Assignment 1

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

Dashboard.html

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Ecommerce Website</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
sRIl4kxILFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkuIAvNWLzeN8tE5YBujZqJLB"
crossorigin="anonymous">
  <link rel="stylesheet" href="dashboard.css">
</head>

<body>
  <nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid">
      <a class="navbar-brand" href="#"
        style="background-color: black; padding-left: 20px; padding-right: 20px;">Rushi's
Shop</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
        data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false"
        aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Admin Panel</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Manage</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Products</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">About</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Contact</a>
          </li>
```

Assignment 1

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

Dashboard.html

```
            <form class="d-flex" role="search">
              <input class="form-control me-2" type="search" placeholder="Search" aria-
label="Search" />
              <button class="btn btn-outline-success" type="submit">Search</button>
            </form>
        </div>
      </div>
    </nav>

    <div class="card-container">
      <div class="card">
        <img src="https://rukminim2.flixcart.com/image/240/240/xif0q/monitor/e/m/s/sa272-
p1-full-hd-27-2025-um-hs2si-101-acer-original-imahey8jdh87j8kg.jpeg?q=60"
          class="card-img-top" alt="...">
        <div class="card-body">
          <h5 class="card-title">Acer Super-Slim 144Hz Display Screen</h5>
          <p class="card-text">Some quick example text to build on the card title and make up
the bulk of the
            card's content.</p>
          <a href="#" class="btn btn-primary">Buy Now</a>
        </div>
      </div>
      <div class="card">
        <img src="https://rukminim2.flixcart.com/image/240/240/xif0q/monitor/w/i/f/-
original-imahagy2cefszqgy.jpeg?q=60"
          class="card-img-top" alt="...">
        <div class="card-body">
          <h5 class="card-title">Helix Ultra-Eye-Saver FHD Display Screen</h5>
          <p class="card-text">Some quick example text to build on the card title and make up
the bulk of the
            card's content.</p>
          <a href="#" class="btn btn-primary">Buy Now</a>
        </div>
      </div>
      <div class="card">
        <img src="https://rukminim2.flixcart.com/image/312/312/xif0q/monitor/l/r/k/-
original-imagwdpyagcv2zz6.jpeg?q=70"
          class="card-img-top" alt="...">
        <div class="card-body">
          <h5 class="card-title">Acer FreeSync 100Hz Super Glare Display Screen</h5>
          <p class="card-text">Some quick example text to build on the card title and make up
the bulk of the
            card’s content.</p>
          <a href="#" class="btn btn-primary">Buy Now</a>
        </div>
      </div>
```

Assignment 1

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

Dashboard.html

```
    </div>
</body>

</html>
```

Dashboard.css

```
body {
    background-color: #f8f9fa;
}

.navbar-brand {
    color: white !important;
    font-weight: bold;
}

.card {
    border: none;
    border-radius: 10px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.card:hover {
    transform: translateY(-5px);
    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.15);
}

.card-img-top {
    height: 200px;
    object-fit: contain;
    padding: 1rem;
}

.card-container {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
    gap: 20px;
    padding: 20px;
    max-width: 1200px;
    margin: 0 auto;
}
```

Assignment 1

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

Dashboard.css

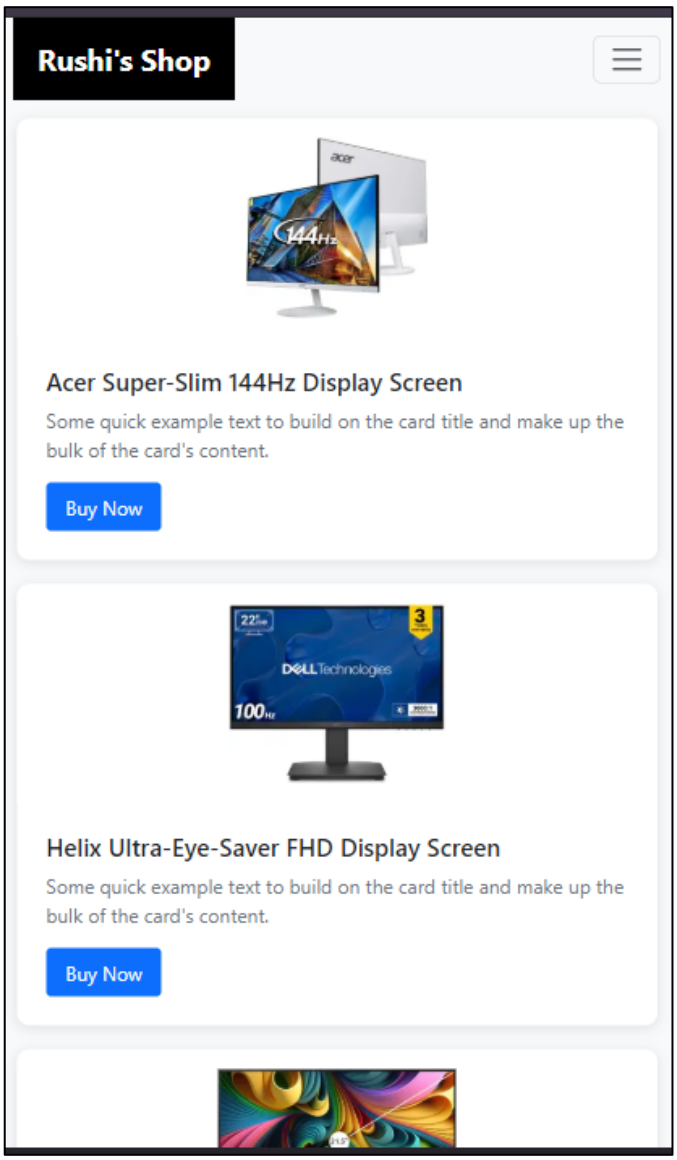
```
@media (max-width: 576px) {
   .card-container {
      grid-template-columns: 1fr;
   }

   .card-img-top {
      height: 180px;
   }
}

.nav-item:hover {
   background-color: black;
   border-radius: 5px;
}
```

Assignment 1

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

Output

Mobile View:

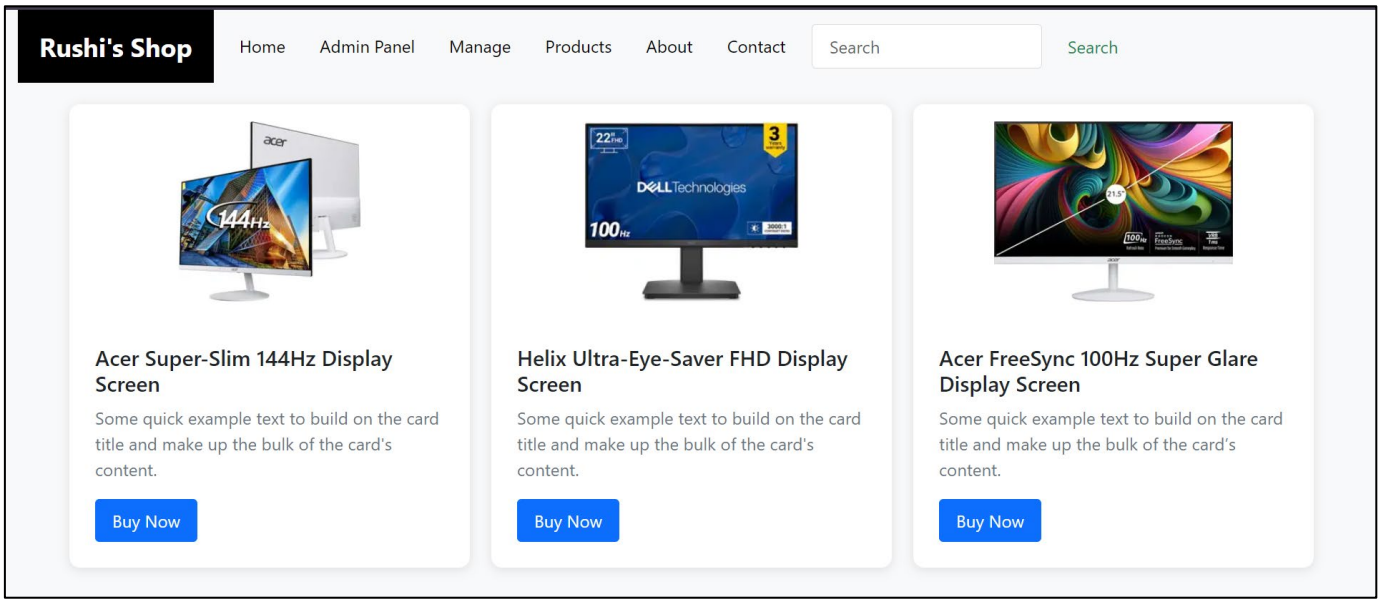


Assignment 1

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

Output

Desktop View:



Assignment 2

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

index.html

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>User Registration</title>
  <link rel="stylesheet" href="styles.css" />
</head>

<body>
  <main class="container">
    <h1>User Registration</h1>

    <form id="regForm">
      <div class="row">
        <label for="firstName">First name</label>
        <input id="firstName" name="firstName" type="text" required />
      </div>

      <div class="row">
        <label for="lastName">Last name</label>
        <input id="lastName" name="lastName" type="text" required />
      </div>

      <div class="row">
        <label for="email">Email</label>
        <input id="email" name="email" type="email" required />
      </div>

      <div class="row">
        <label for="password">Password</label>
        <input id="password" name="password" type="password" minlength="6" required />
      </div>

      <div class="row">
        <label for="phone">Phone</label>
        <input id="phone" name="phone" type="tel" />
      </div>

      <div class="row">
        <label for="gender">Gender</label>
        <select id="gender" name="gender">
          <option value="">-- Select --</option>
          <option>Male</option>
          <option>Female</option>
          <option>Other</option>
```

Assignment 2

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

index.html

```
          </select>
        </div>

        <div class="actions">
          <button type="submit">Register (Save locally)</button>
          <button type="button" id="viewList">View Registrations</button>
        </div>

        <div id="message" aria-live="polite"></div>
      </form>
    </main>

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

</html>
```

list.html

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Registrations List</title>
    <link rel="stylesheet" href="styles.css" />
</head>

<body>
    <main class="container">
        <h1>Registered Users</h1>

        <div class="controls">
            <button id="clear">Clear All</button>
            <button id="export">Export JSON</button>
            <a id="backLink" href="index.html">← Back to form</a>
        </div>

        <div id="listArea"></div>
    </main>

    <script>
        function renderList() {
            const area = document.getElementById('listArea');
```

Assignment 2

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

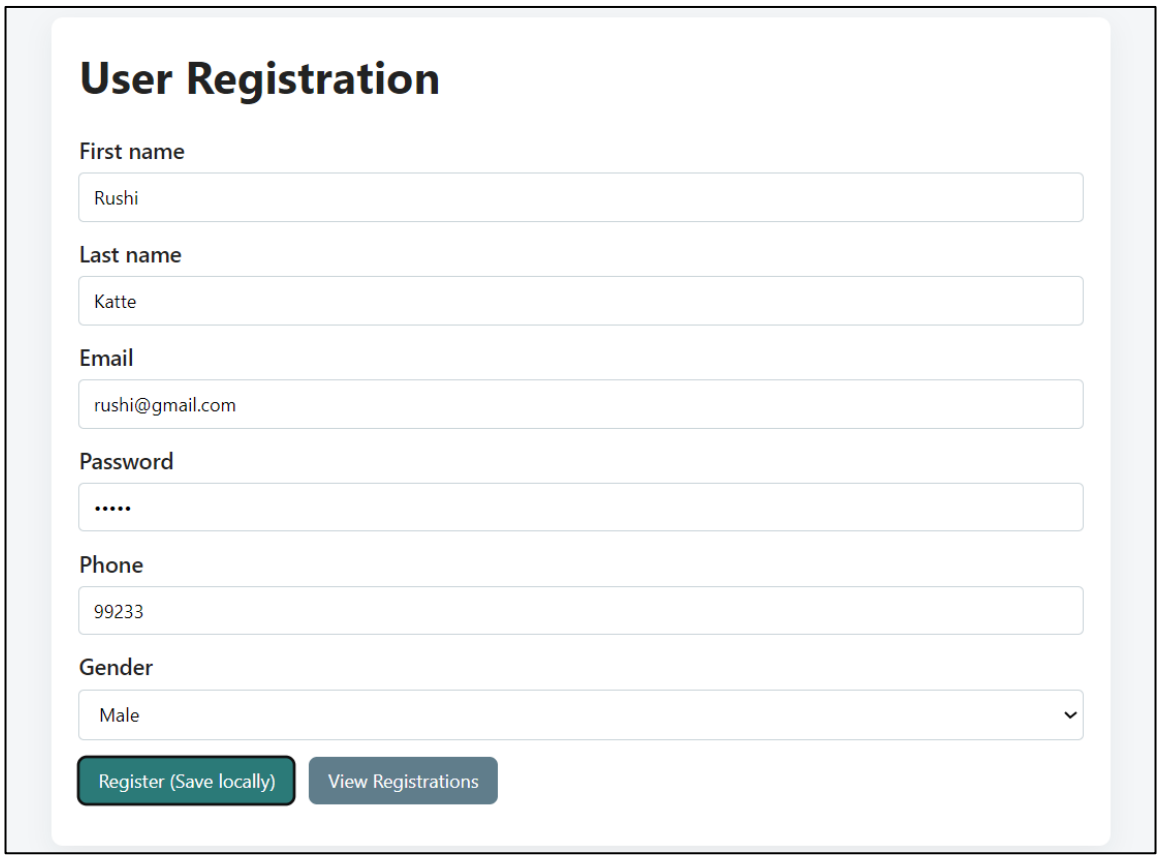
list.html

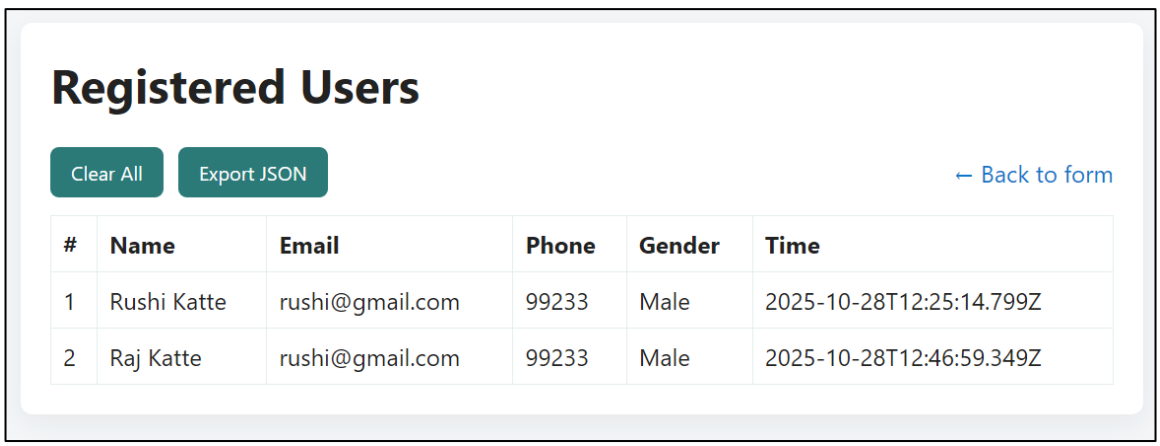
```
const raw = localStorage.getItem('registrations');
        const regs = raw ? JSON.parse(raw) : [];
        if (!regs.length) {
           area.innerHTML = '<p>No registrations found.</p>';
           return;
        }
        let html =
'<table><thead><tr><th>#</th><th>Name</th><th>Email</th><th>Phone</th><th>Gender</
th><th>Time</th></tr></thead><tbody>';
        regs.forEach((r, i) => {
           html += `<tr><td>${i + 1}</td><td>${escapeHtml(r.firstName)}
${escapeHtml(r.lastName)}</td><td>${escapeHtml(r.email)}</td><td>${escapeHtml(r.phone
|| '')}</td><td>${escapeHtml(r.gender || '')}</td><td>${escapeHtml(r.time)}</td></tr>`;
        });
        html += '</tbody></table>';
        area.innerHTML = html;
     }

     function escapeHtml(s) {
        if (!s) return '';
        return String(s).replace(/&/g, '&amp;').replace(/</g, '&lt;').replace(/>/g,
'&gt;').replace(/"/g, '&quot;');
     }

     document.getElementById('clear').addEventListener('click', () => {
        if (!confirm('Clear all registrations from localStorage?')) return;
        localStorage.removeItem('registrations');
        renderList();
     });

     document.getElementById('export').addEventListener('click', () => {
        const raw = localStorage.getItem('registrations') || '[]';
        const blob = new Blob([raw], { type: 'application/json' });
        const url = URL.createObjectURL(blob);
        const a = document.createElement('a');
        a.href = url;
        a.download = 'registrations.json';
        document.body.appendChild(a);
        a.click();
        a.remove();
        URL.revokeObjectURL(url);
     });
     renderList();
  </script>
</body>
</html>
```

Assignment 2

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

Output





Assignment 3A

Name: Rushikesh Katte

Roll No: 23572

Sub : Web Technology

Open the git bash and check if it is installed properly or not using git version command.

```
rushi@Admin MINGW64 ~
$ git --version
git version 2.47.0.windows.2
```

If the output comes like this then your git is installed successfully.

Create Repository using git commands –

Git Set up commands –

1. The first thing you should do when you install Git is to set your user name and email address. This is important because every Git commit uses this information, and it's immutably baked into the commits you start creating:

```
$ git config --global user.name "Yuvraj Pardeshi"

$ git config --global user.email yuvrajpardeshi75@gmail.com
```

```
rushi@Admin MINGW64 ~
$ git config --global user.name "Rushikesh Katte"

rushi@Admin MINGW64 ~
$ git config --global user.email rushikeshrkatte2005@gmail.com
```

Create a new project –

2. Create an folder using command mkdir <dir name>
   and to enter into this create folder use command cd <dir name>

```
rushi@Admin MINGW64 ~
$ mkdir samplerepo

rushi@Admin MINGW64 ~
$ cd samplerepo

rushi@Admin MINGW64 ~/samplerepo
```

Create Repository –

3. To create your current folder into repository use command git init

```
rushi@Admin MINGW64 ~/samplerepo
$ git init
Initialized empty Git repository in C:/Users/rushi/samplerepo/.git/
```

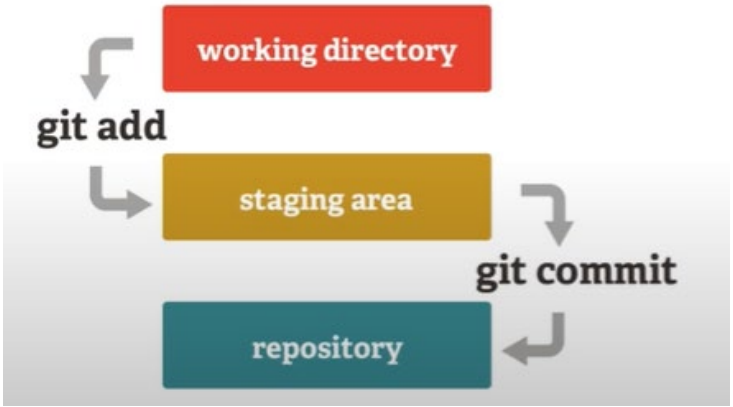
as shown in above figure your empty repo is created on the path shown in the figure.

Adding files to Repository -

4. Know here comes the concept of staging area and committing a file.

Staging area – Initially when you add files to the git folder they are not tracked by git, these files are also referred to as “untracked files”. Staging area is files that are going to be a part of next commit, which lets the git know what changes in the file are going to occur for the next commit. The command use to add the file to staging area is (git add).

Committing file - The git commit command captures a snapshot of the project's currently staged changes. Committed snapshots can be thought of as “safe” versions of a project—Git will never change them unless you explicitly ask it to. Prior to the execution of git commit, The git add command is used to promote or 'stage' changes to the project that will be stored in a commit. These two commands “git commit” and “git add” are two of the most frequently used.



```
rushi@Admin MINGW64 ~/samplerepo (master)
$ git add index.html

rushi@Admin MINGW64 ~/samplerepo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .add.cpp.swp
```

(Added file to staging area)

```
rushi@Admin MINGW64 ~/samplerepo (master)
$ git commit -m "adding index.html"
[master (root-commit) 88bc4ea] adding index.html
 1 file changed, 8 insertions(+)
 create mode 100644 index.html
```

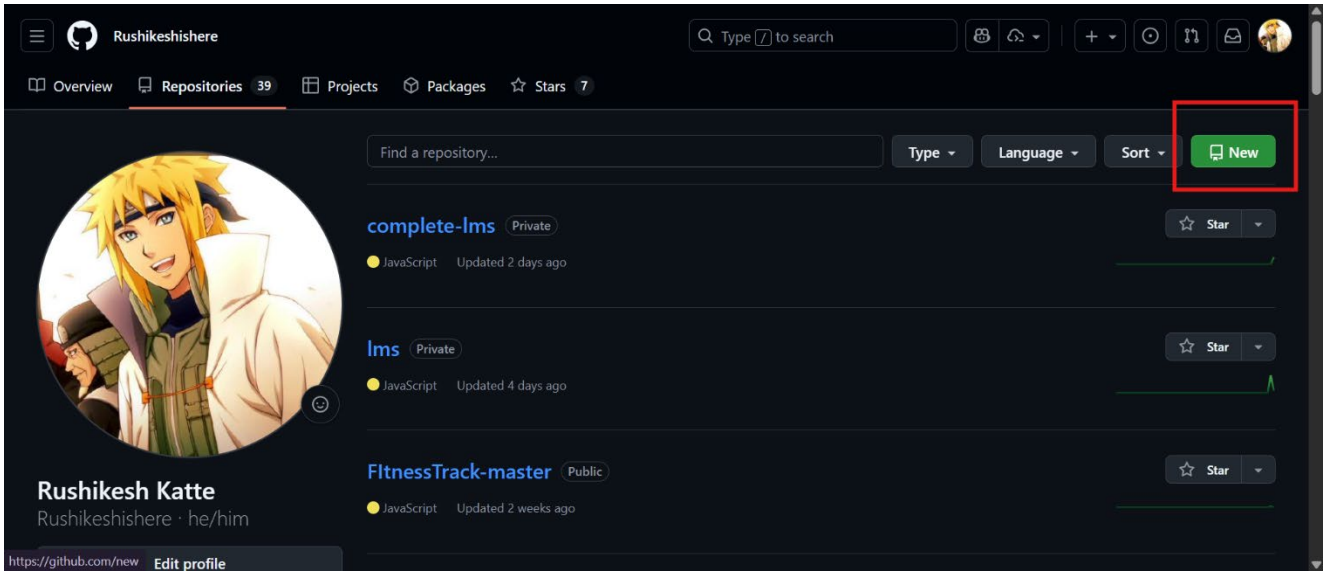
(Committed the file)

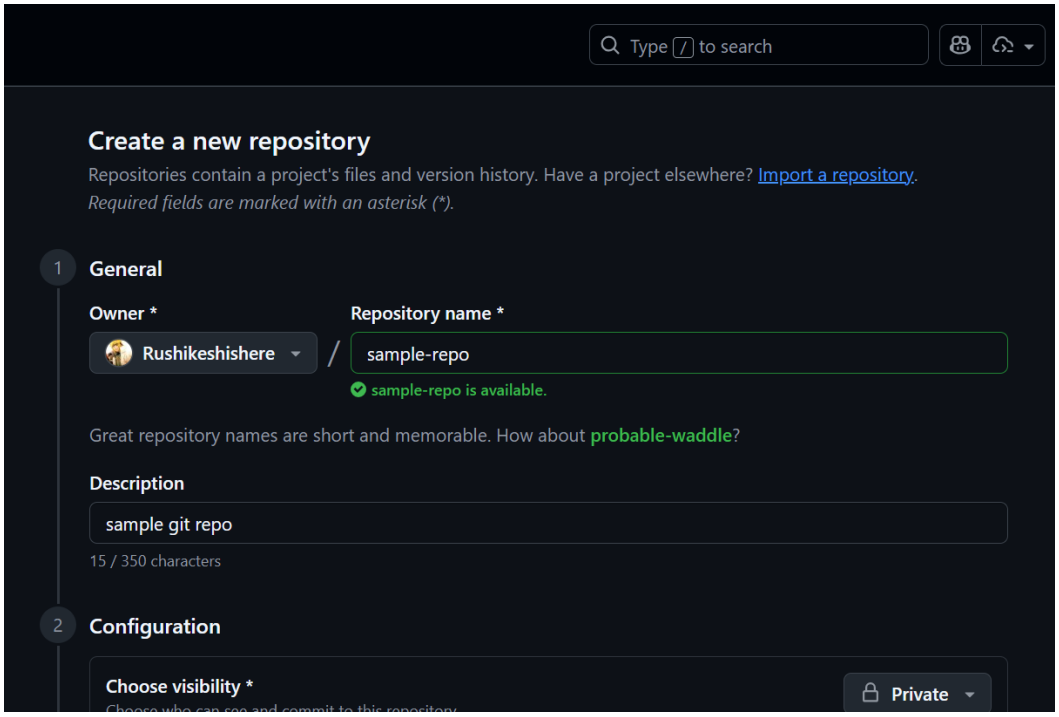
To check the commit status use git log command –

```
rushi@Admin MINGW64 ~/samplerepo (master)
$ git log
commit 88bc4eac5d08fb98bf51af6210c106ef4703cd03 (HEAD -> master)
Author: Rushikesh Katte <rushikeshrkatte2005@gmail.com>
Date:   Tue Oct 28 18:55:27 2025 +0530

    adding index.html
```

Pushing your files to GitHub –

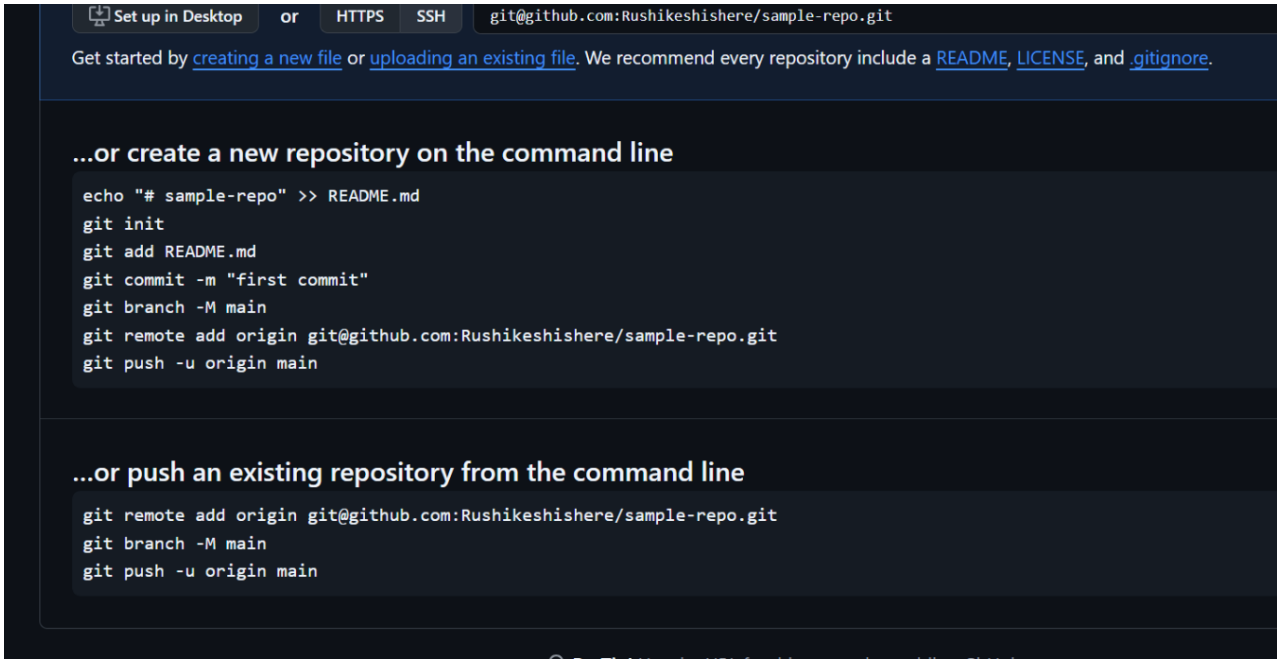
5 **.** Login your github account and create a new repository.





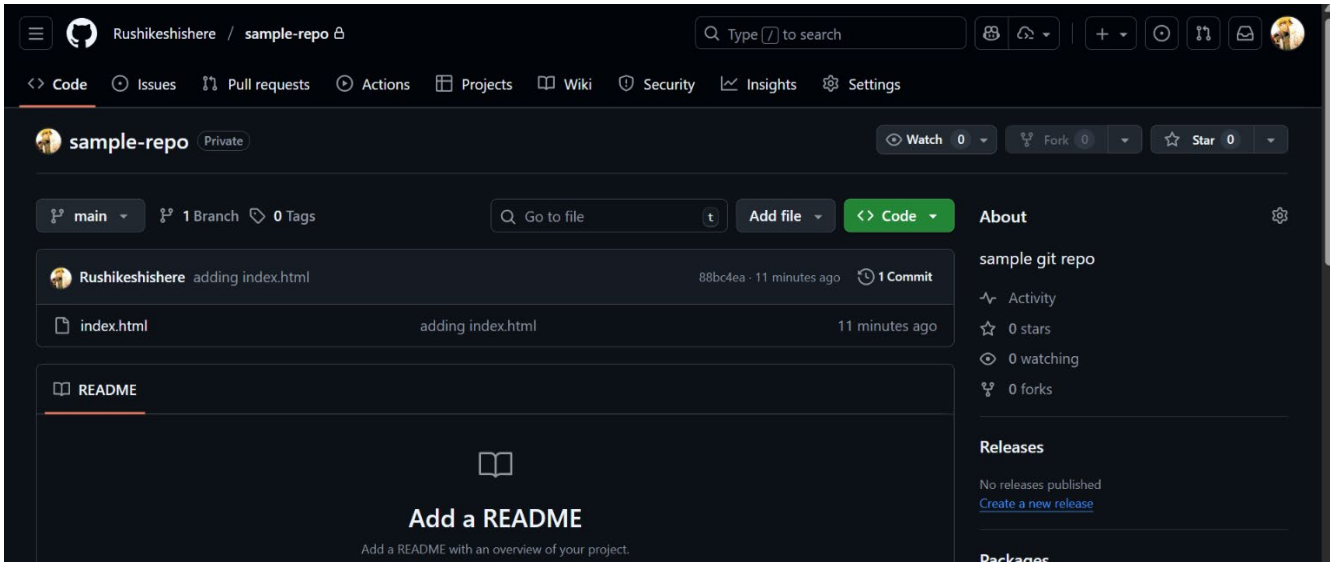
After creating the repository in github, there will the option to create your repo from scratch or if there is some local repo already created then push it. We are going to push our local repo created using git.

Pushing the local repo –



(Use the above commands to push your local repo)

```
rushi@Admin MINGW64 ~/samplerepo (main)
$ git remote remove origin

rushi@Admin MINGW64 ~/samplerepo (main)
$ git remote add origin https://github.com/Rushikeshishere/sample-repo.git

rushi@Admin MINGW64 ~/samplerepo (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 304 bytes | 304.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Rushikeshishere/sample-repo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

rushi@Admin MINGW64 ~/samplerepo (main)
$
```



Hence we have create your repo on github and added file to the repo successfully

Assignment 3B

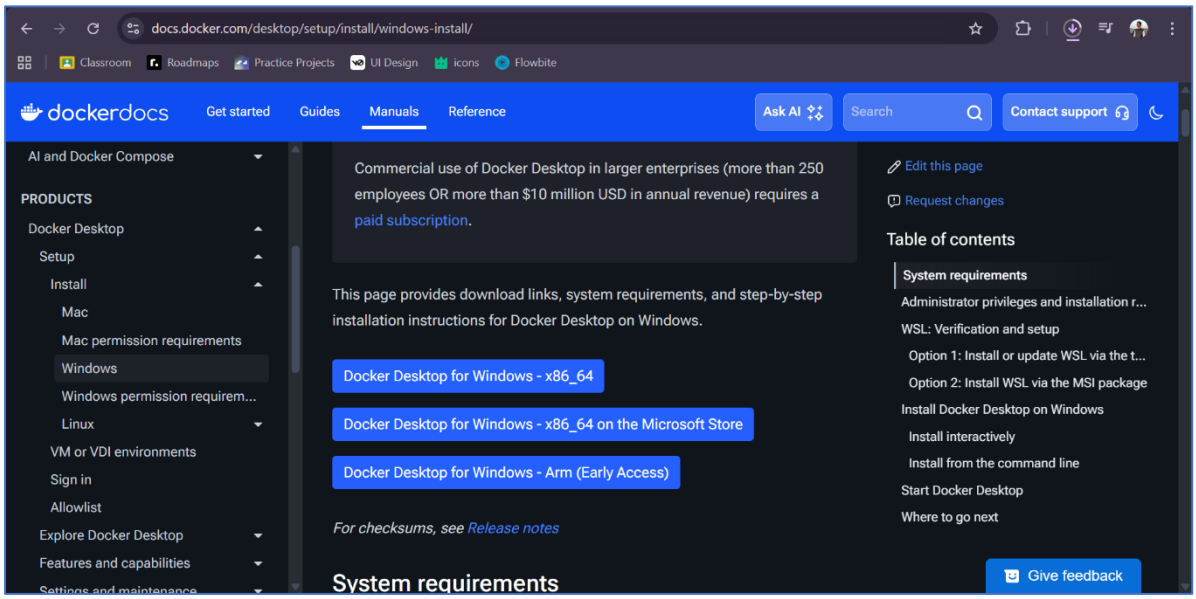
Name: Rushikesh Katte

Roll Number: 23572

Subject: Web Technology

1. Installing the docker

https://docs.docker.com/desktop/install/windows-install/



Run the executable file.

2. Install WSL distribution from :

https://learn.microsoft.com/en-us/windows/wsl/install-manual

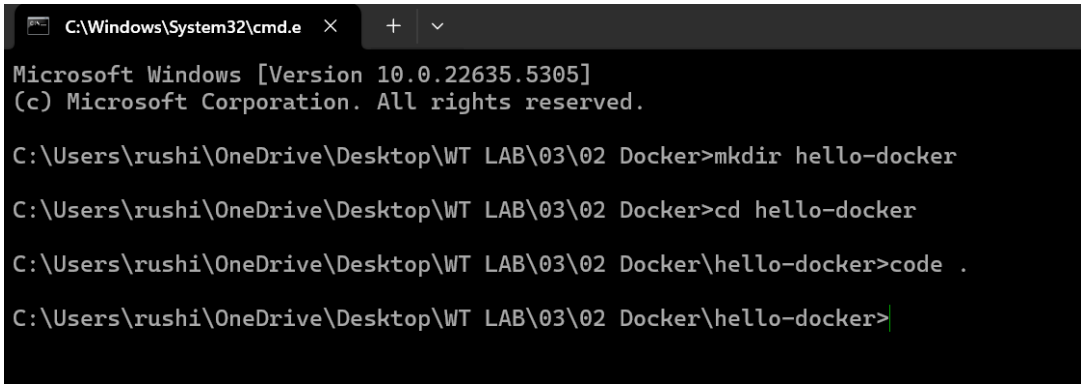
```
PS C:\Users\rushi> wsl --install
Downloading: Ubuntu
Installing: Ubuntu
Distribution successfully installed. It can be launched via 'wsl.exe -d Ubuntu'
Launching Ubuntu...
Provisioning the new WSL instance Ubuntu
This might take a while...
Create a default Unix user account: rushi
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

rushi@Admin:/mnt/c/Users/rushi$
```

3. Restart the system.

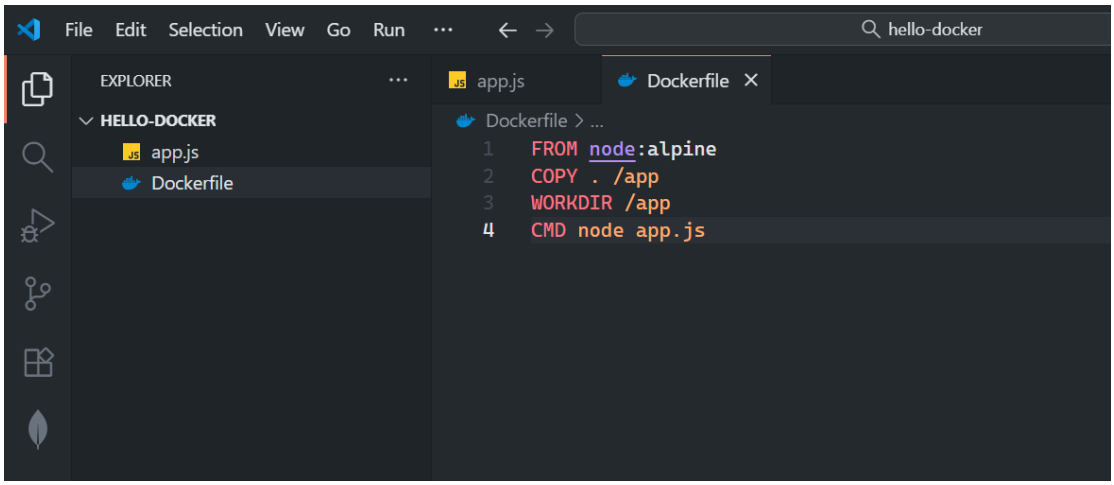
4. On command prompt run the command docker version. It should look like :

```
Administrator: C:\Windows\system32\cmd.exe
C:\Windows\System32>docker version
Client:
 Version:           28.5.1
 API version:       1.51
 Go version:        go1.24.8
 Git commit:        e180ab8
 Built:             Wed Oct  8 12:19:16 2025
 OS/Arch:           windows/amd64
 Context:           desktop-linux

Server: Docker Desktop 4.49.0 (208700)
 Engine:
  Version:          28.5.1
  API version:      1.51 (minimum version 1.24)
  Go version:       go1.24.8
  Git commit:       f8215cc
  Built:            Wed Oct  8 12:17:24 2025
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.7.27
  GitCommit:        05044ec0a9a75232cad458027ca83437aae3f4da
 runc:
  Version:          1.2.5
  GitCommit:        v1.2.5-0-g59923ef
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0

C:\Windows\System32>
```

5.Run the following commands :



6. The Visual Studio Code will open. Inside VS code open folder docker-hello and create file app.js and Docker file:



7. Inside app.js file type the code:

console.log("Hello docker");

8. Inside Dockerfile type the code :

FROM node:alpine

COPY . /app

WORKDIR /app

CMD node app.js

9. In the command prompt type the command :

   docker build -t hello-docker .

```
Microsoft Windows [Version 10.0.22635.5305]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rushi\OneDrive\Desktop\WT LAB\03\02 Docker\hello-docker>docker build -t hello-docker .
[+] Building 46.6s (9/9) FINISHED                                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                0.1s
 => => transferring dockerfile: 97B                                                                                 0.0s
 => [internal] load metadata for docker.io/library/node:alpine                                                      8.0s
 => [auth] library/node:pull token for registry-1.docker.io                                                         0.0s
 => [internal] load .dockerignore                                                                                   0.1s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load build context                                                                                   0.1s
 => => transferring context: 162B                                                                                   0.1s
 => [1/3] FROM docker.io/library/node:alpine@sha256:bce79c648e05d584ad9ae2b45ed663ae6f07ebfa9e5fe6f5b7f0165aca  37.1s
 => => resolve docker.io/library/node:alpine@sha256:bce79c648e05d584ad9ae2b45ed663ae6f07ebfa9e5fe6f5b7f0165aca0  0.0s
 => => sha256:d9f0123823b504c41373c028b427fcf56cf338c454bf4291afc415de5e070464 443B / 443B                          0.7s
 => => sha256:4350616372df2de9f6b15ad5b0bfc771662215b9192defaeb7d94a96978bb0ec 1.26MB / 1.26MB                      3.0s
 => => sha256:6d0bdc936f33e7a2e7faae4e44192a62c33e1c9f1fe253e3f046afb31737eeb5 54.99MB / 54.99MB                   35.3s
 => => sha256:2d35ebdb57d9971fea0cac1582aa78935adf8058b2cc32db163c98822e5dfa1b 3.80MB / 3.80MB                      5.9s
 => => extracting sha256:2d35ebdb57d9971fea0cac1582aa78935adf8058b2cc32db163c98822e5dfa1b                            0.3s
 => => extracting sha256:6d0bdc936f33e7a2e7faae4e44192a62c33e1c9f1fe253e3f046afb31737eeb5                            1.5s
 => => extracting sha256:4350616372df2de9f6b15ad5b0bfc771662215b9192defaeb7d94a96978bb0ec                            0.1s
 => => extracting sha256:d9f0123823b504c41373c028b427fcf56cf338c454bf4291afc415de5e070464                            0.0s
 => [2/3] COPY . /app                                                                                               0.1s
 => [3/3] WORKDIR /app                                                                                              0.0s
 => exporting to image                                                                                              0.3s
 => => exporting layers                                                                                             0.1s
 => => exporting manifest sha256:b1d40101763fe38cc935cf6e91e414c67c59c0398bb26ea152cc43fe957d0e15                   0.0s
 => => exporting config sha256:a2435c2e6ed7428e29a42dbbc4234f63338870f636f3a9fde0e2f83146868dea                     0.0s
 => => exporting attestation manifest sha256:a7887ec84e67f656af30545853bd555cfe61204aedf19105598aae8588906805       0.0s
 => => exporting manifest list sha256:c078fc605168782926cc0431bdbfae95881fa8cb81d9b014a5515c23101e82a7              0.0s
```

10. Then type the command :

    docker image ls

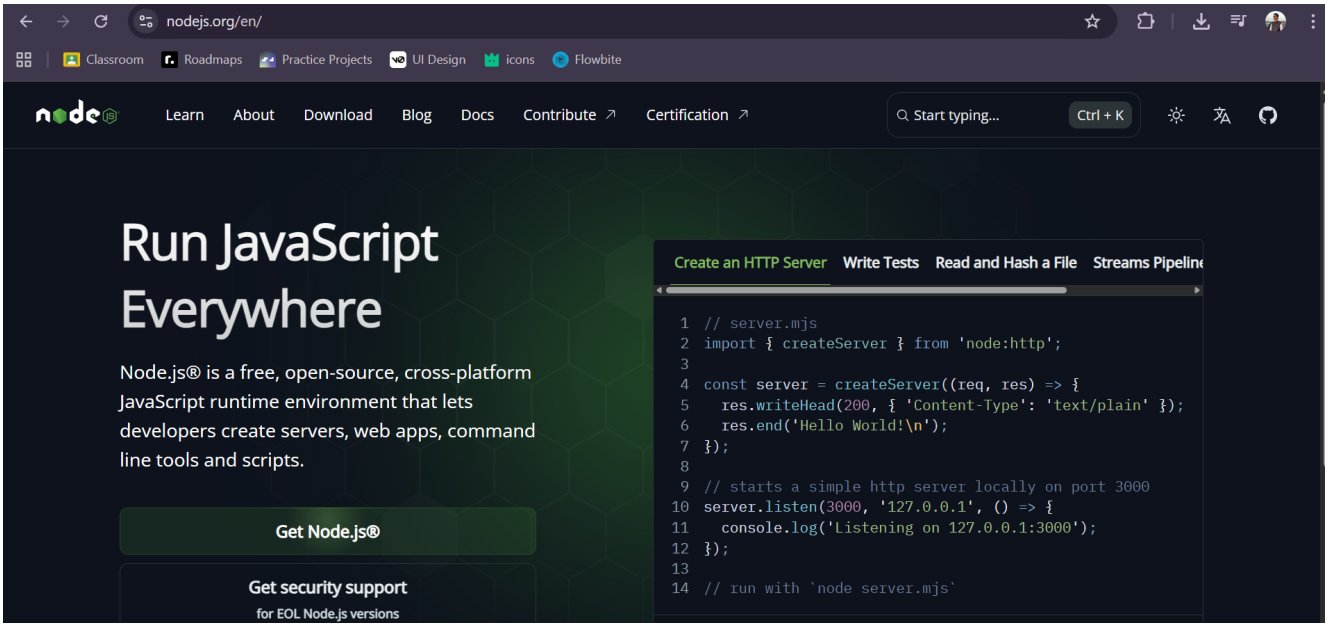
11. You should get the output image :

```
C:\Users\rushi\OneDrive\Desktop\WT LAB\03\02 Docker\hello-docker>docker image ls
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
hello-docker   latest    c078fc605168   8 seconds ago    242MB
```

Assignment 3C

Name: Rushikesh Katte
Roll Number: 23572
Subject: Web Technology

Firstly, lets install Node.js from https://nodejs.org/

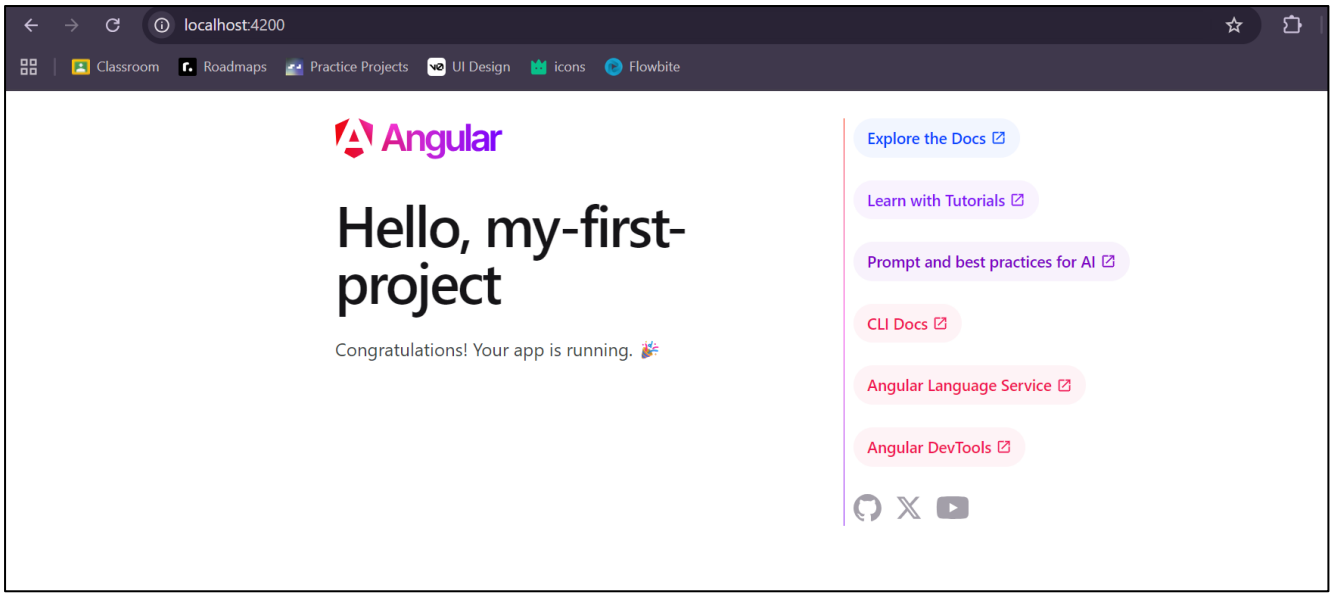


Follow the steps to complete the download.

1. Create Angular App

```
C:\Users\rushi\OneDrive\Desktop\WT LAB\03\03 Angular\my-first-project>ng new user-auth-app
✓ Which stylesheet format would you like to use? CSS             [ https://developer.mozilla.org/docs/Web/CSS
                ]
✓ Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? No
✓ Do you want to create a 'zoneless' application without zone.js? Yes
✓ Which AI tools do you want to configure with Angular best practices? https://angular.dev/ai/develop-with-ai
None
CREATE user-auth-app/angular.json (2337 bytes)
CREATE user-auth-app/package.json (1110 bytes)
CREATE user-auth-app/README.md (1533 bytes)
CREATE user-auth-app/tsconfig.json (1026 bytes)
CREATE user-auth-app/.editorconfig (331 bytes)
CREATE user-auth-app/.gitignore (647 bytes)
CREATE user-auth-app/tsconfig.app.json (444 bytes)
CREATE user-auth-app/tsconfig.spec.json (422 bytes)
CREATE user-auth-app/.vscode/extensions.json (134 bytes)
CREATE user-auth-app/.vscode/launch.json (490 bytes)
CREATE user-auth-app/.vscode/tasks.json (980 bytes)
CREATE user-auth-app/src/main.ts (228 bytes)
CREATE user-auth-app/src/index.html (310 bytes)
CREATE user-auth-app/src/styles.css (81 bytes)
CREATE user-auth-app/src/app/app.spec.ts (812 bytes)
CREATE user-auth-app/src/app/app.ts (307 bytes)
CREATE user-auth-app/src/app/app.css (0 bytes)
CREATE user-auth-app/src/app/app.html (20464 bytes)
CREATE user-auth-app/src/app/app.config.ts (395 bytes)
CREATE user-auth-app/src/app/app.routes.ts (80 bytes)
CREATE user-auth-app/public/favicon.ico (15086 bytes)
✓ Packages installed successfully.
warning: in the working copy of 'package-lock.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
    Successfully initialized git.
```

```
C:\Users\rushi\OneDrive\Desktop\WT LAB\03\03 Angular\my-first-project>cd user-auth-app

C:\Users\rushi\OneDrive\Desktop\WT LAB\03\03 Angular\my-first-project\user-auth-app>ng serve -o

Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.dev/cli/analytics.

   No
Global setting: enabled
Local setting: disabled
Effective status: disabled
Initial chunk files | Names         | Raw size
main.js             | main          | 48.48 kB |
styles.css          | styles        | 95 bytes |

                    | Initial total | 48.57 kB

Application bundle generation complete. [2.308 seconds] - 2025-10-28T16:38:18.195Z

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
  → Local:   http://localhost:4200/
  → press h + enter to show help
```



Angular User Authentication Application

Project Overview

This is a comprehensive **User Authentication Application** built with **Angular 20** featuring modern Angular capabilities including standalone components, signals, and control flow syntax. The application provides complete user authentication functionality with registration, login, and profile management.

Project Architecture

Technology Stack
- **Frontend Framework**: Angular 20.3.0
- **Language**: TypeScript 5.9.2
- **Styling**: CSS3 with modern features
- **State Management**: Angular Signals
- **Data Persistence**: Browser localStorage
- **Build Tool**: Angular CLI 20.3.7

Project Structure

```
user-auth-app/
├── src/
│   ├── app/
│   │   ├── app.ts              # Main application component
│   │   ├── app.html            # Application template
│   │   ├── app.css             # Global styles
│   │   ├── app.config.ts       # Application configuration
│   │   ├── app.routes.ts       # Routing configuration
│   │   ├── models/
│   │   │   └── user.model.ts   # User interface definition
│   │   ├── services/
│   │   │   └── auth.ts         # Authentication service
│   │   ├── login/
│   │   │   ├── login.ts        # Login component
│   │   │   ├── login.html      # Login template
│   │   │   └── login.css       # Login styles
│   │   ├── register/
│   │   │   ├── register.ts     # Registration component
│   │   │   ├── register.html   # Registration template
│   │   │   └── register.css    # Registration styles
│   │   └── profile/
│   │       ├── profile.ts      # Profile component
│   │       ├── profile.html    # Profile template
│   │       └── profile.css     # Profile styles
│   ├── main.ts                 # Application bootstrap
│   ├── index.html              # Main HTML file
│   └── styles.css              # Global styles
├── package.json                # Dependencies and scripts
├── angular.json                # Angular configuration
├── tsconfig.json               # TypeScript configuration
└── tsconfig.app.json           # TypeScript app configuration
```

🔧 Complete Source Code

1. Package Configuration

`package.json`
```json
{
  "name": "user-auth-app",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "watch": "ng build --watch --configuration development"
  },
  "private": true,
  "dependencies": {
    "@angular/common": "^20.3.0",
    "@angular/compiler": "^20.3.0",
    "@angular/core": "^20.3.0",
    "@angular/forms": "^20.3.0",
    "@angular/platform-browser": "^20.3.0",
    "@angular/router": "^20.3.0",
    "rxjs": "~7.8.0",
```

```
    "tslib": "^2.3.0"
  },
  "devDependencies": {
    "@angular/build": "^20.3.7",
    "@angular/cli": "^20.3.7",
    "@angular/compiler-cli": "^20.3.0",
    "typescript": "~5.9.2"
  }
}
```

`angular.json`

```json
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "user-auth-app": {
      "projectType": "application",
      "schematics": {},
      "root": "",
      "sourceRoot": "src",
      "prefix": "app",
      "architect": {
        "build": {
          "builder": "@angular/build:application",
          "options": {
            "browser": "src/main.ts",
            "tsConfig": "tsconfig.app.json",
            "assets": [
              {
                "glob": "**/*",
                "input": "public"
              }
            ],
            "styles": [
              "src/styles.css"
            ]
          },
          "configurations": {
            "production": {
              "budgets": [
                {
                  "type": "initial",
                  "maximumWarning": "500kB",
                  "maximumError": "1MB"
                },
                {
                  "type": "anyComponentStyle",
                  "maximumWarning": "4kB",
                  "maximumError": "8kB"
                }
              ],
              "outputHashing": "all"
            },
            "development": {
              "optimization": false,
```

```
            "extractLicenses": false,
            "sourceMap": true
          }
        },
        "defaultConfiguration": "production"
      },
      "serve": {
        "builder": "@angular/build:dev-server",
        "configurations": {
          "production": {
            "buildTarget": "user-auth-app:build:production"
          },
          "development": {
            "buildTarget": "user-auth-app:build:development"
          }
        },
        "defaultConfiguration": "development"
      },
      "extract-i18n": {
        "builder": "@angular/build:extract-i18n"
      }
    }
  }
 },
 "cli": {
  "analytics": false
 }
}
```

2. Application Bootstrap

`src/main.ts`
```typescript
import { bootstrapApplication } from '@angular/platform-browser';
import { appConfig } from './app/app.config';
import { App } from './app/app';

bootstrapApplication(App, appConfig)
 .catch((err) => console.error(err));
```

`src/index.html`
```html
<!doctype html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <title>UserAuthApp</title>
 <base href="/">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
 <app-root></app-root>
</body>
</html>
```

3. Application Configuration

`src/app/app.config.ts`
```typescript
import { ApplicationConfig, provideBrowserGlobalErrorListeners, provideZonelessChangeDetection } from
'@angular/core';
import { provideRouter } from '@angular/router';

import { routes } from './app.routes';

export const appConfig: ApplicationConfig = {
  providers: [
    provideBrowserGlobalErrorListeners(),
    provideZonelessChangeDetection(),
    provideRouter(routes)
  ]
};
```

`src/app/app.routes.ts`
```typescript
import { Routes } from '@angular/router';
import { Register } from './register/register';
import { Login } from './login/login';
import { Profile } from './profile/profile';

export const routes: Routes = [
    { path: '', redirectTo: '/login', pathMatch: 'full' },
    { path: 'register', component: Register },
    { path: 'login', component: Login },
    { path: 'profile', component: Profile },
    { path: '**', redirectTo: '/login' }
];
```

4. Main Application Component

`src/app/app.ts`
```typescript
import { Component, signal } from '@angular/core';
import { RouterOutlet } from '@angular/router';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet],
  templateUrl: './app.html',
  styleUrl: './app.css'
})
export class App {
  protected readonly title = signal('user-auth-app');
}
```

`src/app/app.html`
```html
<router-outlet></router-outlet>
```

`src/app/app.css`
```css
/* Empty - styles handled by individual components */
```

5. Data Models

`src/app/models/user.model.ts`
```typescript
export interface User {
  id?: number;
  username: string;
  email: string;
  password?: string;
  firstName?: string;
  lastName?: string;
  phone?: string;
}
```

6. Authentication Service

`src/app/services/auth.ts`
```typescript
import { Injectable, signal } from '@angular/core';
import { User } from '../models/user.model';

@Injectable({
 providedIn: 'root'
})
export class Auth {
 private users: User[] = [];
 private currentUser = signal<User | null>(null);
 private nextId = 1;

 constructor() {
  // Load users from localStorage
  const storedUsers = localStorage.getItem('users');
  if (storedUsers) {
   this.users = JSON.parse(storedUsers);
   this.nextId = Math.max(...this.users.map(u => u.id || 0), 0) + 1;
  }

  // Load current user from localStorage
  const storedCurrentUser = localStorage.getItem('currentUser');
  if (storedCurrentUser) {
   this.currentUser.set(JSON.parse(storedCurrentUser));
  }
 }

 // Register a new user
 register(user: User): { success: boolean; message: string } {
  // Check if user already exists
  const existingUser = this.users.find(
   u => u.username === user.username || u.email === user.email
  );
```

```
    if (existingUser) {
      return {
        success: false,
        message: 'Username or email already exists!'
      };
    }

    // Add new user
    const newUser: User = {
      ...user,
      id: this.nextId++
    };

    this.users.push(newUser);
    this.saveUsers();

    return {
      success: true,
      message: 'Registration successful! Please login.'
    };
  }

  // Login user
  login(username: string, password: string): { success: boolean; message: string } {
    const user = this.users.find(
      u => (u.username === username || u.email === username) && u.password === password
    );

    if (user) {
      // Don't store password in current user
      const { password: _, ...userWithoutPassword } = user;
      this.currentUser.set(userWithoutPassword);
      localStorage.setItem('currentUser', JSON.stringify(userWithoutPassword));

      return {
        success: true,
        message: 'Login successful!'
      };
    }

    return {
      success: false,
      message: 'Invalid username or password!'
    };
  }

  // Logout user
  logout(): void {
    this.currentUser.set(null);
    localStorage.removeItem('currentUser');
  }

  // Get current user
  getCurrentUser() {
    return this.currentUser;
  }

  // Check if user is logged in
```

```
  isLoggedIn(): boolean {
    return this.currentUser() !== null;
  }

  // Save users to localStorage
  private saveUsers(): void {
    localStorage.setItem('users', JSON.stringify(this.users));
  }
}
```

7. Login Component

`src/app/login/login.ts`
```typescript
import { Component, signal } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { Router, RouterLink } from '@angular/router';
import { Auth } from '../services/auth';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'app-login',
  imports: [FormsModule, CommonModule, RouterLink],
  templateUrl: './login.html',
  styleUrl: './login.css',
})
export class Login {
  username = '';
  password = '';
  message = signal('');
  messageType = signal<'success' | 'error'>('success');

  constructor(private authService: Auth, private router: Router) {
    // Redirect if already logged in
    if (this.authService.isLoggedIn()) {
      this.router.navigate(['/profile']);
    }
  }

  onSubmit(): void {
    if (!this.username || !this.password) {
      this.showMessage('Please enter username and password!', 'error');
      return;
    }

    const result = this.authService.login(this.username, this.password);

    if (result.success) {
      this.showMessage(result.message, 'success');
      setTimeout(() => {
        this.router.navigate(['/profile']);
      }, 1000);
    } else {
      this.showMessage(result.message, 'error');
    }
  }
```

```
  private showMessage(msg: string, type: 'success' | 'error'): void {
    this.message.set(msg);
    this.messageType.set(type);
    setTimeout(() => this.message.set(''), 5000);
  }
}
```

`src/app/login/login.html`
```html
<div class="login-container">
  <div class="login-card">
    <h2>Login</h2>

    @if (message()) {
    <div class="alert" [class.alert-success]="messageType() === 'success'"
      [class.alert-error]="messageType() === 'error'">
      {{ message() }}
    </div>
    }

    <form (ngSubmit)="onSubmit()" #loginForm="ngForm">
      <div class="form-group">
        <label for="username">Username or Email</label>
        <input type="text" id="username" name="username" [(ngModel)]="username" required class="form-
control"
          placeholder="Enter username or email">
      </div>

      <div class="form-group">
        <label for="password">Password</label>
        <input type="password" id="password" name="password" [(ngModel)]="password" required
          class="form-control" placeholder="Enter password">
      </div>

      <button type="submit" class="btn btn-primary" [disabled]="!loginForm.form.valid">
        Login
      </button>
    </form>

    <p class="register-link">
      Don't have an account? <a routerLink="/register">Register here</a>
    </p>
  </div>
</div>
```

`src/app/login/login.css`
```css
.login-container {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  padding: 20px;
}
```

```
.login-card {
    background: white;
    border-radius: 10px;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.2);
    padding: 40px;
    width: 100%;
    max-width: 400px;
}

.login-card h2 {
    text-align: center;
    color: #333;
    margin-bottom: 30px;
    font-size: 28px;
}

.alert {
    padding: 12px;
    border-radius: 5px;
    margin-bottom: 20px;
    text-align: center;
    font-weight: 500;
}

.alert-success {
    background-color: #d4edda;
    color: #155724;
    border: 1px solid #c3e6cb;
}

.alert-error {
    background-color: #f8d7da;
    color: #721c24;
    border: 1px solid #f5c6cb;
}

.form-group {
    margin-bottom: 20px;
}

.form-group label {
    display: block;
    margin-bottom: 5px;
    color: #555;
    font-weight: 600;
}

.form-control {
    width: 100%;
    padding: 12px;
    border: 1px solid #ddd;
    border-radius: 5px;
    font-size: 14px;
    transition: border-color 0.3s;
    box-sizing: border-box;
}

.form-control:focus {
```

```
  outline: none;
  border-color: #667eea;
}

.btn {
  width: 100%;
  padding: 12px;
  border: none;
  border-radius: 5px;
  font-size: 16px;
  font-weight: 600;
  cursor: pointer;
  transition: background-color 0.3s;
}

.btn-primary {
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  color: white;
}

.btn-primary:hover:not(:disabled) {
  opacity: 0.9;
}

.btn-primary:disabled {
  opacity: 0.6;
  cursor: not-allowed;
}

.register-link {
  text-align: center;
  margin-top: 20px;
  color: #666;
}

.register-link a {
  color: #667eea;
  text-decoration: none;
  font-weight: 600;
}

.register-link a:hover {
  text-decoration: underline;
}
```

8. Register Component

`src/app/register/register.ts`
```typescript
import { Component, signal } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { Router, RouterLink } from '@angular/router';
import { Auth } from '../services/auth';
import { User } from '../models/user.model';
import { CommonModule } from '@angular/common';

@Component({
```

```
  selector: 'app-register',
  imports: [FormsModule, CommonModule, RouterLink],
  templateUrl: './register.html',
  styleUrl: './register.css',
})
export class Register {
  user: User = {
    username: '',
    email: '',
    password: '',
    firstName: '',
    lastName: '',
    phone: ''
  };

  confirmPassword = '';
  message = signal('');
  messageType = signal<'success' | 'error'>('success');

  constructor(private authService: Auth, private router: Router) { }

  onSubmit(): void {
    // Validation
    if (!this.user.username || !this.user.email || !this.user.password) {
      this.showMessage('Please fill in all required fields!', 'error');
      return;
    }

    if (this.user.password !== this.confirmPassword) {
      this.showMessage('Passwords do not match!', 'error');
      return;
    }

    if (this.user.password.length < 6) {
      this.showMessage('Password must be at least 6 characters long!', 'error');
      return;
    }

    // Register user
    const result = this.authService.register(this.user);

    if (result.success) {
      this.showMessage(result.message, 'success');
      setTimeout(() => {
        this.router.navigate(['/login']);
      }, 1500);
    } else {
      this.showMessage(result.message, 'error');
    }
  }

  private showMessage(msg: string, type: 'success' | 'error'): void {
    this.message.set(msg);
    this.messageType.set(type);
    setTimeout(() => this.message.set(''), 5000);
  }
}
```

`src/app/register/register.html`
```html
<div class="register-container">
  <div class="register-card">
    <h2>Register</h2>

    @if (message()) {
    <div class="alert" [class.alert-success]="messageType() === 'success'"
      [class.alert-error]="messageType() === 'error'">
      {{ message() }}
    </div>
    }

    <form (ngSubmit)="onSubmit()" #registerForm="ngForm">
      <div class="form-group">
        <label for="username">Username *</label>
        <input type="text" id="username" name="username" [(ngModel)]="user.username" required
          class="form-control" placeholder="Enter username">
      </div>

      <div class="form-group">
        <label for="email">Email *</label>
        <input type="email" id="email" name="email" [(ngModel)]="user.email" required class="form-
control"
          placeholder="Enter email">
      </div>

      <div class="form-row">
        <div class="form-group">
          <label for="firstName">First Name</label>
          <input type="text" id="firstName" name="firstName" [(ngModel)]="user.firstName" class="form-
control"
            placeholder="First name">
        </div>

        <div class="form-group">
          <label for="lastName">Last Name</label>
          <input type="text" id="lastName" name="lastName" [(ngModel)]="user.lastName" class="form-
control"
            placeholder="Last name">
        </div>
      </div>

      <div class="form-group">
        <label for="phone">Phone</label>
        <input type="tel" id="phone" name="phone" [(ngModel)]="user.phone" class="form-control"
          placeholder="Enter phone number">
      </div>

      <div class="form-group">
        <label for="password">Password *</label>
        <input type="password" id="password" name="password" [(ngModel)]="user.password" required
minlength="6"
          class="form-control" placeholder="Enter password">
      </div>

      <div class="form-group">
```

```
            <label for="confirmPassword">Confirm Password *</label>
            <input type="password" id="confirmPassword" name="confirmPassword"
[(ngModel)]="confirmPassword"
                   required class="form-control" placeholder="Confirm password">
          </div>

          <button type="submit" class="btn btn-primary" [disabled]="!registerForm.form.valid">
            Register
          </button>
        </form>

        <p class="login-link">
          Already have an account? <a routerLink="/login">Login here</a>
        </p>
    </div>
</div>
```

`src/app/register/register.css`
```css
.register-container {
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    padding: 20px;
}

.register-card {
    background: white;
    border-radius: 10px;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.2);
    padding: 40px;
    width: 100%;
    max-width: 500px;
}

.register-card h2 {
    text-align: center;
    color: #333;
    margin-bottom: 30px;
    font-size: 28px;
}

.alert {
    padding: 12px;
    border-radius: 5px;
    margin-bottom: 20px;
    text-align: center;
    font-weight: 500;
}

.alert-success {
    background-color: #d4edda;
    color: #155724;
    border: 1px solid #c3e6cb;
}
```

```
.alert-error {
    background-color: #f8d7da;
    color: #721c24;
    border: 1px solid #f5c6cb;
}

.form-group {
    margin-bottom: 20px;
}

.form-row {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 15px;
}

.form-group label {
    display: block;
    margin-bottom: 5px;
    color: #555;
    font-weight: 600;
}

.form-control {
    width: 100%;
    padding: 12px;
    border: 1px solid #ddd;
    border-radius: 5px;
    font-size: 14px;
    transition: border-color 0.3s;
    box-sizing: border-box;
}

.form-control:focus {
    outline: none;
    border-color: #667eea;
}

.btn {
    width: 100%;
    padding: 12px;
    border: none;
    border-radius: 5px;
    font-size: 16px;
    font-weight: 600;
    cursor: pointer;
    transition: background-color 0.3s;
}

.btn-primary {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
}

.btn-primary:hover:not(:disabled) {
    opacity: 0.9;
}
```

```
.btn-primary:disabled {
    opacity: 0.6;
    cursor: not-allowed;
}

.login-link {
    text-align: center;
    margin-top: 20px;
    color: #666;
}

.login-link a {
    color: #667eea;
    text-decoration: none;
    font-weight: 600;
}

.login-link a:hover {
    text-decoration: underline;
}

@media (max-width: 600px) {
    .form-row {
        grid-template-columns: 1fr;
    }

    .register-card {
        padding: 30px 20px;
    }
}
```

9. Profile Component

`src/app/profile/profile.ts`
```typescript
import { Component, OnInit, computed } from '@angular/core';
import { Router } from '@angular/router';
import { Auth } from '../services/auth';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'app-profile',
  imports: [CommonModule],
  templateUrl: './profile.html',
  styleUrl: './profile.css',
})
export class Profile implements OnInit {
  currentUser = computed(() => this.authService.getCurrentUser()());

  constructor(private authService: Auth, private router: Router) { }

  ngOnInit(): void {
    // Redirect to login if not authenticated
    if (!this.authService.isLoggedIn()) {
      this.router.navigate(['/login']);
    }
```

```
  }

  onLogout(): void {
    this.authService.logout();
    this.router.navigate(['/login']);
  }
}
```

`src/app/profile/profile.html`
```html
<div class="profile-container">
  <div class="profile-card">
    <div class="profile-header">
      <div class="profile-avatar">
        {{ (currentUser()?.username && currentUser()!.username.charAt(0).toUpperCase()) || 'U' }}
      </div>
      <h2>Welcome, {{ currentUser()?.username }}!</h2>
    </div>

    <div class="profile-info">
      <h3>User Information</h3>

      <div class="info-group">
        <label>Username:</label>
        <span>{{ currentUser()?.username }}</span>
      </div>

      <div class="info-group">
        <label>Email:</label>
        <span>{{ currentUser()?.email }}</span>
      </div>

      @if (currentUser()?.firstName) {
      <div class="info-group">
        <label>First Name:</label>
        <span>{{ currentUser()?.firstName }}</span>
      </div>
      }

      @if (currentUser()?.lastName) {
      <div class="info-group">
        <label>Last Name:</label>
        <span>{{ currentUser()?.lastName }}</span>
      </div>
      }

      @if (currentUser()?.phone) {
      <div class="info-group">
        <label>Phone:</label>
        <span>{{ currentUser()?.phone }}</span>
      </div>
      }

      @if (currentUser()?.id) {
      <div class="info-group">
        <label>User ID:</label>
        <span>{{ currentUser()?.id }}</span>
```

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

      <button class="btn btn-logout" (click)="onLogout()">
        Logout
      </button>
    </div>
  </div>
```

`src/app/profile/profile.css`
```css
.profile-container {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  padding: 20px;
}

.profile-card {
  background: white;
  border-radius: 10px;
  box-shadow: 0 10px 25px rgba(0, 0, 0, 0.2);
  padding: 40px;
  width: 100%;
  max-width: 600px;
}

.profile-header {
  text-align: center;
  margin-bottom: 30px;
  padding-bottom: 20px;
  border-bottom: 2px solid #f0f0f0;
}

.profile-avatar {
  width: 100px;
  height: 100px;
  border-radius: 50%;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  color: white;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 48px;
  font-weight: bold;
  margin: 0 auto 20px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
}

.profile-header h2 {
  color: #333;
  margin: 0;
  font-size: 28px;
}
```

```
.profile-info {
    margin-bottom: 30px;
}

.profile-info h3 {
    color: #555;
    margin-bottom: 20px;
    font-size: 20px;
    border-bottom: 2px solid #667eea;
    padding-bottom: 10px;
}

.info-group {
    display: flex;
    padding: 15px;
    margin-bottom: 10px;
    background-color: #f8f9fa;
    border-radius: 5px;
    align-items: center;
}

.info-group label {
    font-weight: 600;
    color: #555;
    min-width: 120px;
    margin-right: 15px;
}

.info-group span {
    color: #333;
    flex: 1;
    word-break: break-word;
}

.btn {
    width: 100%;
    padding: 12px;
    border: none;
    border-radius: 5px;
    font-size: 16px;
    font-weight: 600;
    cursor: pointer;
    transition: all 0.3s;
}

.btn-logout {
    background-color: #dc3545;
    color: white;
}

.btn-logout:hover {
    background-color: #c82333;
    transform: translateY(-2px);
    box-shadow: 0 4px 10px rgba(220, 53, 69, 0.3);
}

@media (max-width: 600px) {
```

```
    .profile-card {
        padding: 30px 20px;
    }

    .info-group {
        flex-direction: column;
        align-items: flex-start;
    }

    .info-group label {
        min-width: auto;
        margin-bottom: 5px;
    }
}
```

10. Global Styles

`src/styles.css`
```css
/* You can add global styles to this file, and also import other style files */

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background-color: #f5f5f5;
}
```

🚀 Getting Started

Prerequisites
- Node.js (version 18 or higher)
- npm (Node Package Manager)

Installation Steps

1. **Clone or download the project**
   ```bash
   # If using git
   git clone <repository-url>
   cd user-auth-app
   ```

2. **Install dependencies**
   ```bash
   npm install
   ```

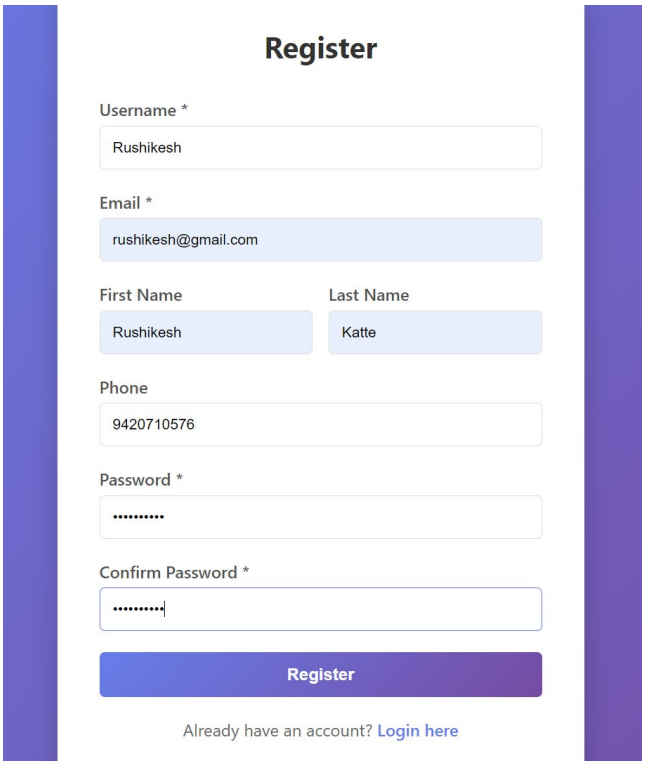
3. **Start the development server**
   ```bash
   npm start
   ```

4. **Open your browser**
   Navigate to `http://localhost:4200`

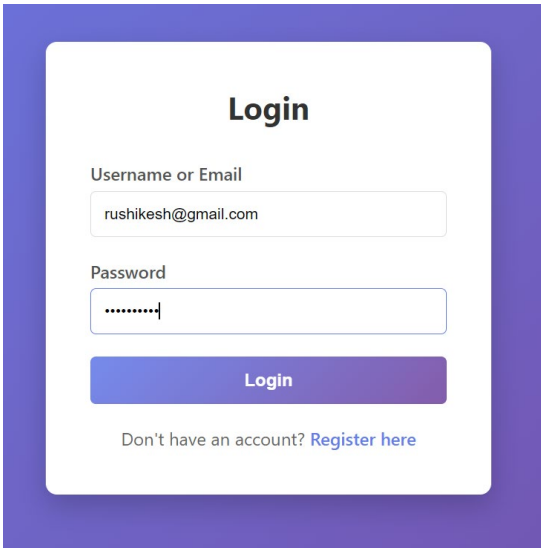
Build for Production
```bash
npm run build
```

Output

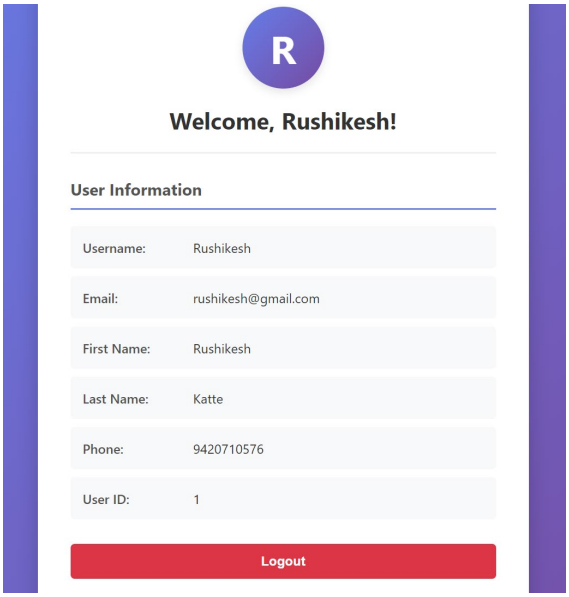
Register



Login



User Profile



Assignment 4A

Name: **Rushikesh Katte**

Roll Number: **23572**

Subject: **Web Technology**

1 Create a folder

mkdir static-website

cd static-website

2 Initialize Node.js project

npm init -y

This will create a package.json file automatically.

3 Install Express

npm install express

4 Create folder for static files

mkdir public

Inside public folder, create an index.html file:

📌 **public/index.html**

<!DOCTYPE html>

```
<html>
<head>
  <title>My Static Website</title>
</head>
<body>
  <h1>Hello World! This is a static website served by Node.js</h1>
</body>
</html>
```

5 Create main server file

Create server.js in the root directory:

📌 **server.js**

```
const express = require('express');
const app = express();
const PORT = 3000;

// Serve static files from "public" folder
app.use(express.static('public'));

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

6 Start the server

node server.js

Now open your browser → **http://localhost:3000**

You will see your static website served by Node.js

Assignment 4B

Name: **Rushikesh Katte**

Roll Number: **23572**

Subject: **Web Technology**

BACKEND:

1 Create project folder

mkdir crud-api
cd crud-api

2 Initialize Node project

npm init -y

3 Install dependencies

npm install express mongoose cors

Folder Structure

```
crud-api
 ├─ server.js
 ├─ models
 │  └─ itemModel.js
```

Create MongoDB Model

📌 **models/itemModel.js**

```
const mongoose = require('mongoose');

const ItemSchema = new mongoose.Schema({
  name: String,
  description: String,
  price: Number
});

module.exports = mongoose.model('Item', ItemSchema);
```

Main Server Code (CRUD APIs)

📌 server.js

```
const express = require('express');
const mongoose = require('mongoose');
const Item = require('./models/itemModel');
const cors = require('cors');

const app = express();
app.use(cors());
app.use(express.json());

mongoose.connect('mongodb://localhost:27017/assignment3C')
.then(() => console.log("MongoDB Connected"))
.catch(error => console.log(error));

app.post('/items', async (req, res) => {

  try {
    const item = await Item.create(req.body);
    res.status(201).json(item);
  } catch (error) {
    res.status(400).json({ error: error.message });
  }
});

app.get('/items', async (req, res) => {
  const items = await Item.find();
  res.json(items);
});

app.put('/items/:id', async (req, res) => {
  try {
    const item = await Item.findByIdAndUpdate(req.params.id, req.body, { new: true });
    res.json(item);
  } catch (error) {
    res.status(400).json({ error: error.message });
  }
});
```

```
app.delete('/items/:id', async (req, res) => {
  try {
    await Item.findByIdAndDelete(req.params.id);
    res.json({ message: "Item deleted successfully" });
  } catch (error) {
    res.status(400).json({ error: error.message });
  }
});


const PORT = 3000;
app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));
```

Run Server

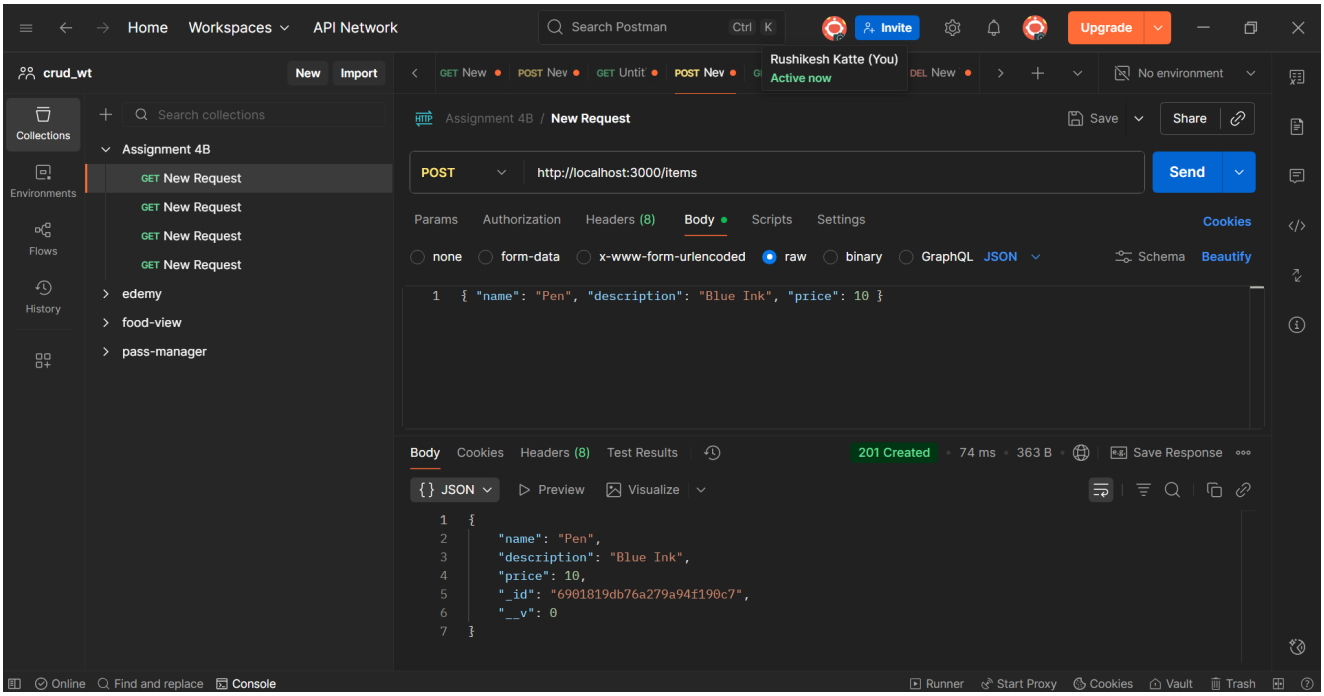
node server.js

```
C:\Users\rushi\OneDrive\Desktop\WT LAB\04\02. CRUD\crud-api>node server.js
Server running on http://localhost:3000
MongoDB Connected
```

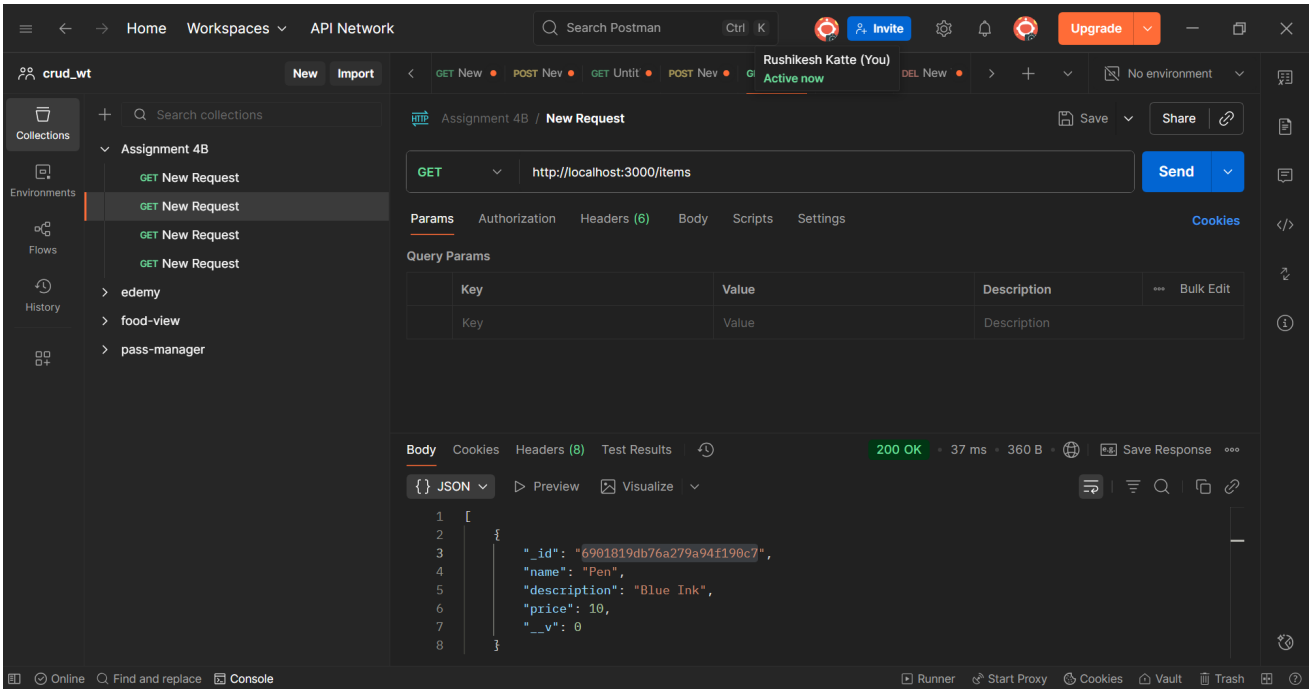
Test API using Postman

Operation	Method	Endpoint	Request Body Example
Create	POST	/items	{ "name": "Pen", "description": "Blue Ink", "price": 10 }
Read	GET	/items	Not required
Update	PUT	/items/:id	{ "price": 15 }
Delete	DELETE	/items/:id	Not required

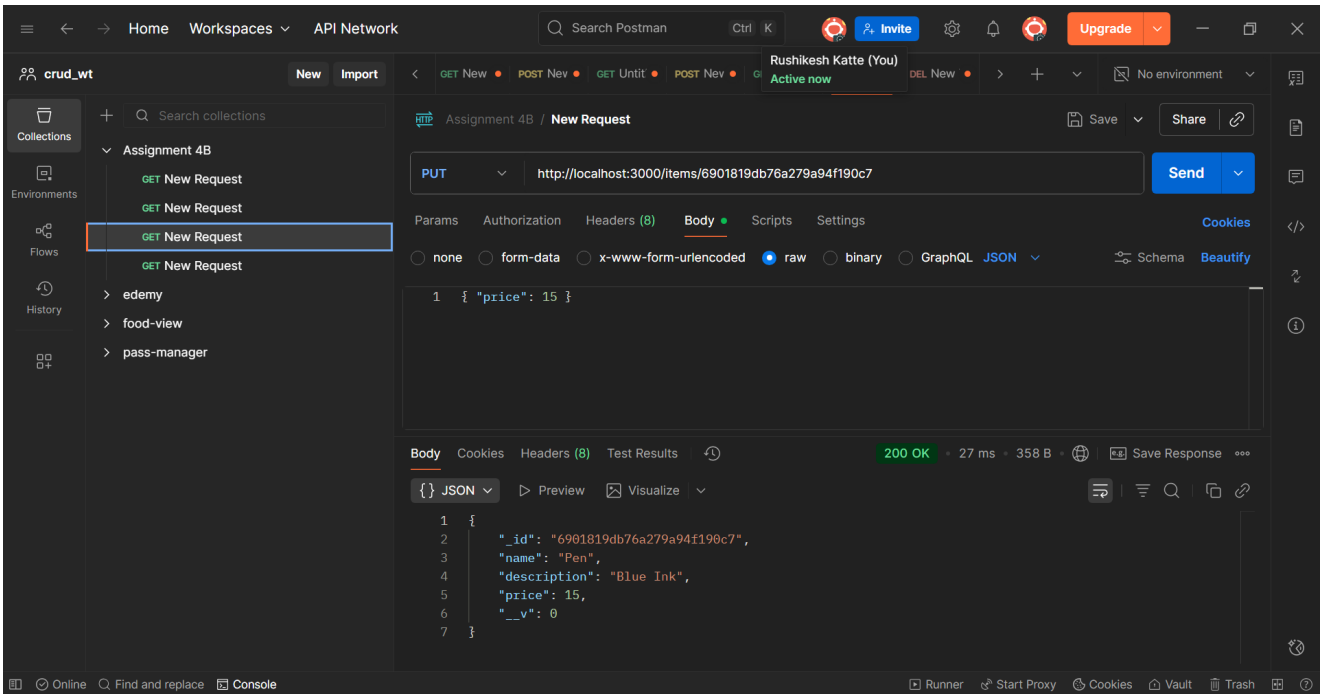
Create:


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
crud_wt
New
Import
Rushikesh Katte (You)
Active now
Collections
Environments
Flows
History
Search collections
Assignment 4B
GET New Request
edemy
food-view
pass-manager
No environment
Assignment 4B / New Request
Save
Share
POST
http://localhost:3000/items
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Schema
Beautify
{ "name": "Pen", "description": "Blue Ink", "price": 10 }
Body
Cookies
Headers (8)
Test Results
201 Created
74 ms
363 B
Save Response
JSON
Preview
Visualize
{
"name": "Pen",
"description": "Blue Ink",
"price": 10,
"_id": "6901819db76a279a94f190c7",
"__v": 0
}
Online
Find and replace
Console
Runner
Start Proxy
Cookies
Vault
Trash

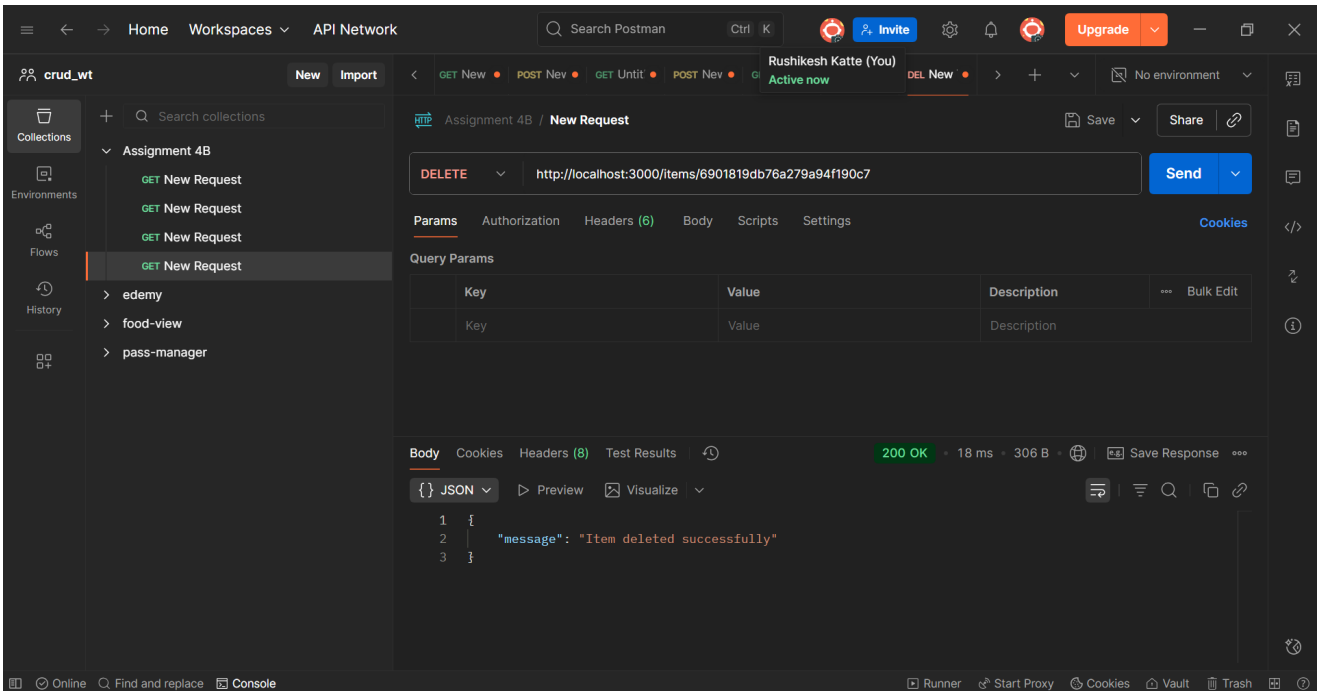
Read:


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
crud_wt
New
Import
Rushikesh Katte (You)
Active now
Collections
Environments
Flows
History
Search collections
Assignment 4B
GET New Request
edemy
food-view
pass-manager
No environment
Assignment 4B / New Request
Save
Share
GET
http://localhost:3000/items
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (8)
Test Results
200 OK
37 ms
360 B
Save Response
JSON
Preview
Visualize
[
{
"_id": "6901819db76a279a94f190c7",
"name": "Pen",
"description": "Blue Ink",
"price": 10,
"__v": 0
}
Online
Find and replace
Console
Runner
Start Proxy
Cookies
Vault
Trash

Update:



Delete:



Frontend:

📌 Folder Structure

Add a folder in the same project:

```
crud-api
 ├─ server.js
 ├─ models/
 ├─ public/
 │   ├─ index.html
 │   ├─ script.js
```

Also update your server to serve static files:
In **server.js**, add:

```
app.use(express.static('public'));
```

Frontend Code

1 index.html

📌 public/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CRUD Frontend</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body class="p-4">
  <div class="container">
    <h2 class="mb-4 text-center">CRUD Application</h2>

    <!-- Add / Update Form -->
    <div class="card p-3">
      <input type="hidden" id="itemId">
      <div class="mb-2">
        <label>Name</label>
        <input type="text" id="name" class="form-control">
      </div>
      <div class="mb-2">
        <label>Description</label>
        <input type="text" id="description" class="form-control">
```

```
      </div>
      <div class="mb-2">
        <label>Price</label>
        <input type="number" id="price" class="form-control">
      </div>
      <button class="btn btn-success" onclick="addItem()">Add Item</button>
      <button class="btn btn-warning d-none" id="updateBtn" onclick="updateItem()">Update Item</button>
    </div>

    <hr>

    <h4>Items</h4>
    <table class="table table-bordered">
      <thead>
        <tr>
          <th>Name</th>
          <th>Description</th>
          <th>Price</th>
          <th>Actions</th>
        </tr>
      </thead>
      <tbody id="itemTable"></tbody>
    </table>
  </div>

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

---

2 script.js

📌 public/script.js

```
const API = "http://localhost:3000/items";

async function fetchItems() {
  const res = await fetch(API);
  const items = await res.json();

  const table = document.getElementById("itemTable");
  table.innerHTML = "";

  items.forEach(item => {
    table.innerHTML += `
      <tr>
        <td>${item.name}</td>
        <td>${item.description}</td>
        <td>${item.price}</td>
        <td>
          <button class="btn btn-primary btn-sm" onclick="editItem('${item._id}')">Edit</button>
```

```
            <button class="btn btn-danger btn-sm" onclick="deleteItem('${item._id}')">Delete</button>
          </td>
        </tr>
      `;
    });
}

async function addItem() {
    const name = document.getElementById("name").value;
    const description = document.getElementById("description").value;
    const price = document.getElementById("price").value;

    await fetch(API, {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ name, description, price })
    });

    clearForm();
    fetchItems();
}

async function editItem(id) {
    const res = await fetch(`${API}/${id}`);
    const item = await res.json();

    document.getElementById("itemId").value = item._id;
    document.getElementById("name").value = item.name;
    document.getElementById("description").value = item.description;
    document.getElementById("price").value = item.price;

    document.querySelector("button.btn-success").classList.add("d-none");
    document.getElementById("updateBtn").classList.remove("d-none");
}

async function updateItem() {
    const id = document.getElementById("itemId").value;
    const name = document.getElementById("name").value;
    const description = document.getElementById("description").value;
    const price = document.getElementById("price").value;

    await fetch(`${API}/${id}`, {
      method: "PUT",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ name, description, price })
    });

    clearForm();
    fetchItems();
}

async function deleteItem(id) {
    await fetch(`${API}/${id}`, { method: "DELETE" });
```

```
    fetchItems();
}

function clearForm() {
    document.getElementById("itemId").value = "";
    document.getElementById("name").value = "";
    document.getElementById("description").value = "";
    document.getElementById("price").value = "";

    document.querySelector("button.btn-success").classList.remove("d-none");
    document.getElementById("updateBtn").classList.add("d-none");
}

// Initial load
fetchItems();
```

Run Full Application

1 Start MongoDB

```
mongod
```

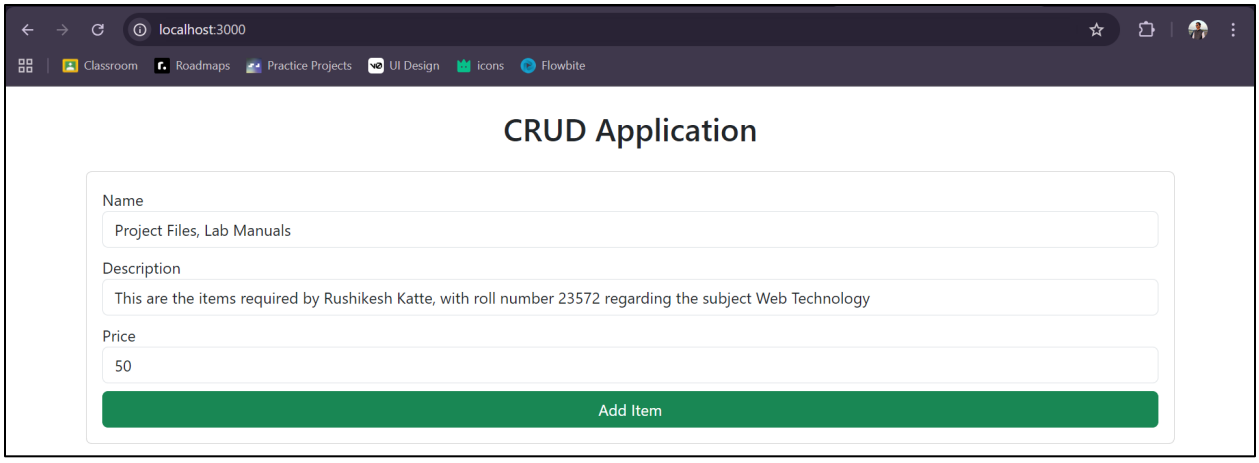
2 Run backend

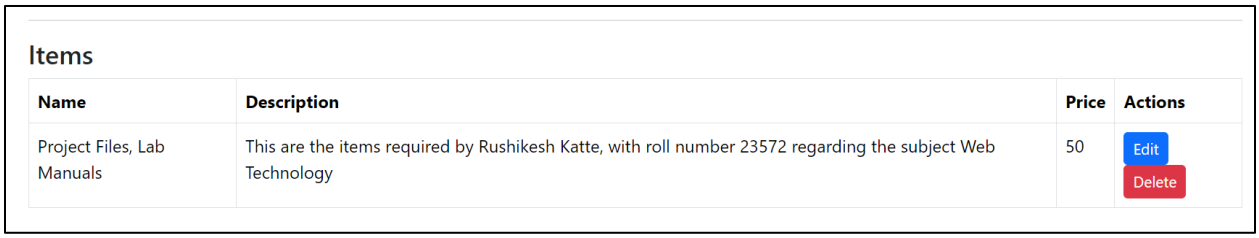
```
node server.js
```

3 Open browser
http://localhost:3000

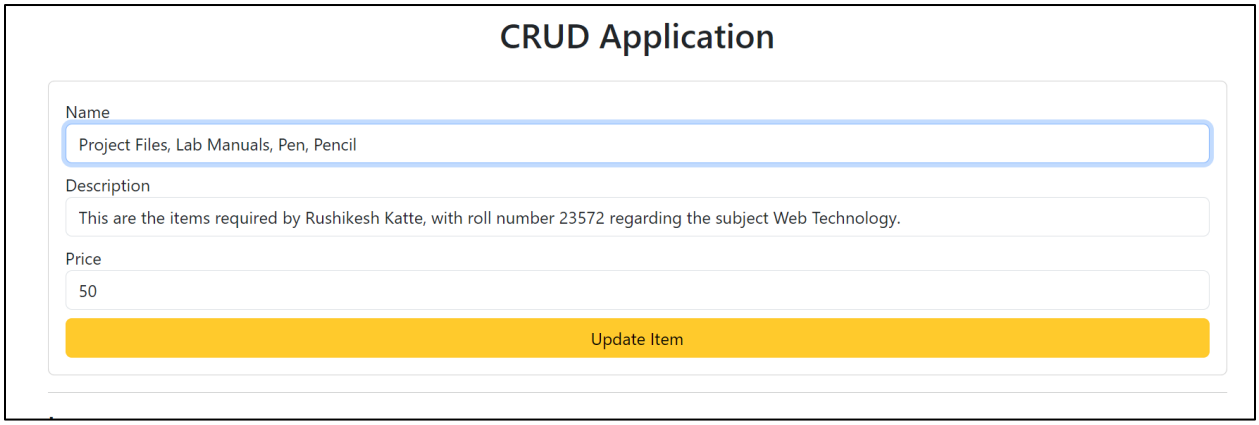
You can now **Add / View / Edit / Delete** items from UI
Full CRUD system working!

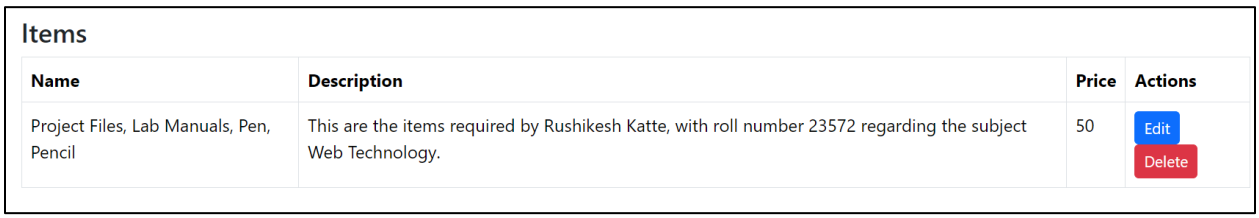
Add Item:





Update Item:





Delete Item:

Assignment 5A

Name: **Rushikesh Katte**

Roll Number: **23572**

Subject: **Web Technology**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery Mobile Demo</title>

  <link rel="stylesheet" href="https://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.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>

  <style>
    .hero {
      text-align: center;
      padding: 24px 12px;
    }

    .hero h2 {
```

```
        margin: 0 0 8px;
      }

      .footer-note {
        font-size: 12px;
        color: #666;
        text-align: center;
        padding: 8px 0;
      }

      .responsive-img {
        max-width: 100%;
        height: auto;
        border-radius: 8px;
      }
    </style>
</head>

<body>

    <!-- HOME PAGE -->
    <div data-role="page" id="home" data-theme="a">
      <div data-role="header" data-position="fixed">
        <h1>My Mobile Site</h1>
        <div data-role="navbar">
          <ul>
            <li><a href="#home" class="ui-btn-active" data-icon="home">Home</a></li>
```

```
        <li><a href="#list" data-icon="bullets">List</a></li>
        <li><a href="#contact" data-icon="mail">Contact</a></li>
      </ul>
    </div>
  </div>

  <div role="main" class="ui-content">
    <div class="hero">
      <h2>Welcome</h2>
      <p>A simple, responsive jQuery Mobile starter.</p>
    </div>

    <!-- jQuery Mobile responsive grid -->
    <div class="ui-grid-a ui-responsive">
      <div class="ui-block-a">
        <div class="ui-body ui-body-a">
          <h3>Fast Setup</h3>
          <p>Built with CDN links—no local setup required.</p>
        </div>
      </div>
      <div class="ui-block-b">
        <div class="ui-body ui-body-a">
          <h3>Responsive</h3>
          <p>Adapts to phones, tablets, and desktops.</p>
        </div>
      </div>
    </div>
```

```
      <div style="margin-top:16px; text-align:center;">
        <img class="responsive-img" src="https://picsum.photos/800/400" alt="Sample">
      </div>

      <a href="#list" class="ui-btn ui-btn-b ui-corner-all ui-shadow" data-icon="arrow-r"
        data-iconpos="right">Explore the List</a>
    </div>

    <div data-role="footer" data-position="fixed">
      <h4>Home</h4>
    </div>
  </div>

  <!-- LIST PAGE -->
  <div data-role="page" id="list" data-theme="a">
    <div data-role="header" data-position="fixed">
      <h1>Items</h1>
      <div data-role="navbar">
        <ul>
          <li><a href="#home" data-icon="home">Home</a></li>
          <li><a href="#list" class="ui-btn-active" data-icon="bullets">List</a></li>
          <li><a href="#contact" data-icon="mail">Contact</a></li>
        </ul>
      </div>
    </div>
```

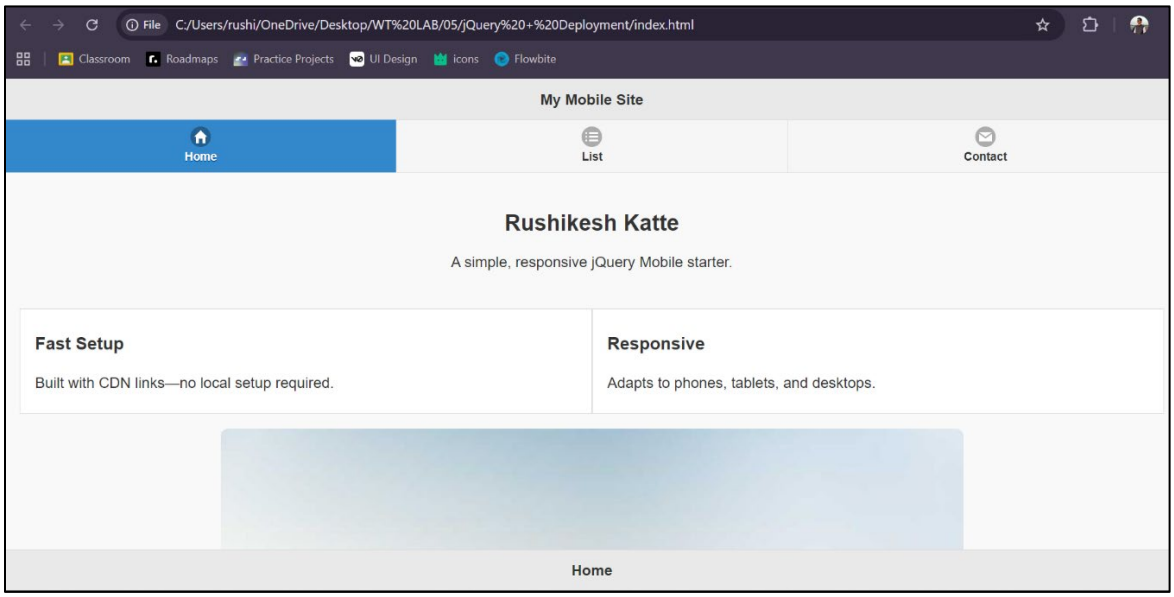
```
    <div role="main" class="ui-content">
        <form class="ui-filterable">
            <input id="filter-input" data-type="search" placeholder="Search items...">
        </form>

        <ul data-role="listview" data-inset="true" data-filter="true" data-input="#filter-input">
            <li data-role="list-divider">Featured</li>
            <li><a href="#">Alpha</a></li>
            <li><a href="#">Bravo</a></li>
            <li><a href="#">Charlie</a></li>
            <li><a href="#">Delta</a></li>
            <li data-role="list-divider">More</li>
            <li><a href="#">Echo</a></li>
            <li><a href="#">Foxtrot</a></li>
            <li><a href="#">Golf</a></li>
            <li><a href="#">Hotel</a></li>
        </ul>

        <a href="#home" class="ui-btn ui-corner-all" data-icon="arrow-l">Back to Home</a>
    </div>

    <div data-role="footer" data-position="fixed">
        <h4>List</h4>
    </div>
</div>

<!-- CONTACT PAGE -->
```

```
  <div data-role="page" id="contact" data-theme="a">
    <div data-role="header" data-position="fixed">
      <h1>Contact</h1>
      <div data-role="navbar">
        <ul>
          <li><a href="#home" data-icon="home">Home</a></li>
          <li><a href="#list" data-icon="bullets">List</a></li>
          <li><a href="#contact" class="ui-btn-active" data-icon="mail">Contact</a></li>
        </ul>
      </div>
    </div>

    <div role="main" class="ui-content">
      <form id="contact-form">
        <div class="ui-field-contain">
          <label for="name">Name</label>
          <input type="text" name="name" id="name" placeholder="Jane Doe" required>
        </div>
        <div class="ui-field-contain">
          <label for="email">Email</label>
          <input type="email" name="email" id="email" placeholder="jane@example.com"
required>
        </div>
        <div class="ui-field-contain">
          <label for="message">Message</label>
          <textarea name="message" id="message" placeholder="Write your message..."
required></textarea>
        </div>
```

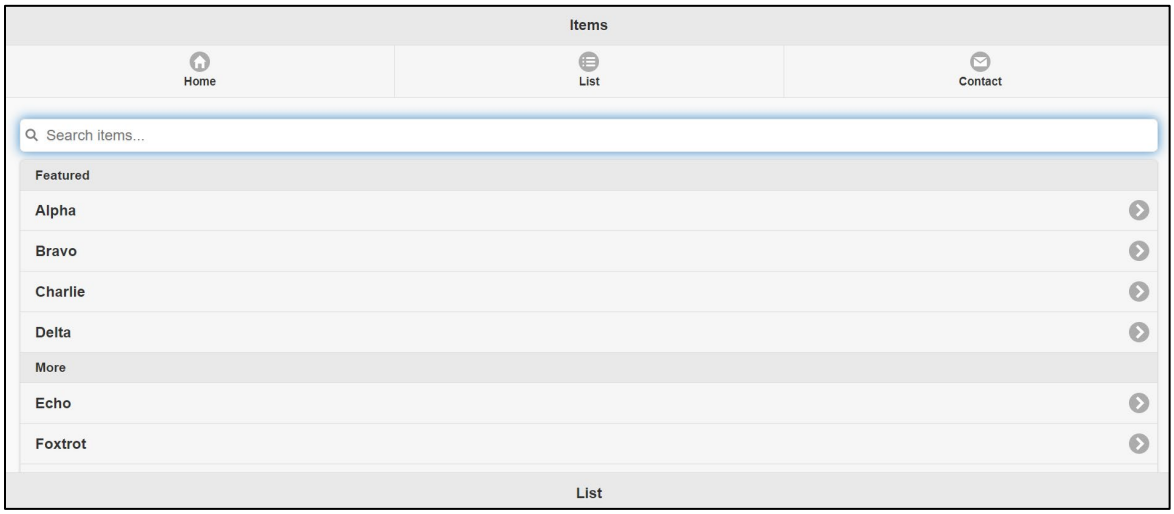
```
        <button type="submit" class="ui-btn ui-btn-b ui-corner-all ui-shadow" data-icon="check">Send</button>
      </form>
    </div>

    <div data-role="footer" data-position="fixed">
      <h4>Contact</h4>
    </div>
  </div>

  <script>
    // Prevent actual navigation for '#' links in list for demo clarity
    $(document).on('click', 'a[href="#"]', function (e) {
      e.preventDefault();
    });

    // Demo submit handler
    $(document).on('submit', '#contact-form', function (e) {
      e.preventDefault();
      var name = $('#name').val().trim();
      var email = $('#email').val().trim();
      var message = $('#message').val().trim();
      if (!name || !email || !message) {
        alert('Please fill out all fields.');
        return;
      }
      alert('Thanks, ' + name + '! Your message was captured locally.');
```

```
        this.reset();
      });
  </script>
</body>
</html>
```

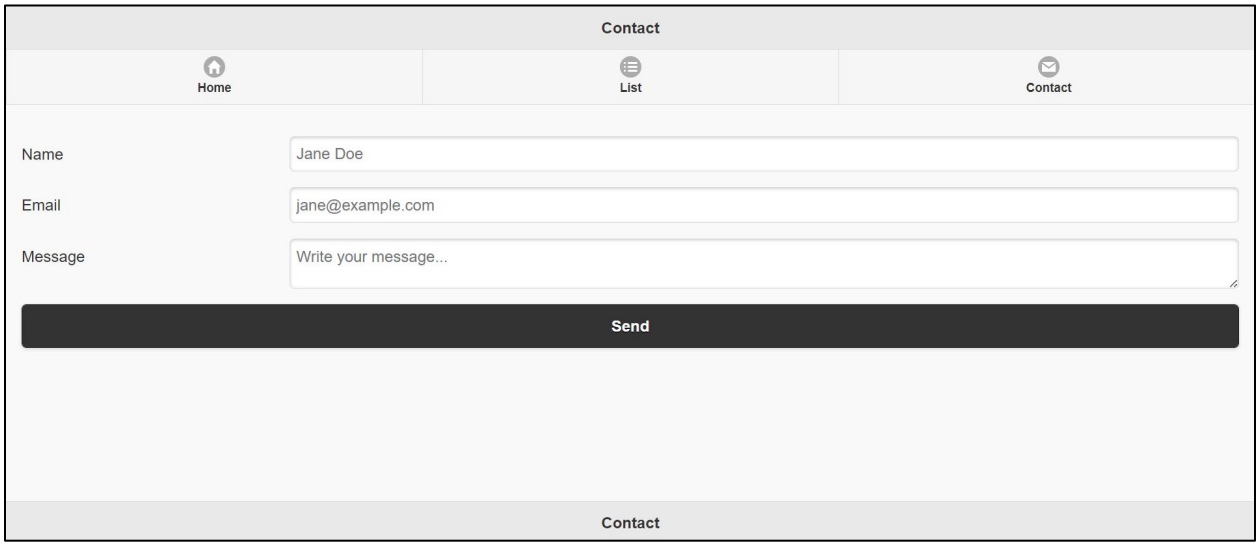
Home:



List:



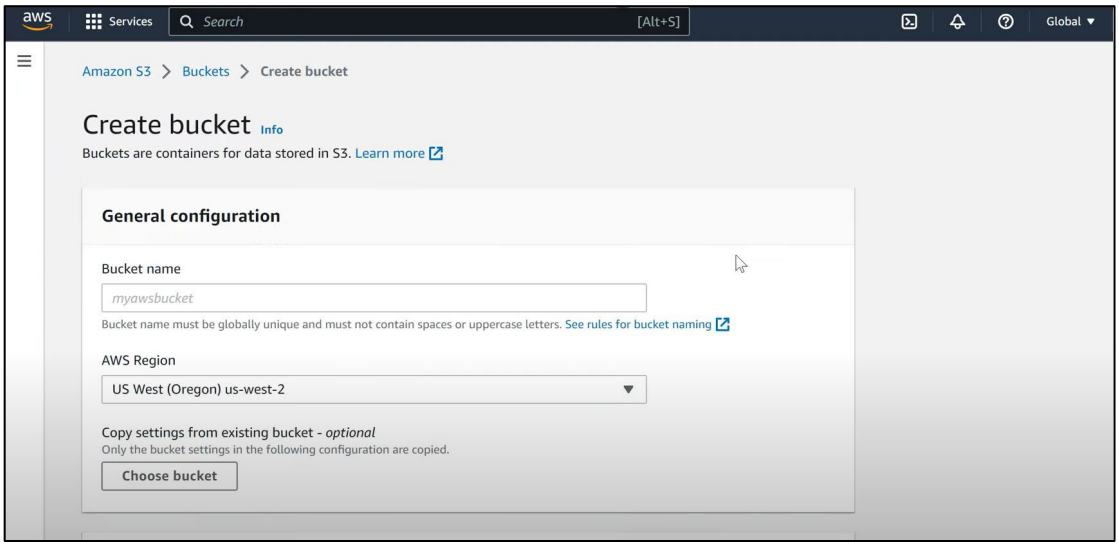
Contact:

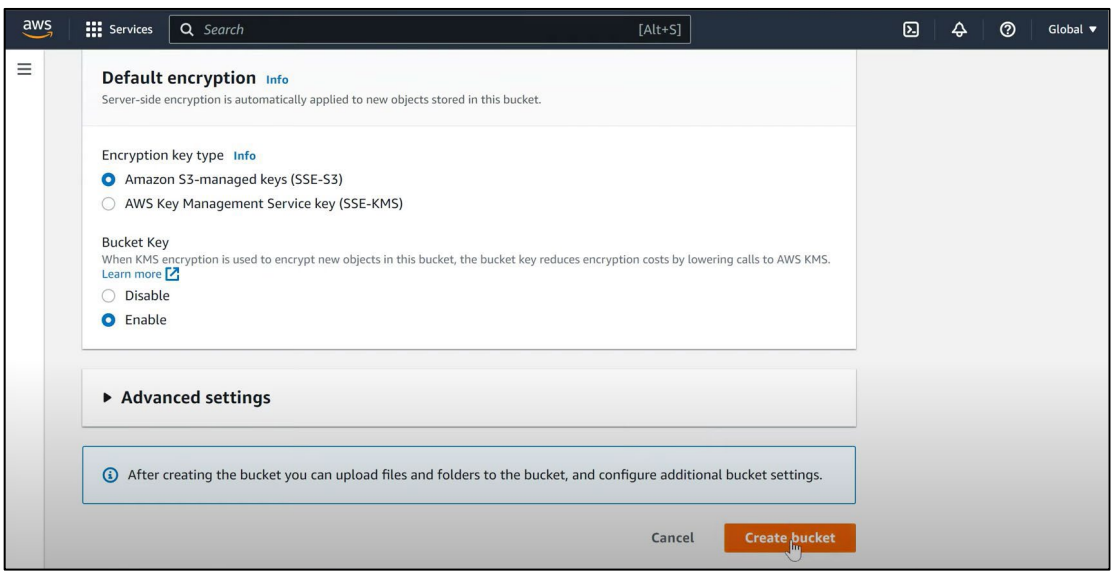

Contact
Home
List
Contact
Name
Jane Doe
Email
jane@example.com
Message
Write your message...
Send
Contact

Assignment 5B

Name: **Rushikesh Katte**
Roll Number: **23572**
Subject: **Web Technology**

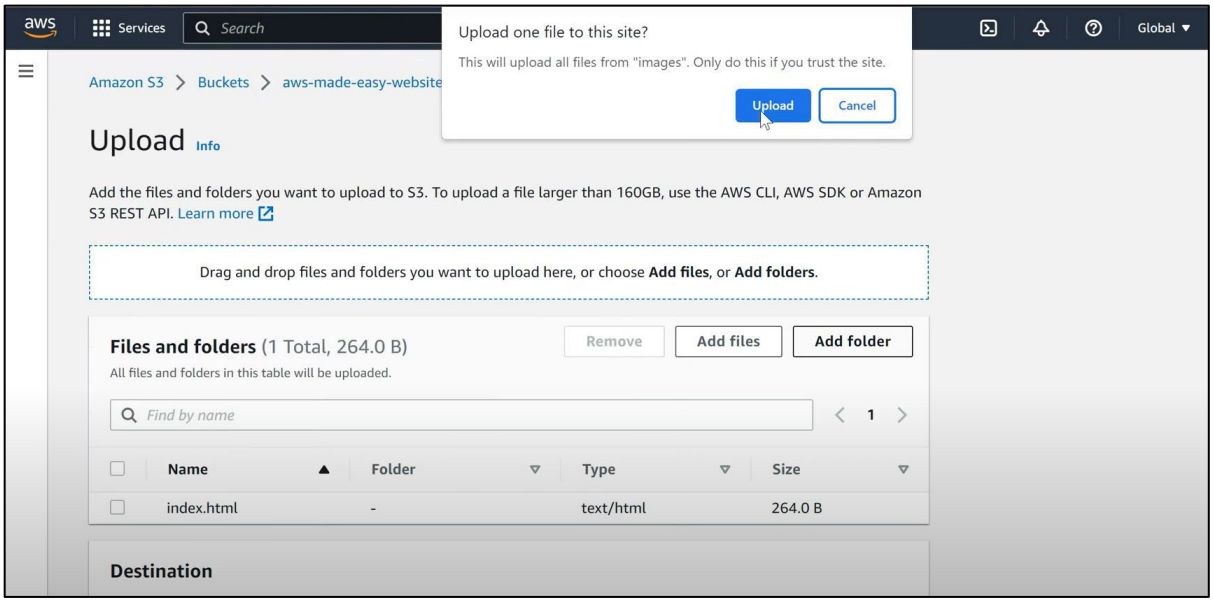
First Create a Bucket with selections as follows:



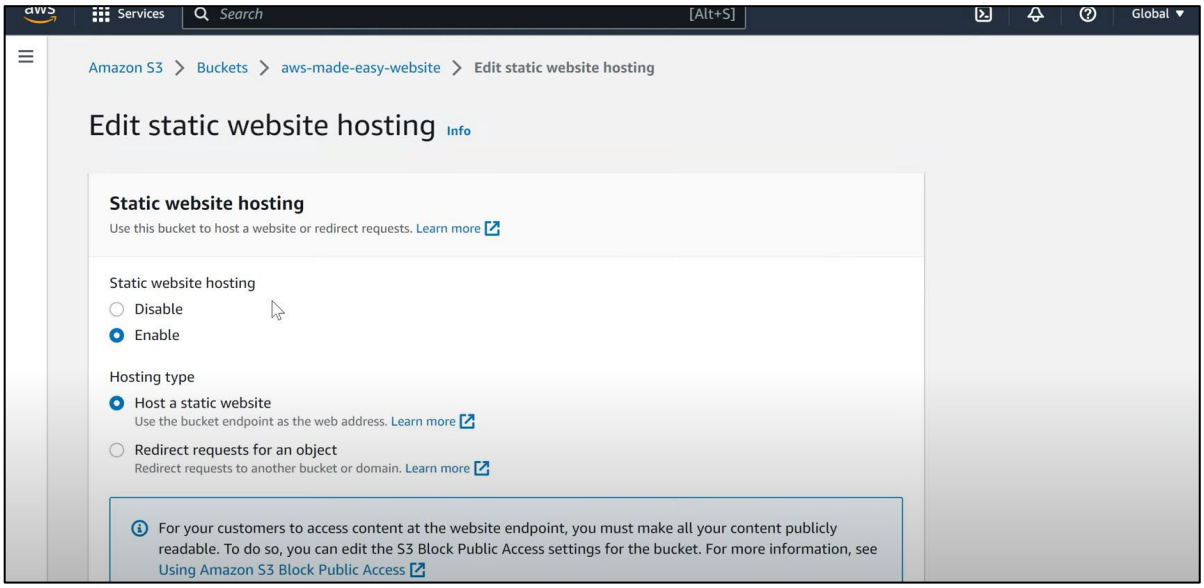


Assignment 5B

Upload Project Files:

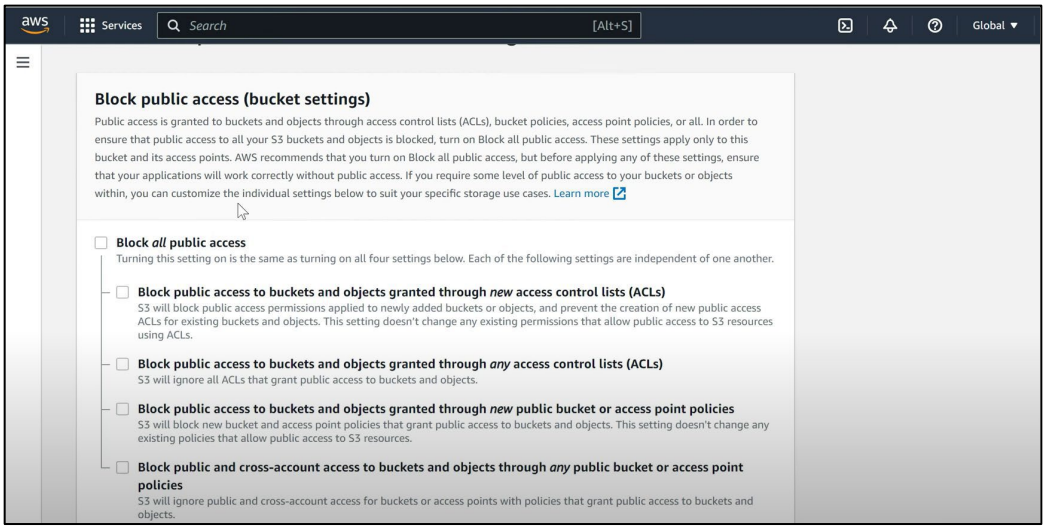


Enable Static Website Hosting:



Assignment 5B

Uncheck Block public access:



Output at the domain as follows:

http://my-static-site.s3-website.us-west-2.amazonaws.com/index.html