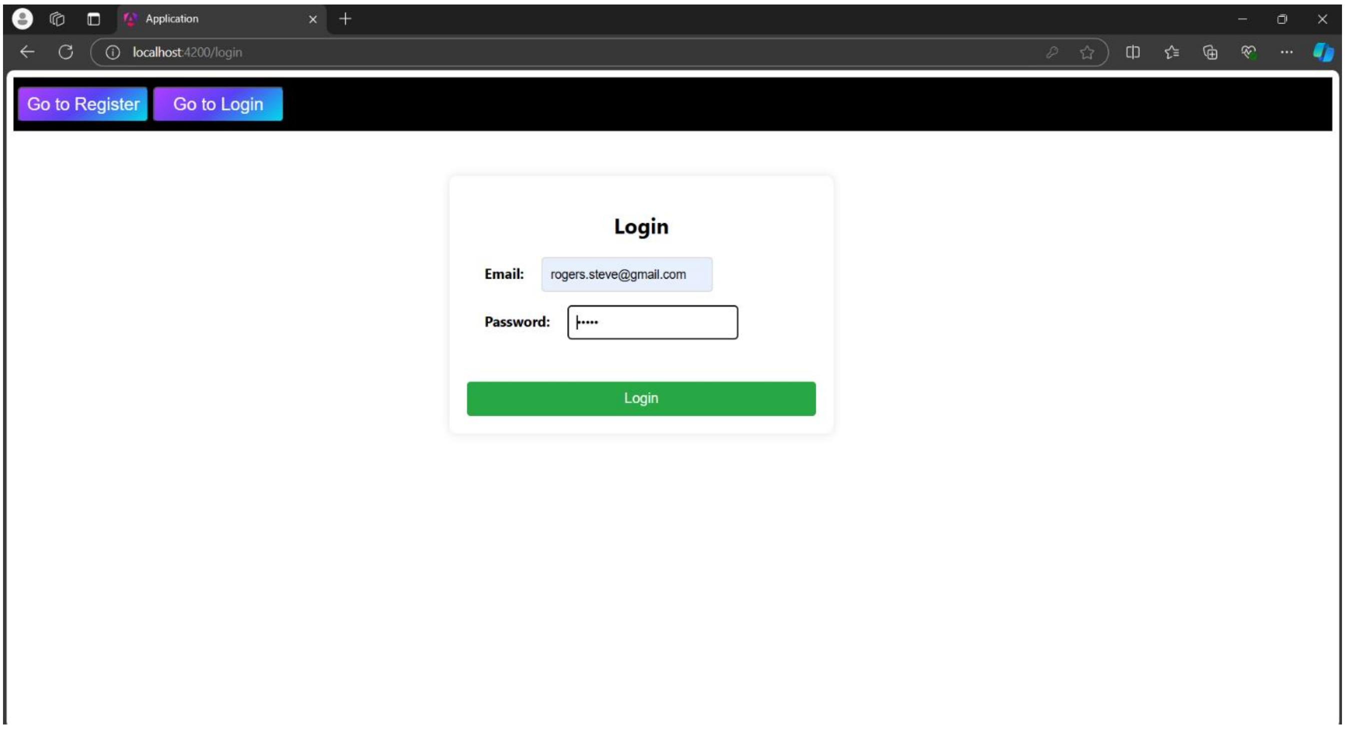
Name: Steve Rogers

Email: rogers.steve@gmail.com

Date of Birth: 09-10-2024

Password: •••••

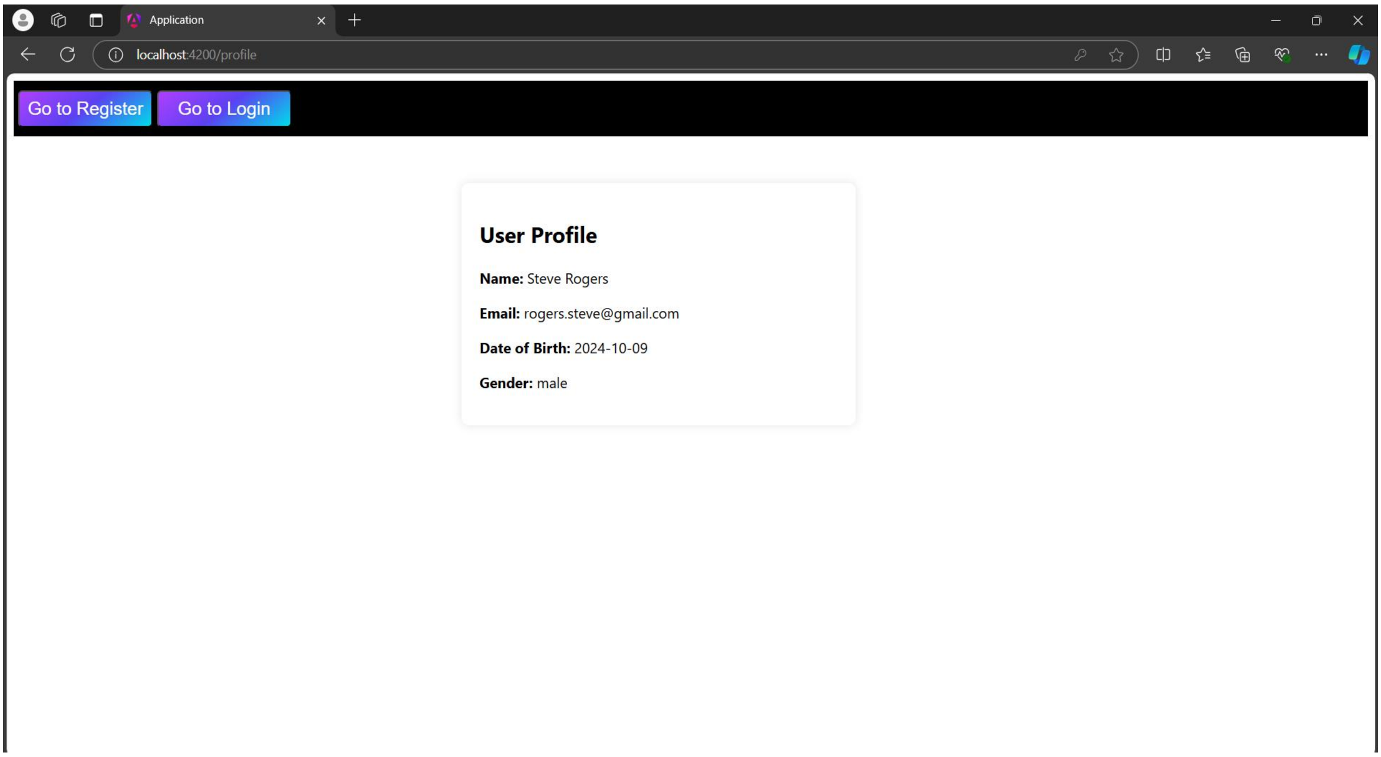
Gender: Male



Login

Email: rogers.steve@gmail.com

Password: •••••



User Profile

Name: Steve Rogers

Email: rogers.steve@gmail.com

Date of Birth: 2024-10-09

Gender: male

WEB TECHNOLOGY PRACTICAL NO. 4 A

Create a Node.JS Application which serves a static website.

SOURCE CODE:

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Wanderlust - Travel Blog</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#destinations">Destinations</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </nav>
  </header>

  <section id="home">
    <h1>Welcome to Wanderlust</h1>
    <p>Exploring the world's most beautiful destinations. Join us on our adventure!</p>
    <img src="pexels-asadphoto-240526.jpg" alt="Beautiful Beach" class="hero-image">
  </section>

  <section id="destinations">
```

```
      <h2>Featured Destinations</h2>
      <div class="destination">
        <img src="pexels-pixabay-417173.jpg" alt="Mountain Adventure">
        <h3>Mountain Adventure</h3>
        <p>Hike through breathtaking mountain ranges and experience nature like never before.</p>
      </div>
      <div class="destination">
        <img src="pexels-lkloeppel-466685.jpg" alt="City Exploration">
        <h3>City Exploration</h3>
        <p>Discover vibrant cities full of culture, history, and modern wonders.</p>
      </div>
    </section>

    <section id="contact">
      <h2>Contact Us</h2>
      <p>Have a question? Want to collaborate? Drop us a message through the form below:</p>
      <form action="https://docs.google.com/forms/d/e/[your_form_id]/formResponse" method="POST" target="_blank">
        <label for="name">Name:</label>
        <input type="text" id="name" name="entry.123456" required>

        <label for="email">Email:</label>
        <input type="email" id="email" name="entry.789012" required>

        <label for="message">Message:</label>
        <textarea id="message" name="entry.345678" rows="4" required></textarea>

        <button type="submit">Send</button>
      </form>
    </section>
    <footer>
        <p>&copy; 2024 Wanderlust. All rights reserved.</p>
    </footer>
</body>
</html>
```

- <u>styles.css</u>

```
body, h1, h2, p, ul, form {
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Arial', sans-serif;
  line-height: 1.6;
}

header {
  background-color: #2c3e50;
  color: #fff;
  padding: 10px 0;
}

header nav ul {
  list-style: none;
  display: flex;
  justify-content: center;
}

header nav ul li {
  margin: 0 20px;
```

```
}

header nav ul li a {
  color: #fff;
  text-decoration: none;
  font-weight: bold;
}

section {
  padding: 20px;
  margin: 20px;
  text-align: center;
}

#home {
  background: #ecf0f1;
  padding: 50px 20px;
}

.hero-image {
  width: 100%;
  max-width: 700px;
  height: auto;
  margin-top: 20px;
}

#destinations {
  background: #fff;
}

.destination {
  margin: 20px auto;
  max-width: 600px;
  text-align: left;
}

.destination img {
  width: 100%;
  height: auto;
  border-radius: 10px;
}

#contact {
  background: #ecf0f1;
}

form {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-top: 20px;
}

form label, form input, form textarea {
  width: 100%;
  max-width: 400px;
  margin-bottom: 10px;
}

form input, form textarea {
  padding: 10px;
  border: 1px solid #ccc;
```

```
}


button {
  padding: 10px 20px;
  background-color: #2c3e50;
  color: white;
  border: none;
  cursor: pointer;
}


button:hover {
  background-color: #34495e;
}


footer {
  background-color: #2c3e50;
  color: #fff;
  text-align: center;
  padding: 10px 0;
}
```

- app.js

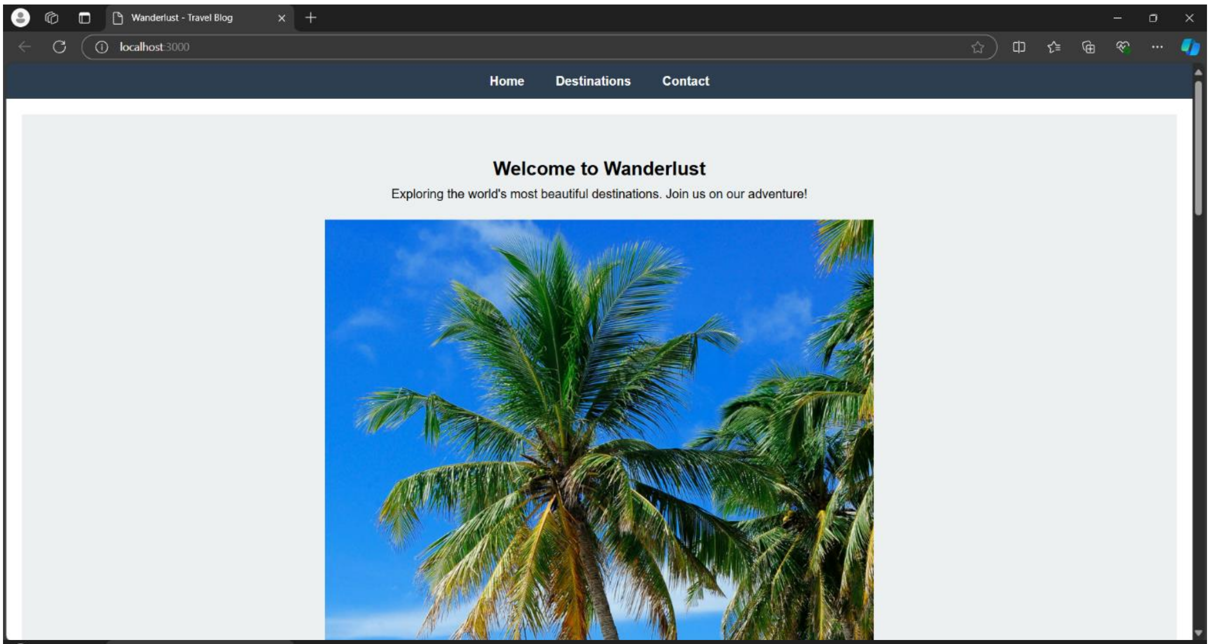
```
const express = require('express');
const path = require('path');

const app = express();

// Serve static files from the public directory
app.use(express.static(path.join(__dirname, 'public')));

// Route for home page
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'index.html'));
});

// Set port
const port = process.env.PORT || 3000;
```

```
app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

- <u>COMMANDS</u>

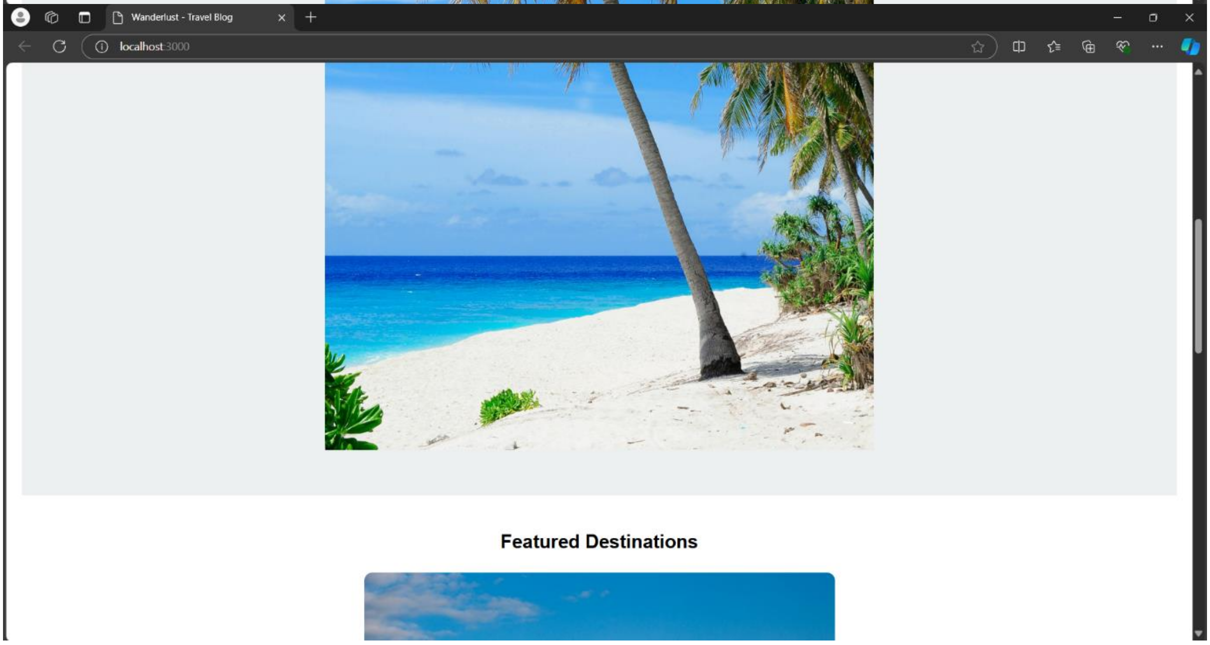
```
mkdir<foldername>

cd<foldername>

npm init -y

npm install express

node app.js
```

OUTPUT:

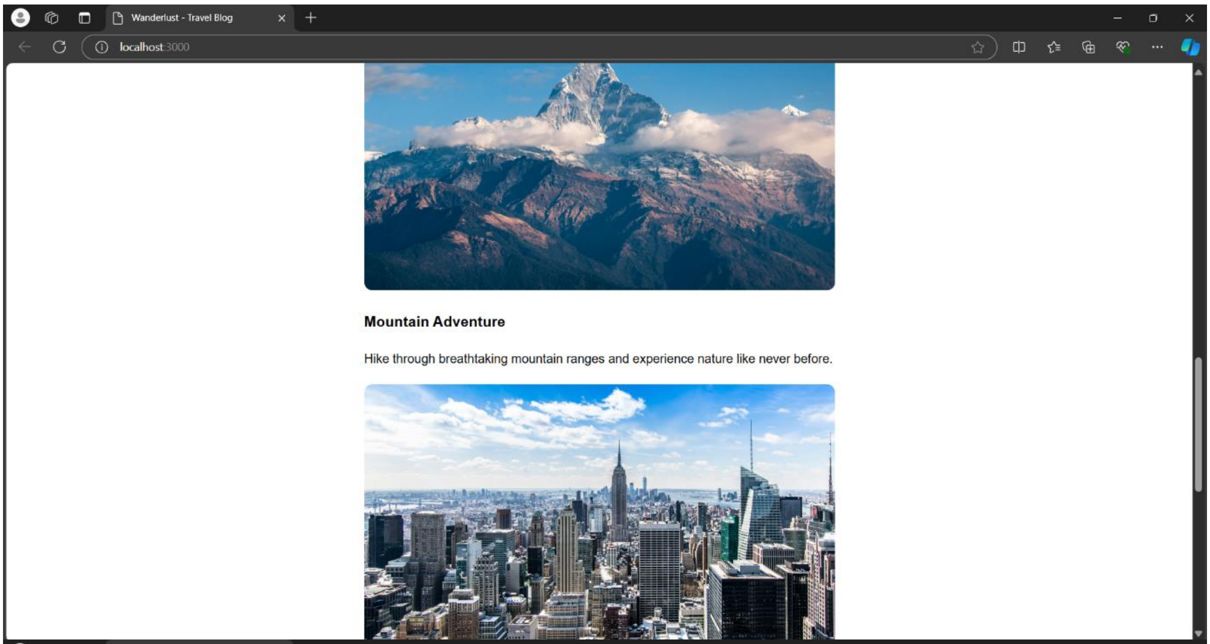


Welcome to Wanderlust

Exploring the world's most beautiful destinations. Join us on our adventure!

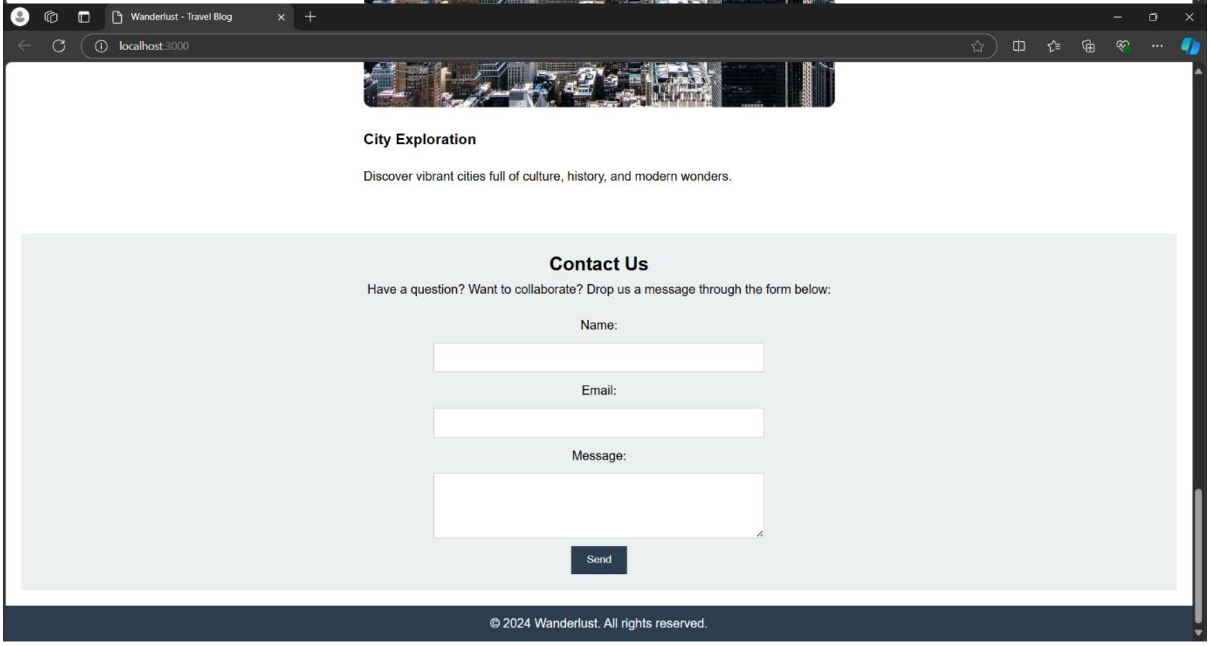


Featured Destinations



Mountain Adventure

Hike through breathtaking mountain ranges and experience nature like never before.



City Exploration

Discover vibrant cities full of culture, history, and modern wonders.

Contact Us

Have a question? Want to collaborate? Drop us a message through the form below:

Name:

Email:

Message:

© 2024 Wanderlust. All rights reserved.

WEB TECHNOLOGY PRACTICAL NO. 4 B

Create four API using Node.JS, ExpressJS and MongoDB for CURD Operations on assignment 3.C.

SOURCE CODE:

- mongobd.js

```
const {MongoClient} = require("mongodb");
const url = "mongodb://localhost:27017";
const database = "student";
const client = new MongoClient(url);

const db_connect = async() =>{
   const result = await client.connect();
   const db = await result.db(database);
   return db.collection("profile");
}

module.exports = db_connect;
```

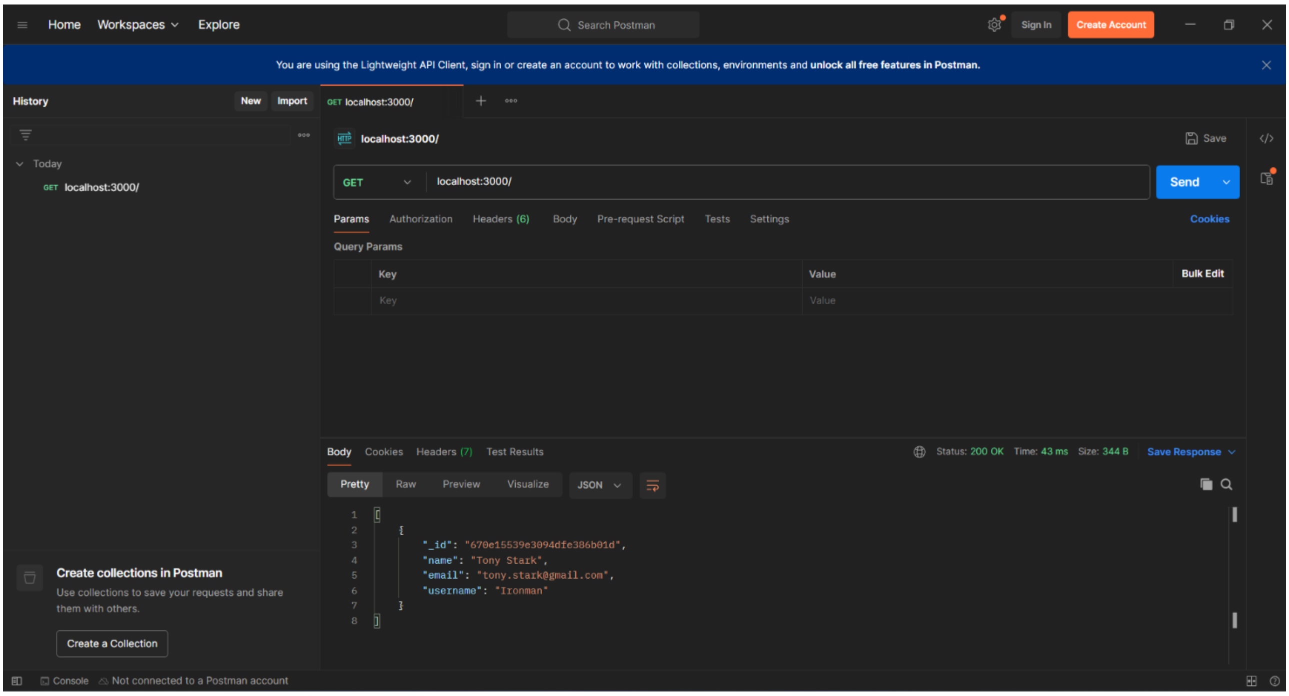
- index.js

```
const db_connect = require("./mongodb");
const express = require("express");
const { response } = require("express");
const app = express();
app.use(express.json())
// get api
app.get("/",async(req,res)=>{
   let result =  await db_connect();
   result = await result.find().toArray();
   res.send(result);
})
// post api
app.post("/",async(req,res)=>{
   let result =  await db_connect();
   result = await result.insertOne(req.body);
   res.send("data inserted successfully.......");
})
// put api
app.put("/:name",async(req,res)=>{
   let result =  await db_connect();
   result = await result.updateOne({name:req.params.name},{$set:req.body});
   res.send("data updated successfully.......");
})

//delete api
app.delete("/:name",async(req,res)=>{
    let result =  await db_connect();
    result = await result.deleteOne({name:req.params.name},{$set:req.body});
    res.send("data deleted successfully.......");
})

app.listen(3000);
```

COMMANDS:

```
// install mongodb and postman
npm init
npm i mongodb
npm i express
node index.js

// To kill a process
netstat -ano | findstr :3000
taskkill /PID <PID> /F
```

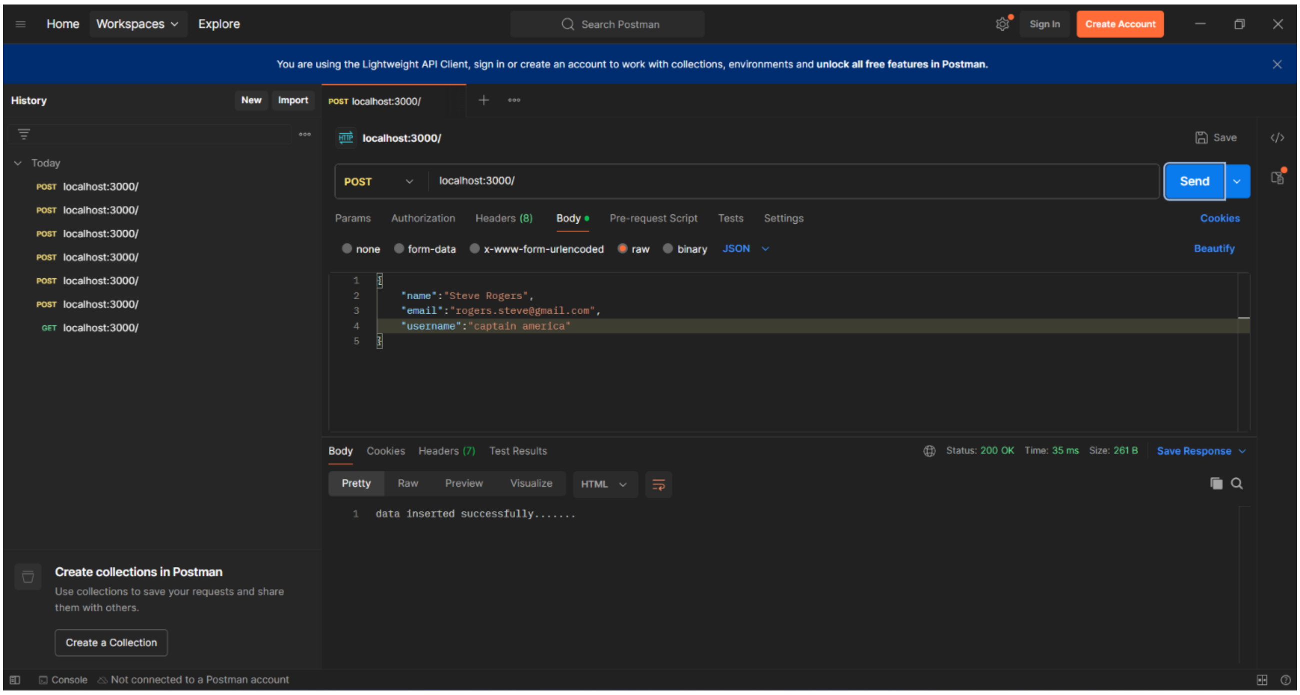
OUTPUT:



GET localhost:3000/

Status: 200 OK Time: 43 ms Size: 344 B

```
[
    {
        "_id": "670e15539e3094dfe386b01d",
        "name": "Tony Stark",
        "email": "tony.stark@gmail.com",
        "username": "Ironman"
    }
]
```



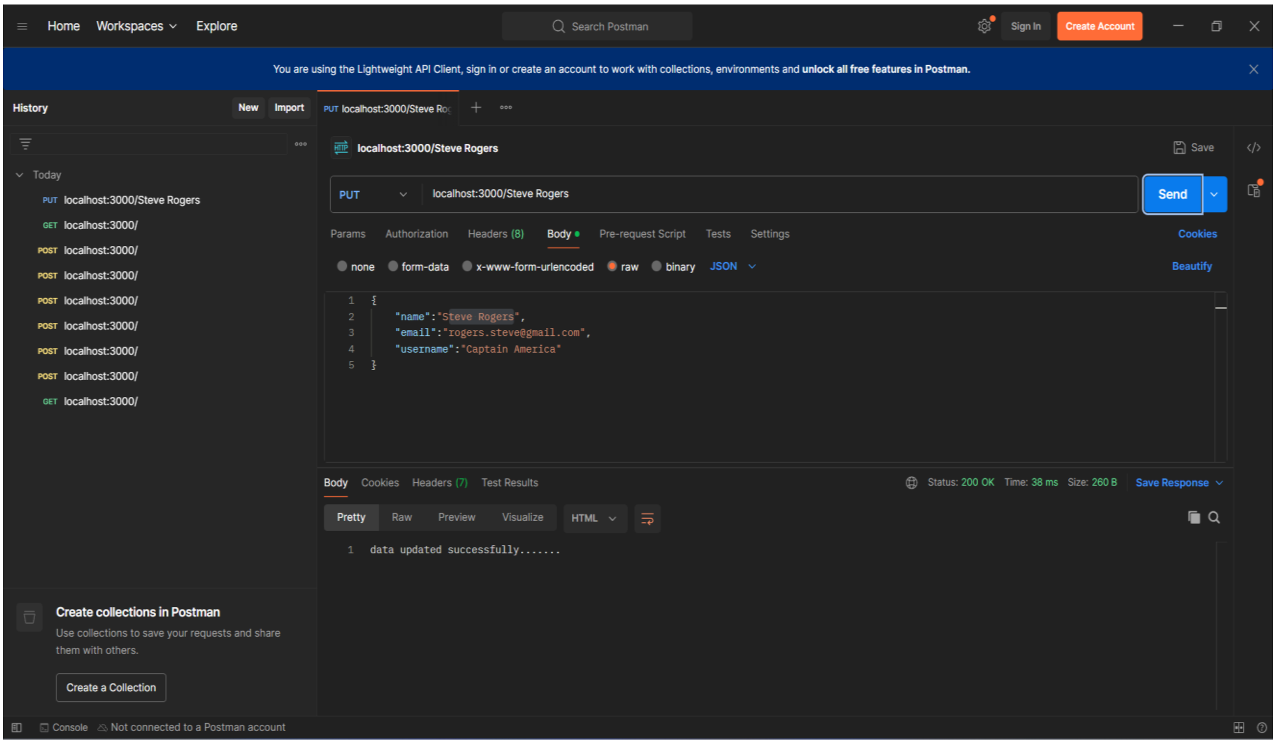
POST localhost:3000/

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON

```
{
    "name":"Steve Rogers",
    "email":"rogers.steve@gmail.com",
    "username":"captain america"
}
```

Status: 200 OK Time: 35 ms Size: 261 B

```
data inserted successfully.......
```



localhost:3000/Steve Rogers

PUT localhost:3000/Steve Rogers

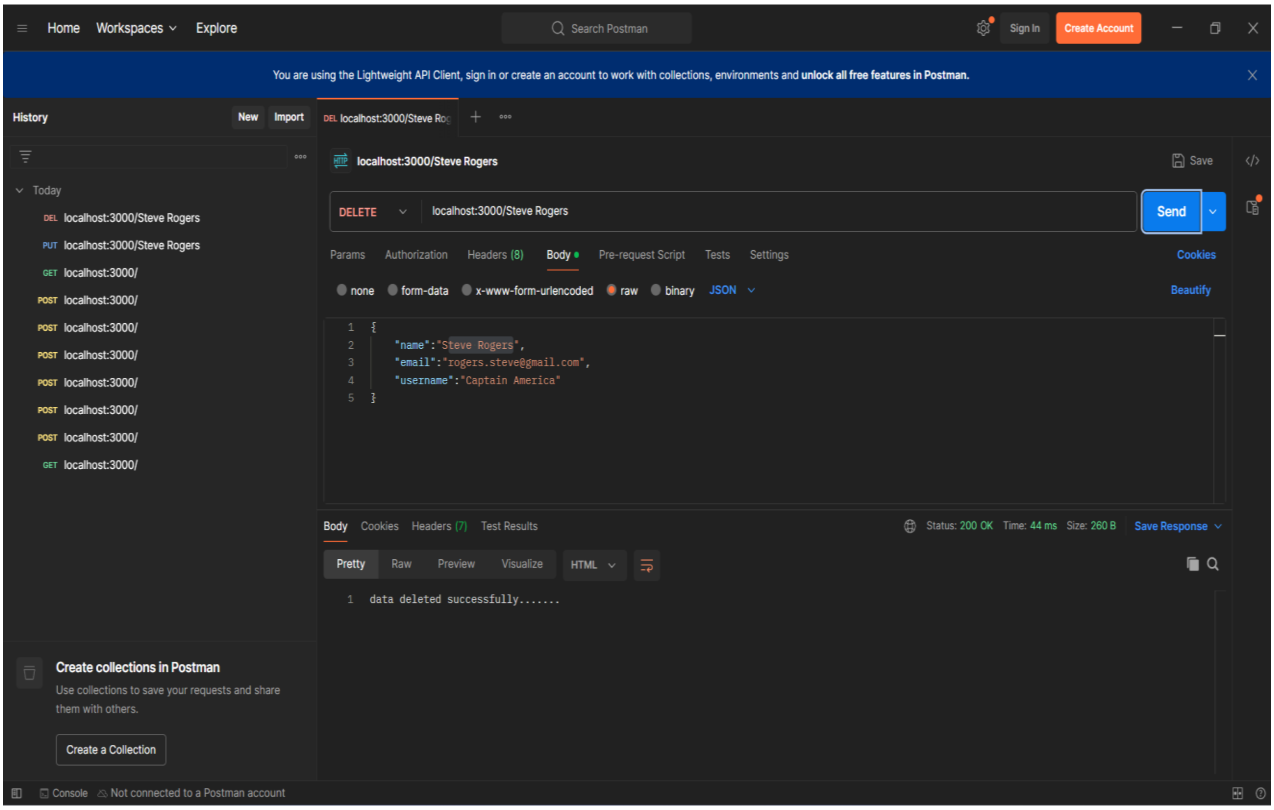
Send

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON

```
{
    "name":"Steve Rogers",
    "email":"rogers.steve@gmail.com",
    "username":"Captain America"
}
```

Body — Status: 200 OK Time: 38 ms Size: 260 B

```
data updated successfully.......
```



localhost:3000/Steve Rogers

DELETE localhost:3000/Steve Rogers

Send

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON

```
{
    "name":"Steve Rogers",
    "email":"rogers.steve@gmail.com",
    "username":"Captain America"
}
```

Body — Status: 200 OK Time: 44 ms Size: 260 B

```
data deleted successfully.......
```

WEB TECHNOLOGY PRACTICAL NO. 5 A

Create a simple Mobile Website using jQuery Mobile.

SOURCE CODE:

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Enhanced Mobile Website with Colors</title>
  <link rel="stylesheet" href="https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css" />
  <link rel="stylesheet" href="styles.css">
  <script src="https://code.jquery.com/jquery-1.11.1.min.js"></script>
  <script src="https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>

</head>
<body>

<div data-role="page" id="home">
  <div data-role="header">
    <h1>Home</h1>
  </div>

  <div role="main" class="ui-content">
    <p id="intro_text">Welcome to our mobile website! We create solutions for a mobile world.</p>

    <!-- Services Section -->
    <h2>Our Services</h2>
    <ul data-role="listview" data-inset="true">
      <li><a href="#service1">Mobile App Development</a></li>
      <li><a href="#service2">Web Design</a></li>
      <li><a href="#service3">SEO Optimization</a></li>
    </ul>

    <!-- Image Gallery -->
    <h2>Gallery</h2>
    <div class="ui-grid-a">
      <div class="ui-block-a"><img src="app-development.png" alt="Image 1"></div>
      <div class="ui-block-b"><img src="app-development (2).png" alt="Image 2"></div>
      <div class="ui-block-a"><img src="search-engine-optimization.png" alt="Image 3"></div>
      <div class="ui-block-b"><img src="coding.png" alt="Image 4"></div>
    </div>

    <a href="#about" class="ui-btn ui-corner-all ui-shadow">About Us</a>
    <a href="#contact" class="ui-btn ui-corner-all ui-shadow">Contact Us</a>
  </div>

  <div data-role="footer">
    <h4>Footer Text</h4>
  </div>
</div>

<!-- Page 2: About -->
<div data-role="page" id="about">
  <div data-role="header">
    <h1>About Us</h1>
  </div>

  <div role="main" class="ui-content">
    <h2>Who We Are</h2>
    <p>We are a passionate team of developers, designers, and strategists who build amazing digital products. Our focus is on mobile-first design to ensure seamless user experiences across all devices.</p>
    <!-- Testimonials Section -->
    <h2>Testimonials</h2>
    <div data-role="collapsible-set">
      <div data-role="collapsible">
        <h3>Customer 1</h3>
        <p >"This company is amazing! They transformed our web presence!"</p>
      </div>
      <div data-role="collapsible">
        <h3>Customer 2</h3>
        <p>"Their mobile app development is top-notch. We’ve seen a 200% increase in mobile traffic!"</p>
      </div>
    </div>

    <a href="#home" class="ui-btn ui-corner-all ui-shadow">Back to Home</a>
  </div>

  <div data-role="footer">
    <h4>Footer Text</h4>
  </div>
</div>

<!-- Page 3: Contact -->
<div data-role="page" id="contact">
  <div data-role="header">
    <h1>Contact Us</h1>
  </div>

  <div role="main" class="ui-content">
    <h2>Get In Touch</h2>
```

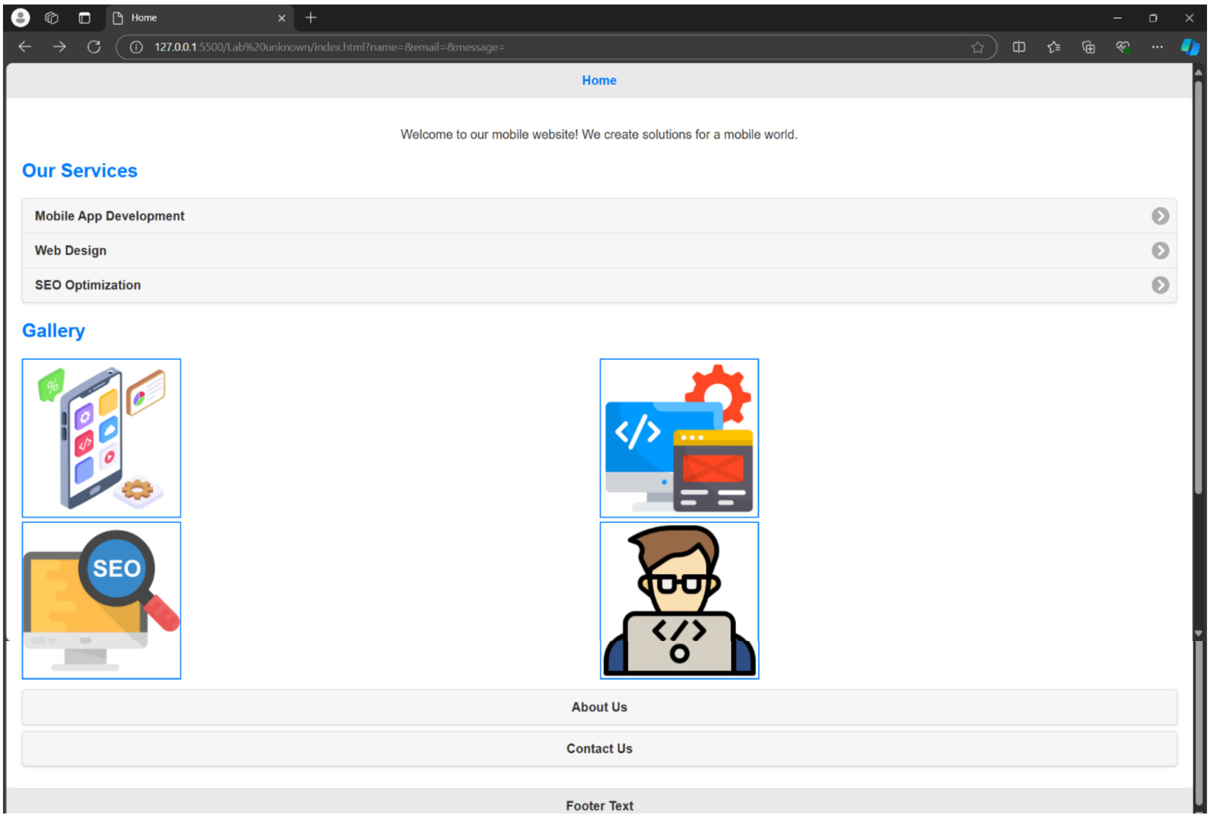
```html
    <form>
      <label for="name">Name:</label>
      <input type="text" name="name" id="name" placeholder="Your Name">

      <label for="email">Email:</label>
      <input type="email" name="email" id="email" placeholder="you@example.com">

      <label for="message">Message:</label>
      <textarea name="message" id="message" placeholder="Your message here"></textarea>

      <input type="submit" value="Send Message" class="ui-btn ui-corner-all ui-shadow">
    </form>
    <a href="#home" class="ui-btn ui-corner-all ui-shadow">Back to Home</a>
  </div>

  <div data-role="footer">
    <h4>Footer Text</h4>
  </div>
</div>

<!-- Additional Pages for Services -->
<div data-role="page" id="service1">
  <div data-role="header">
    <h1>Mobile App Development</h1>
  </div>
  <div role="main" class="ui-content">
    <p>We build powerful, intuitive, and user-friendly mobile apps for all platforms.</p>
    <a href="#home" class="ui-btn ui-corner-all ui-shadow">Back to Home</a>
  </div>
</div>
```

```
<div data-role="page" id="service2">
 <div data-role="header">
  <h1>Web Design</h1>
 </div>
 <div role="main" class="ui-content">
  <p>Our web design services offer responsive and creative solutions that enhance your brand’s digital presence.</p>
  <a href="#home" class="ui-btn ui-corner-all ui-shadow">Back to Home</a>
 </div>
</div>

<div data-role="page" id="service3">
 <div data-role="header">
  <h1>SEO Optimization</h1>
 </div>
 <div role="main" class="ui-content">
  <p>We optimize your website to rank higher on search engines, increasing visibility and traffic.</p>
  <a href="#home" class="ui-btn ui-corner-all ui-shadow">Back to Home</a>
 </div>
</div>

</body>
</html>
```

- styles.css

```
body {
   background-color: #f0f8ff;
}

.ui-header,
.ui-footer {
   background-color: #007bff;
   color: white;
}

.ui-btn {
   background-color: #28a745;
   color: white;
   border-color: #28a745;
}

.ui-btn:hover {
   background-color: #218838;
}

.ui-grid-a img {
   border: 2px solid #007bff;
   width: 200px;
   height: 200px;
}
.ui-content {
   background-color: #ffffff;
   padding: 20px;
   border-radius: 10px;
}
h1,
h2 {
   color: #007bff;
}
input[type="text"],
input[type="email"],
textarea {
   border: 1px solid #007bff;
   border-radius: 5px;
   padding: 10px;
}
#intro_text{
   text-align: center;
}
```

OUTPUT:



Home
127.0.0.1:5500/Lab%20unknown/index.html?name=&email=&message=

Home

Welcome to our mobile website! We create solutions for a mobile world.

Our Services

- Mobile App Development
- Web Design
- SEO Optimization

Gallery

About Us

Contact Us

Footer Text