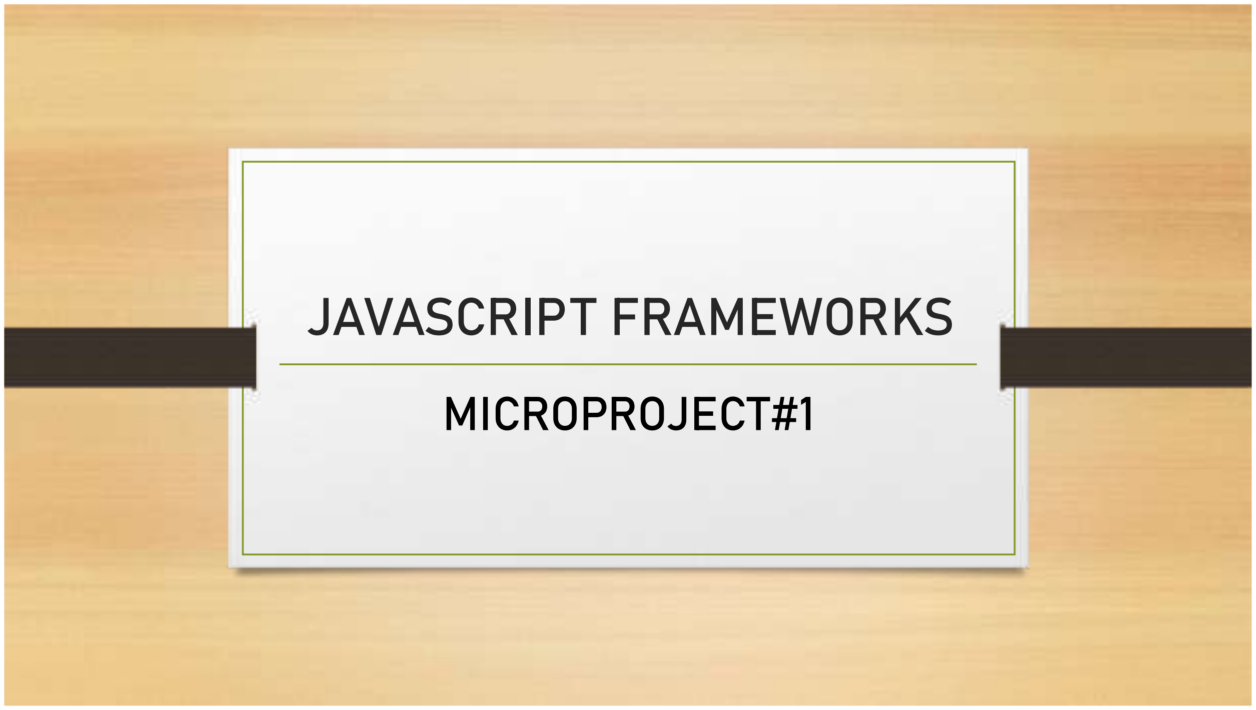


JAVASCRIPT FRAMEWORKS

MICROPROJECT#1

Group Members:

- Smile Smile(200563908)
- Dev Dev(200562142)
- Robin Robin(200571744)

Assignment Instructions:

You will create an express application:

1. Install the full Express required and recommended dependencies:
2. Install Express
3. Install Nodemon
4. Add a folder named “data” inside the project folder. Inside the “data” folder add a JSON file with at least 6 items (objects)
5. Adding a "public" folder for a full well-designed HTML website (static website):
6. 4 HTML pages at least
7. images/media and professional (nice and elegant) CSS
8. One of the HTML pages will be a demo for using API (Application Programming Interface) request to a back-end side (server-side) to load the JSON data (using Express with Node for sure)
9. This page should include a button to communicate with Express URL Route to call (fetch) JSON data
10. The JSON data that you will grab/load from the server-side should be displayed in a nicely formatted and professional layout to the end user (clients) in this page
11. you will need a JavaScript file to handle the front-end interactions with the user where you will attach an event listener to the button of the page. When the user clicks the button you will call function with Async/Await* to fetch the data according to the URL that you specified (or will specify) in your application JS file.
12. * Async/Await is the syntactic sugar of using promises
13. For sure, you will have to add the JavaScript file for running Express using the local virtual server "localhost"
14. Use any name you prefer to this Express application JavaScript file (by default, programmers use app.js or you can use index.js)
15. Write the full template/boilerplate of Express from you repos code examples or the express documentations
16. https://expressjs.com/en/starter/hello-world.html

17. Import or require the JSON file to be saved into a variable of any name of your choice
18. Add the .get() method with a specific route to send the JSON Data which the variable that you have created that contains the JSON content. Notice that the URL and the end-point of this get() method will be used in your client-side JavaScript file.

Assignment Submission Materials:

1. The PDF file that contains the screenshots explained below:

- Use MS-Word to collect all the assignment images (screenshots), put and arrange them all in one professional document then convert it to a PDF file to be uploaded/submitted. Please consider the following screenshots (images):
- The folder structure inside IntelliJ IDEA/Visual Studio Code or any other Editor that you are using that shows the 2 listed files above
- The VS Code embedded terminal window that shows the current working directory (the path of your folder)
- The PDF file should prepared in a professional way with a cover page

2. The link to your GitHub repository where you have your assignment (project) uploaded

1. Introduction

The Sneaker Store Web Application is a simple and dynamic project designed to showcase a collection of sneakers by fetching data from a JSON API. The application displays the sneaker details, including images, descriptions, and prices, in a visually appealing layout. It is built using HTML, CSS, and JavaScript, with a focus on modern web development practices such as fetching data from external sources and dynamically rendering it on the webpage.

This project demonstrates the use of essential web technologies and offers hands-on experience in integrating APIs, handling asynchronous data requests, and ensuring a responsive, user-friendly design.

2. Technologies Used

- HTML (Hypertext Markup Language)
  - ➢ Used for structuring the content of the web pages. It provides the skeleton of the website by defining the layout, forms, and other essential elements.
- CSS (Cascading Style Sheets)
  - ➢ Used for styling the web pages, making them visually appealing. CSS controls the layout, colors, fonts, spacing, and other design aspects. In this project, responsive design principles are implemented to ensure the site looks good on different screen sizes.
- JavaScript
  - ➢ Used for dynamic content manipulation and making the webpage interactive. In this project, JavaScript is used to fetch data from an external JSON API and render it dynamically on the web page.
- JSON (JavaScript Object Notation)
  - ➢ JSON is used as the data format for storing and transmitting sneaker information. It enables easy parsing and rendering of the sneaker data on the webpage.
- API (Application Programming Interface)
  - ➢ An external API is used to fetch sneaker details like name, price, image, and description. JavaScript makes asynchronous requests to this API to retrieve and display the data in real time.

Step1 : Install Node.js

Node.js provides the runtime environment for running JavaScript code server-side. This is required to build the backend for our Sneaker Store application.

Step2: Create Project Folder

We will create a folder named as sneaker store and after that we will Initialize Node.js Project to create a package.json file with default configurations.

Step 3: Install Required Packages

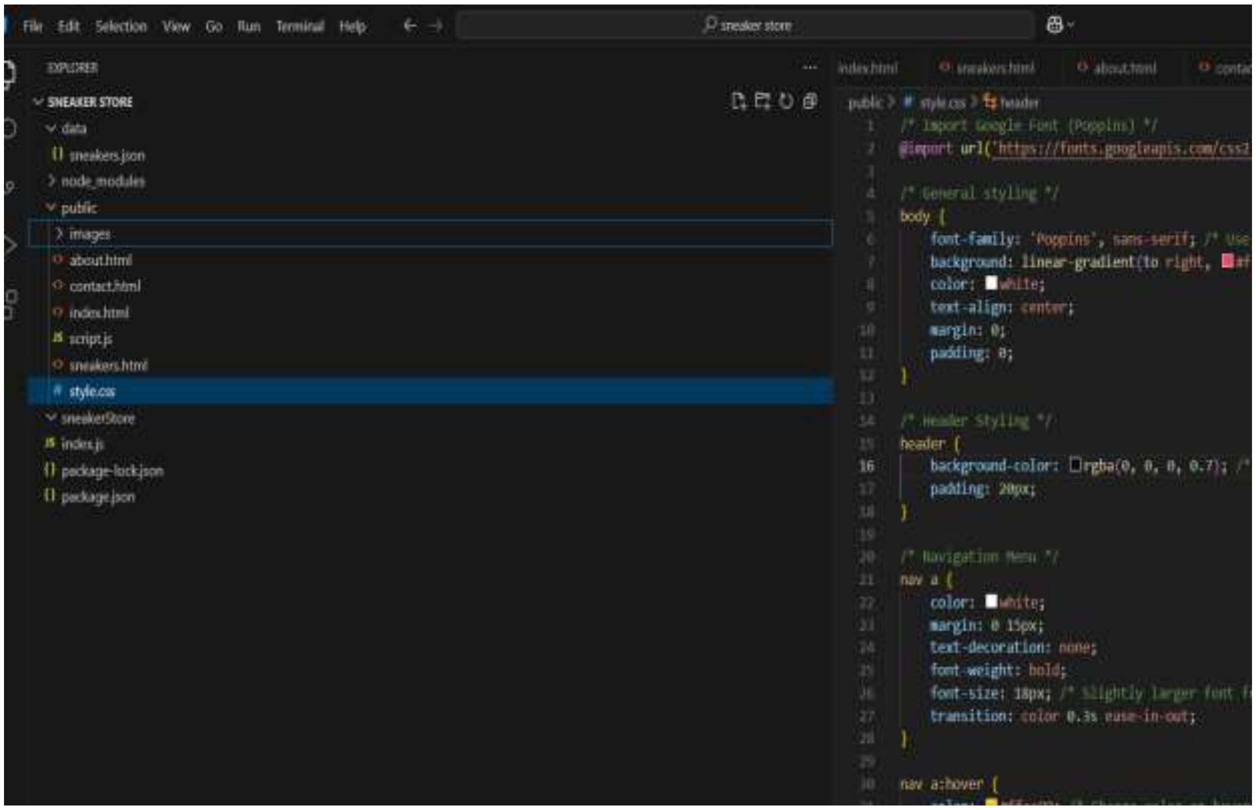
In this step we will Install Express.js and Install Nodemon (for automatic server restart)

- npm install express
- npm install --save-dev nodemon

Step 4: Create Project Files

In step 4 ,we will create all the essential files to get the output and the project structure is as following

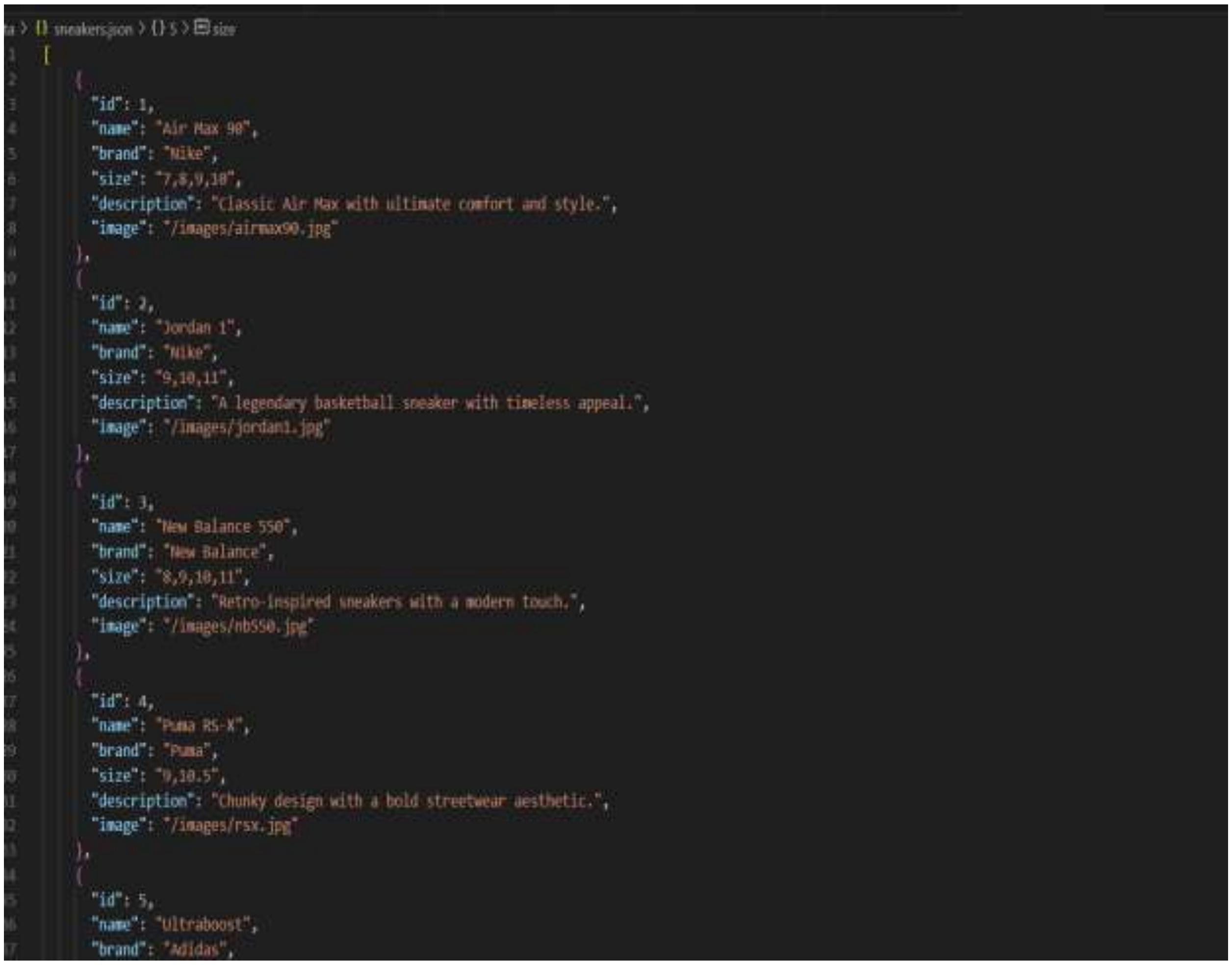
- Create necessary files and folders like:
    - data/sneakers.json (to store sneaker data in JSON format)
    - public/ (to store static files like images, HTML, CSS, and JS files)
    - sneakerStore/ (for backend code and Express setup)
    - package.json (for dependency management)


File Edit Selection View Go Run Terminal Help
sneaker store
EXPLORER
index.html
sneakers.html
about.html
conta
public > # style.css > header

```
SNEAKER STORE
  data
    sneakers.json
  node_modules
  public
    images
    about.html
    contact.html
    index.html
    script.js
    sneakers.html
    style.css
  sneakerStore
  index.js
  package-lock.json
  package.json
```

```css
/* Import Google Font (Poppins) */
@import url('https://fonts.googleapis.com/css2

/* General styling */
body {
    font-family: 'Poppins', sans-serif; /* Use
    background: linear-gradient(to right, #f
    color: white;
    text-align: center;
    margin: 0;
    padding: 0;
}

/* Header Styling */
header {
    background-color: rgba(0, 0, 0, 0.7); /*
    padding: 20px;
}

/* Navigation Menu */
nav a {
    color: white;
    margin: 0 15px;
    text-decoration: none;
    font-weight: bold;
    font-size: 18px; /* Slightly larger font f
    transition: color 0.3s ease-in-out;
}

nav a:hover {
```

Step 5: Create sneakers.json in data/ Folder



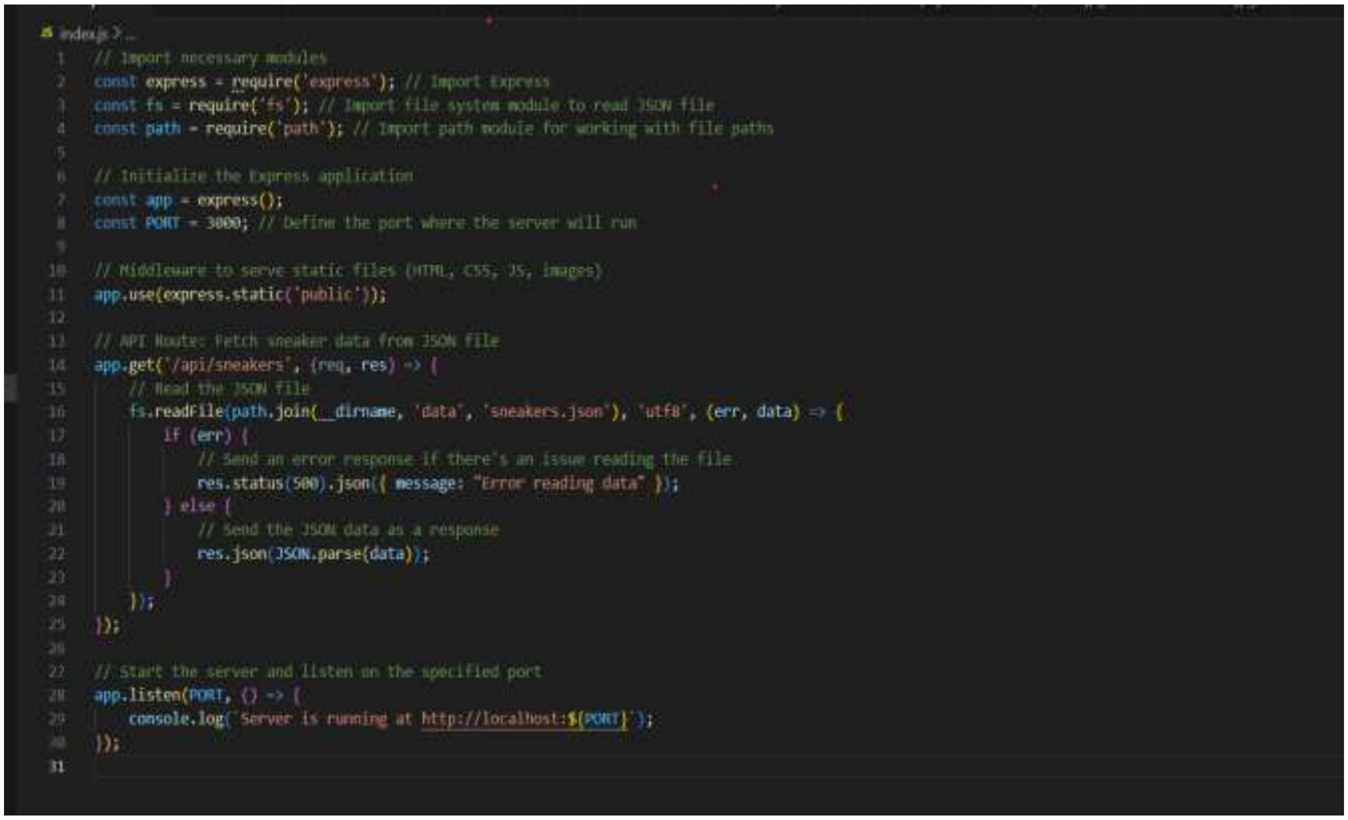
```
[
    {
      "id": 1,
      "name": "Air Max 90",
      "brand": "Nike",
      "size": "7,8,9,10",
      "description": "Classic Air Max with ultimate comfort and style.",
      "image": "/images/airmax90.jpg"
    },
    {
      "id": 2,
      "name": "Jordan 1",
      "brand": "Nike",
      "size": "9,10,11",
      "description": "A legendary basketball sneaker with timeless appeal.",
      "image": "/images/jordan1.jpg"
    },
    {
      "id": 3,
      "name": "New Balance 550",
      "brand": "New Balance",
      "size": "8,9,10,11",
      "description": "Retro-inspired sneakers with a modern touch.",
      "image": "/images/nb550.jpg"
    },
    {
      "id": 4,
      "name": "Puma RS-X",
      "brand": "Puma",
      "size": "9,10.5",
      "description": "Chunky design with a bold streetwear aesthetic.",
      "image": "/images/rsx.jpg"
    },
    {
      "id": 5,
      "name": "Ultraboost",
      "brand": "Adidas",
```

Step 6: Set up Express.js CRUD application:

This is where we set up the backend using Express to serve routes and static files.

Action:

- Inside the sneaker-store folder, create INDEX.JS and add the following content:



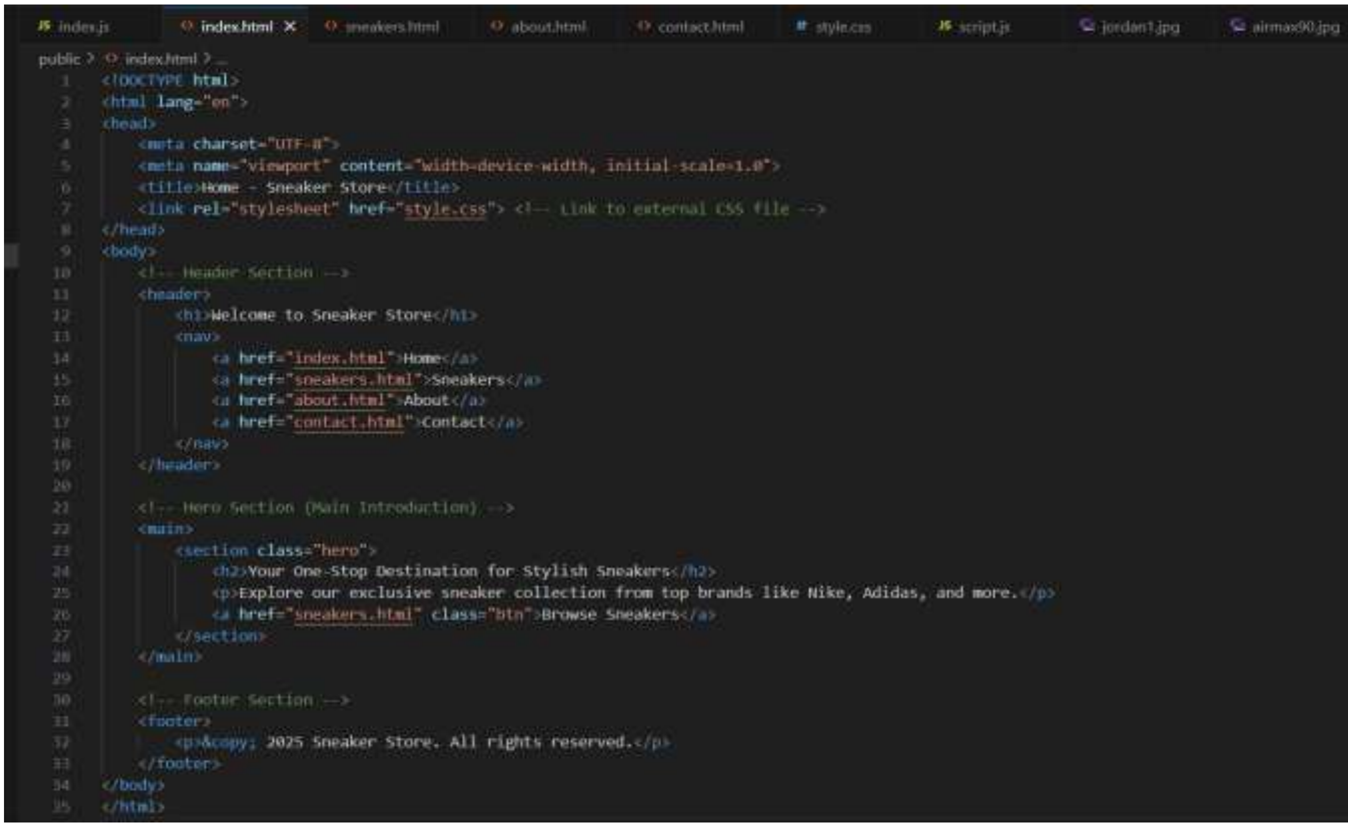
```
// Import necessary modules
const express = require('express'); // Import Express
const fs = require('fs'); // Import file system module to read JSON file
const path = require('path'); // Import path module for working with file paths

// Initialize the Express application
const app = express();
const PORT = 3000; // Define the port where the server will run

// Middleware to serve static files (HTML, CSS, JS, images)
app.use(express.static('public'));

// API Route: Fetch sneaker data from JSON file
app.get('/api/sneakers', (req, res) => {
    // Read the JSON file
    fs.readFile(path.join(__dirname, 'data', 'sneakers.json'), 'utf8', (err, data) => {
        if (err) {
            // Send an error response if there's an issue reading the file
            res.status(500).json({ message: "Error reading data" });
        } else {
            // Send the JSON data as a response
            res.json(JSON.parse(data));
        }
    });
});

// Start the server and listen on the specified port
app.listen(PORT, () => {
    console.log(`Server is running at http://localhost:${PORT}`);
});
```

Step 7: Create Views (HTML Pages)

- index.html (Home Page):
  - Displays sneaker listings fetched from the backend.
  - Links to CSS and JavaScript files.

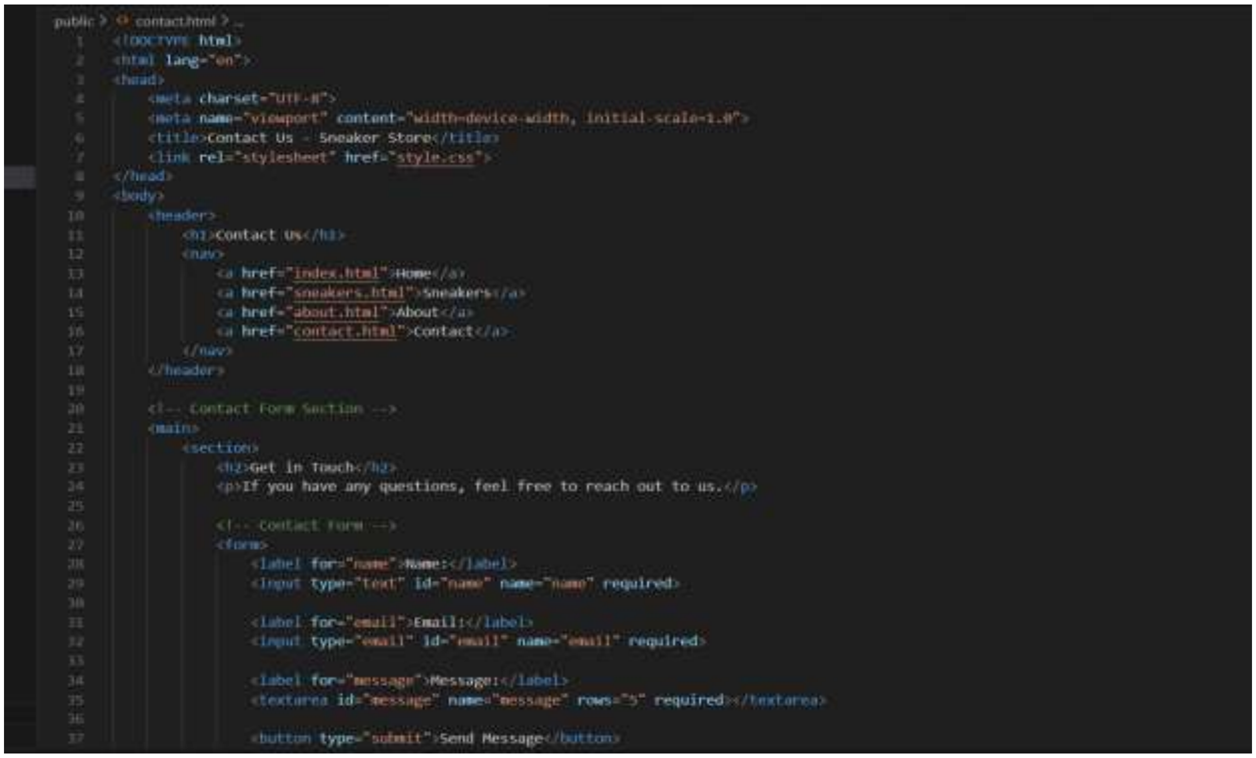
- about.html and contact.html:
    - These pages provide additional information about the store and contact details.

- Index.html



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - Sneaker Store</title>
    <link rel="stylesheet" href="style.css"> <!-- Link to external CSS file -->
</head>
<body>
    <!-- Header Section -->
    <header>
        <h1>Welcome to Sneaker Store</h1>
        <nav>
            <a href="index.html">Home</a>
            <a href="sneakers.html">Sneakers</a>
            <a href="about.html">About</a>
            <a href="contact.html">Contact</a>
        </nav>
    </header>

    <!-- Hero Section (Main Introduction) -->
    <main>
        <section class="hero">
            <h2>Your One-Stop Destination for Stylish Sneakers</h2>
            <p>Explore our exclusive sneaker collection from top brands like Nike, Adidas, and more.</p>
            <a href="sneakers.html" class="btn">Browse Sneakers</a>
        </section>
    </main>

    <!-- Footer Section -->
    <footer>
        <p>&copy; 2025 Sneaker Store. All rights reserved.</p>
    </footer>
</body>
</html>
```

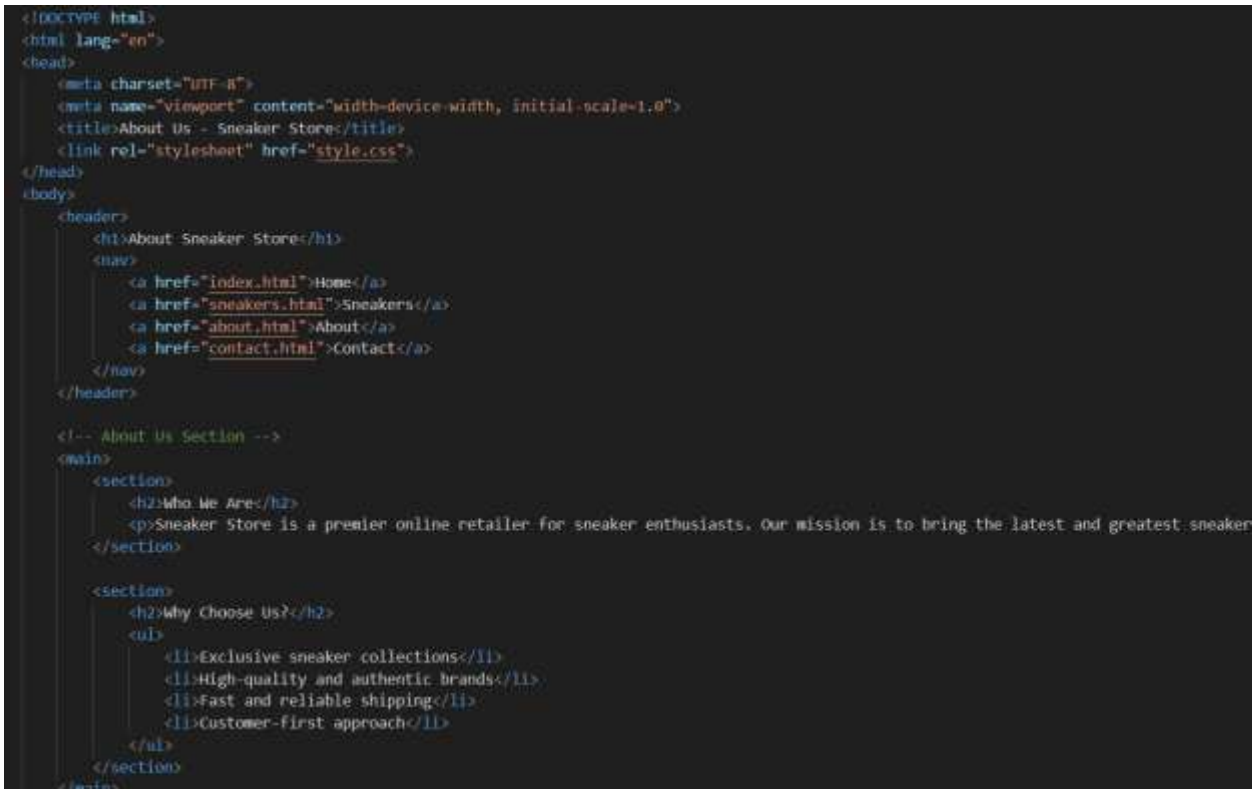
- Contact.html



public > contact.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Us - Sneaker Store</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Contact Us</h1>
        <nav>
            <a href="index.html">Home</a>
            <a href="sneakers.html">Sneakers</a>
            <a href="about.html">About</a>
            <a href="contact.html">Contact</a>
        </nav>
    </header>

    <!-- Contact Form Section -->
    <main>
        <section>
            <h2>Get in Touch</h2>
            <p>If you have any questions, feel free to reach out to us.</p>

            <!-- Contact Form -->
            <form>
                <label for="name">Name:</label>
                <input type="text" id="name" name="name" required>

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

                <label for="message">Message:</label>
                <textarea id="message" name="message" rows="5" required></textarea>

                <button type="submit">Send Message</button>
```

- About.html



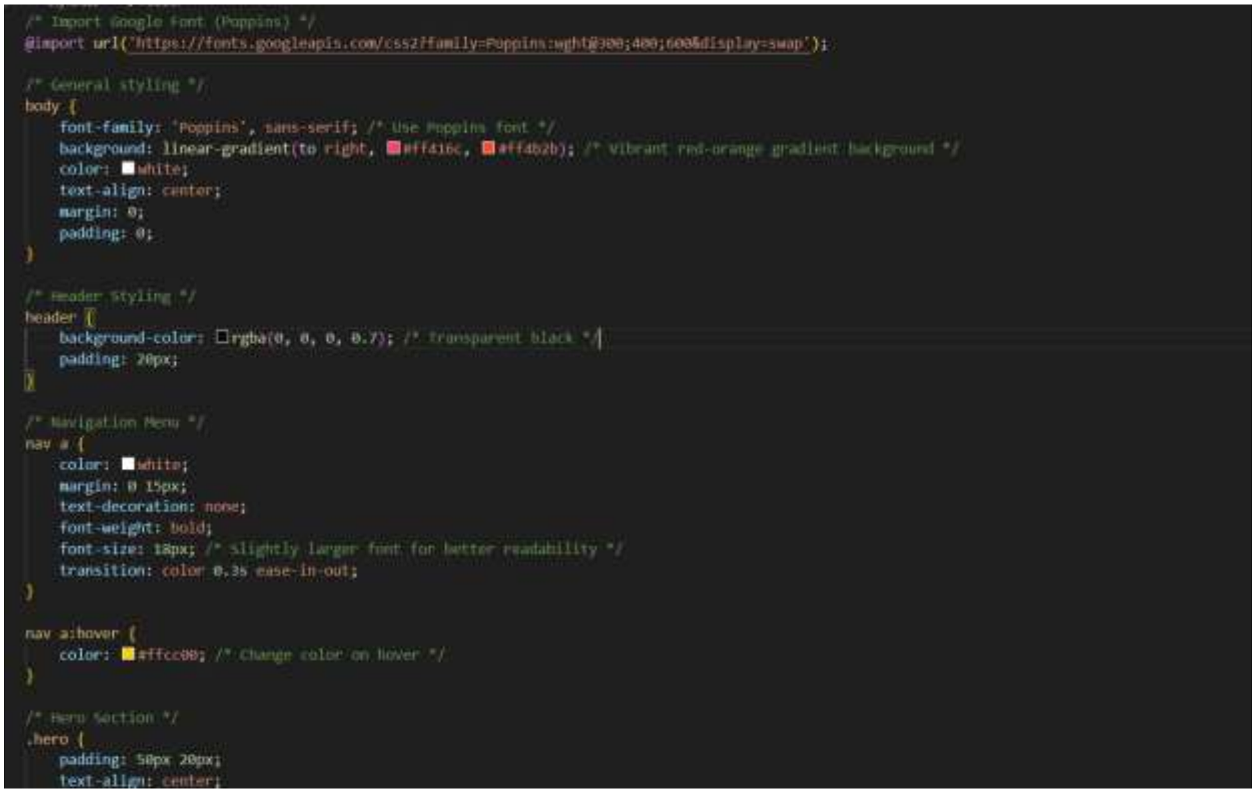
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About Us - Sneaker Store</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>About Sneaker Store</h1>
        <nav>
            <a href="index.html">Home</a>
            <a href="sneakers.html">Sneakers</a>
            <a href="about.html">About</a>
            <a href="contact.html">Contact</a>
        </nav>
    </header>

    <!-- About Us Section -->
    <main>
        <section>
            <h2>Who We Are</h2>
            <p>Sneaker Store is a premier online retailer for sneaker enthusiasts. Our mission is to bring the latest and greatest sneaker
        </section>

        <section>
            <h2>Why Choose Us?</h2>
            <ul>
                <li>Exclusive sneaker collections</li>
                <li>High-quality and authentic brands</li>
                <li>Fast and reliable shipping</li>
                <li>Customer-first approach</li>
            </ul>
        </section>
```

Step 8: Style the Website (CSS)

The CSS will define how the pages and elements look.

Action:

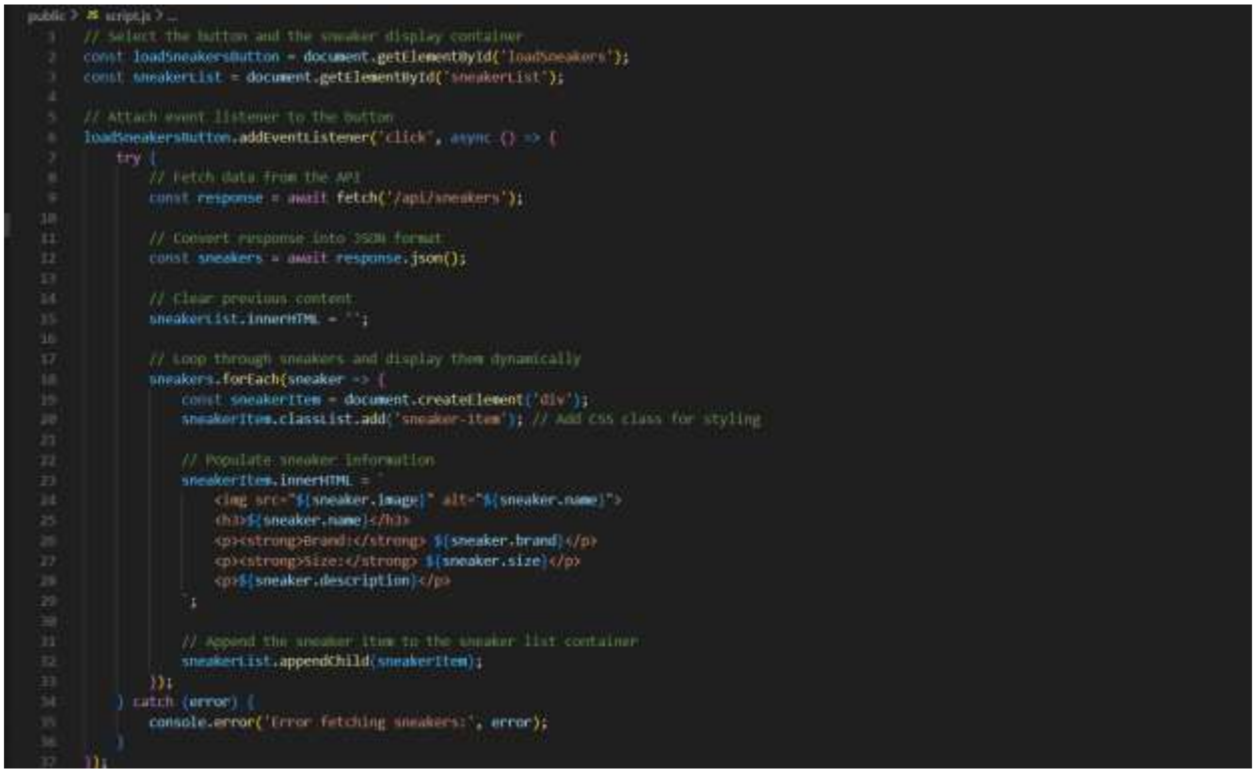
- Inside the public/css/ folder, create a style.css file with the following:



```css
/* Import Google Font (Poppins) */
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&display=swap');

/* General styling */
body {
    font-family: 'Poppins', sans-serif; /* Use Poppins font */
    background: linear-gradient(to right, #ff416c, #ff4b2b); /* Vibrant red-orange gradient background */
    color: white;
    text-align: center;
    margin: 0;
    padding: 0;
}

/* Header Styling */
header {
    background-color: rgba(0, 0, 0, 0.7); /* transparent black */
    padding: 20px;
}

/* Navigation Menu */
nav a {
    color: white;
    margin: 0 15px;
    text-decoration: none;
    font-weight: bold;
    font-size: 18px; /* Slightly larger font for better readability */
    transition: color 0.3s ease-in-out;
}

nav a:hover {
    color: #ffcc00; /* Change color on hover */
}

/* Hero Section */
.hero {
    padding: 50px 20px;
    text-align: center;
```

Step 9: Add Dynamic Sneaker Data in script.js

In public/script.js, fetch the sneaker data from the backend and dynamically display it on the homepage:



public > JS script.js > ...

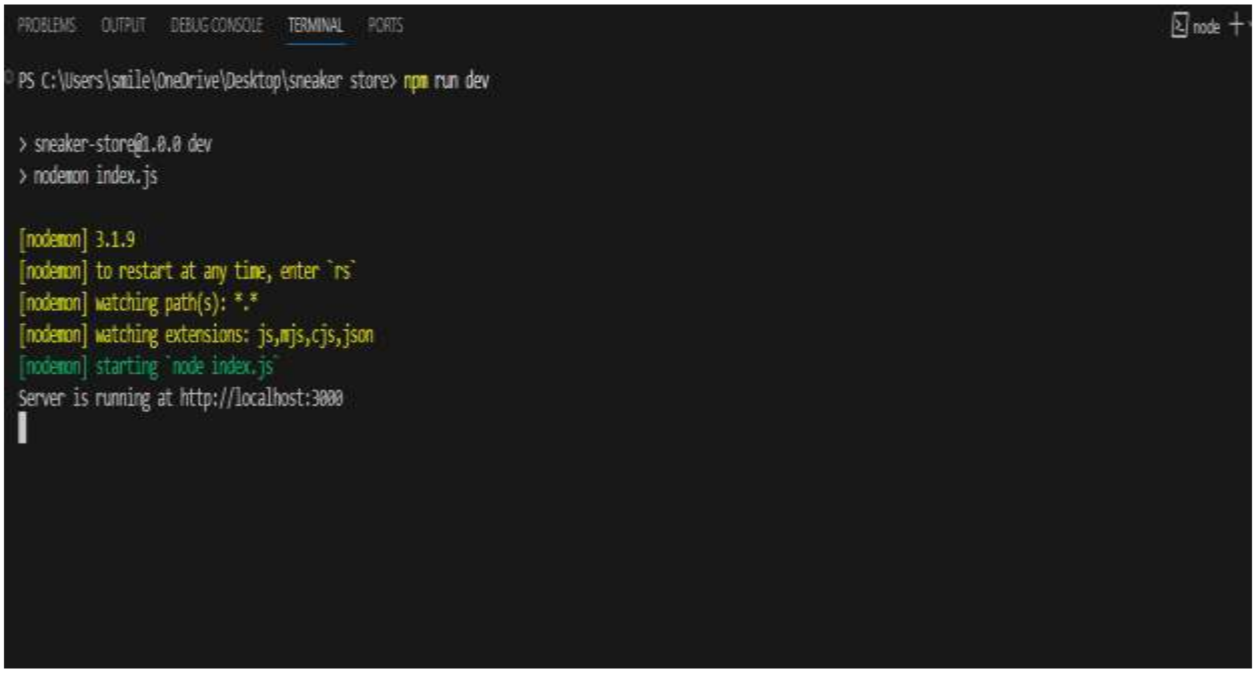
```javascript
// Select the button and the sneaker display container
const loadSneakersButton = document.getElementById('loadSneakers');
const sneakerList = document.getElementById('sneakerList');

// Attach event listener to the button
loadSneakersButton.addEventListener('click', async () => {
    try {
        // Fetch data from the API
        const response = await fetch('/api/sneakers');

        // Convert response into JSON format
        const sneakers = await response.json();

        // Clear previous content
        sneakerList.innerHTML = '';

        // Loop through sneakers and display them dynamically
        sneakers.forEach(sneaker => {
            const sneakerItem = document.createElement('div');
            sneakerItem.classList.add('sneaker-item'); // Add CSS class for styling

            // Populate sneaker information
            sneakerItem.innerHTML = `
                <img src="${sneaker.image}" alt="${sneaker.name}">
                <h3>${sneaker.name}</h3>
                <p><strong>Brand:</strong> ${sneaker.brand}</p>
                <p><strong>Size:</strong> ${sneaker.size}</p>
                <p>${sneaker.description}</p>
            `;

            // Append the sneaker item to the sneaker list container
            sneakerList.appendChild(sneakerItem);
        });
    } catch (error) {
        console.error('Error fetching sneakers:', error);
    }
});
```

Step 10: Run the project

Start the server using:

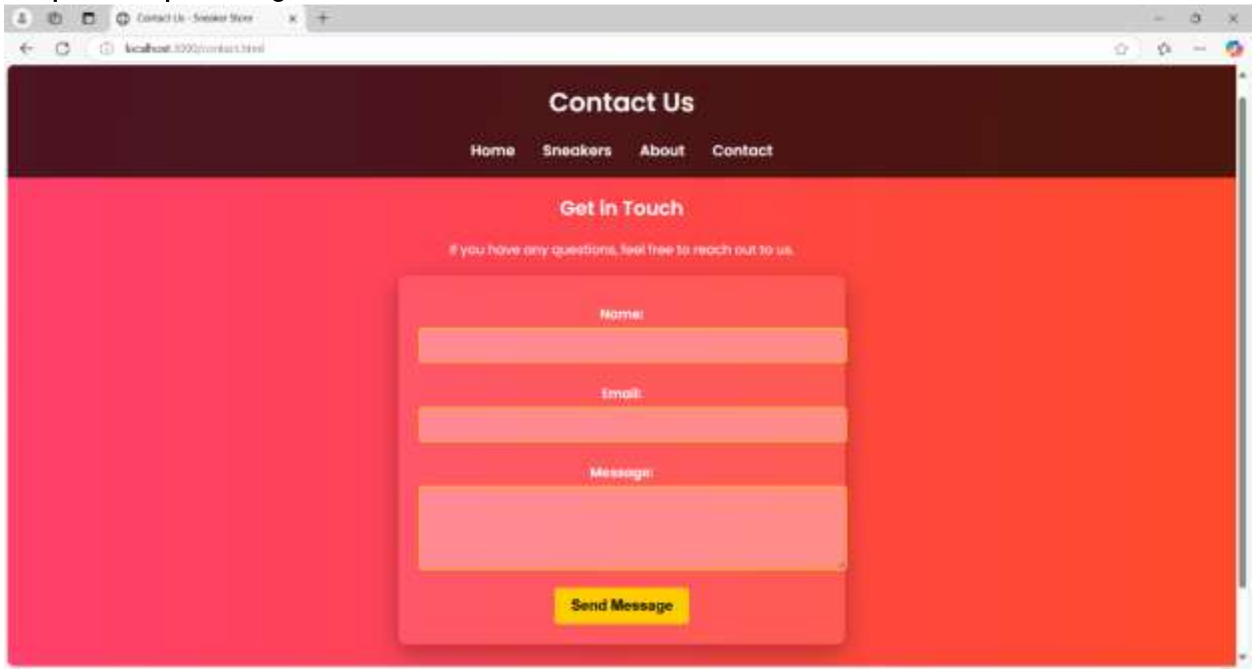
npm run dev

- Open your browser and go to http://localhost:3000 to see the app running.

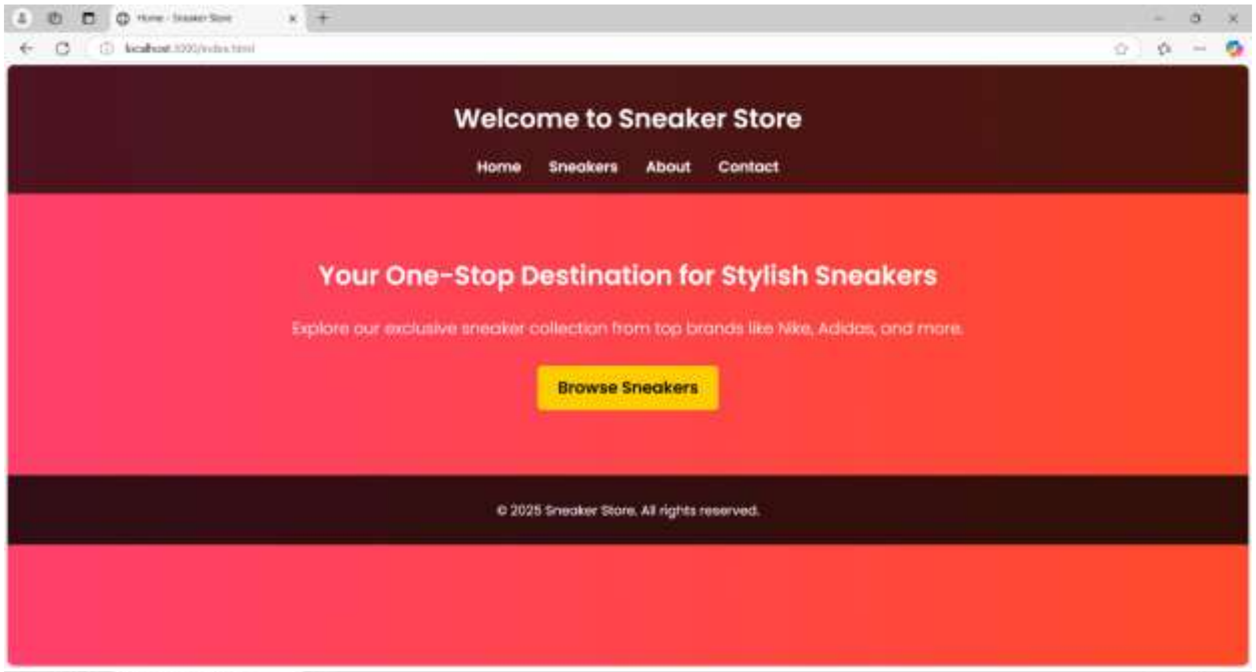


```
PS C:\Users\smile\OneDrive\Desktop\sneaker store> npm run dev

> sneaker-store@1.0.0 dev
> nodemon index.js

[nodemon] 3.1.9
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Server is running at http://localhost:3000
```

Step 11: Output Images:



Contact Us - Sneaker Store
localhost:3000/contact.html
Contact Us
Home
Sneakers
About
Contact
Get in Touch
If you have any questions, feel free to reach out to us.
Name:
Email:
Message:
Send Message



Welcome to Sneaker Store

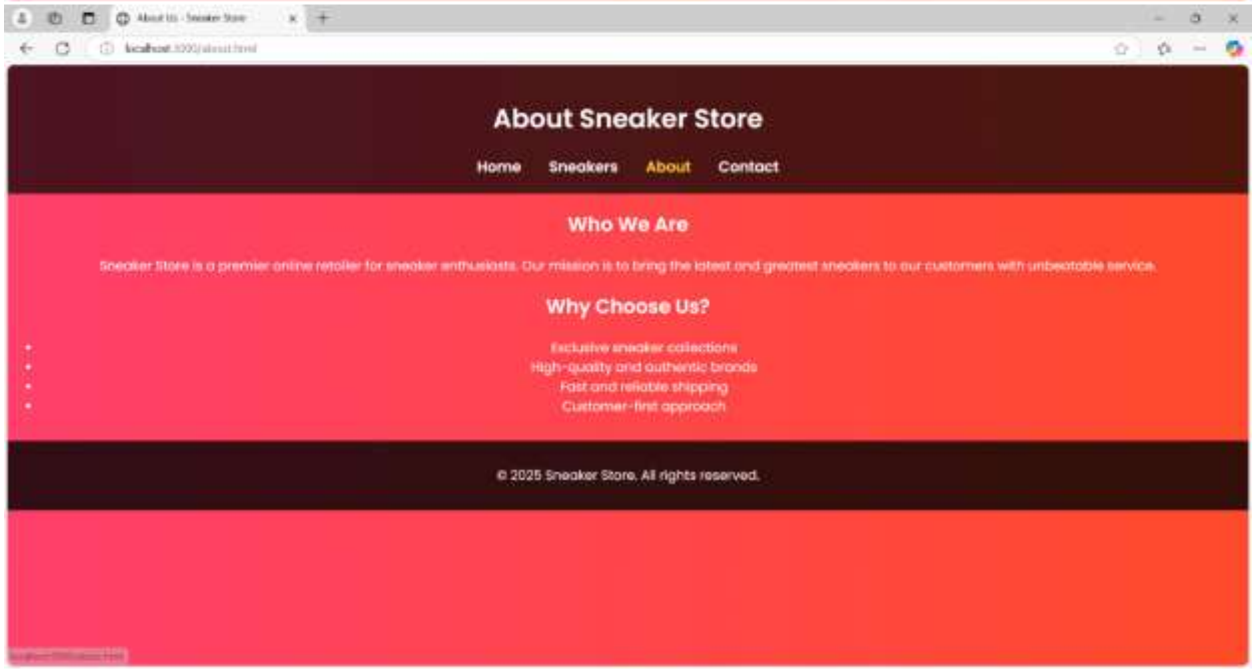
Home Sneakers About Contact

Your One-Stop Destination for Stylish Sneakers

Explore our exclusive sneaker collection from top brands like Nike, Adidas, and more.

Browse Sneakers

© 2025 Sneaker Store. All rights reserved.



About Sneaker Store

Home Sneakers About Contact

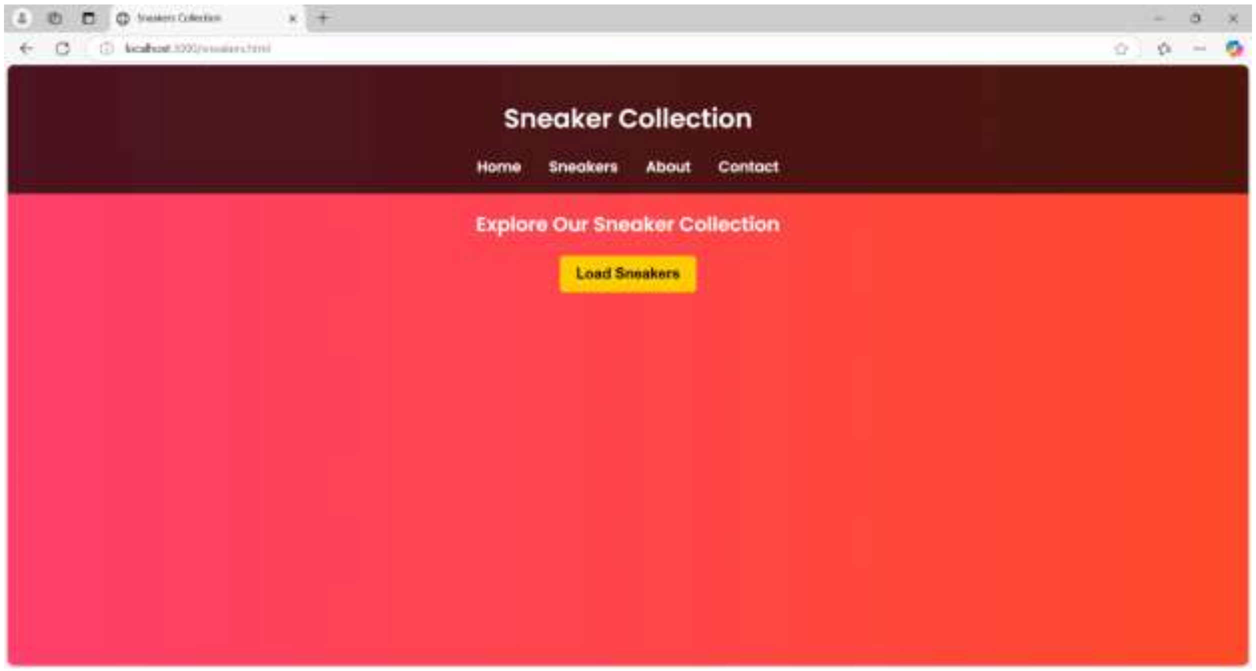
Who We Are

Sneaker Store is a premier online retailer for sneaker enthusiasts. Our mission is to bring the latest and greatest sneakers to our customers with unbeatable service.

Why Choose Us?

- Exclusive sneaker collections
- High-quality and authentic brands
- Fast and reliable shipping
- Customer-first approach

© 2025 Sneaker Store. All rights reserved.



Sneakers Collection

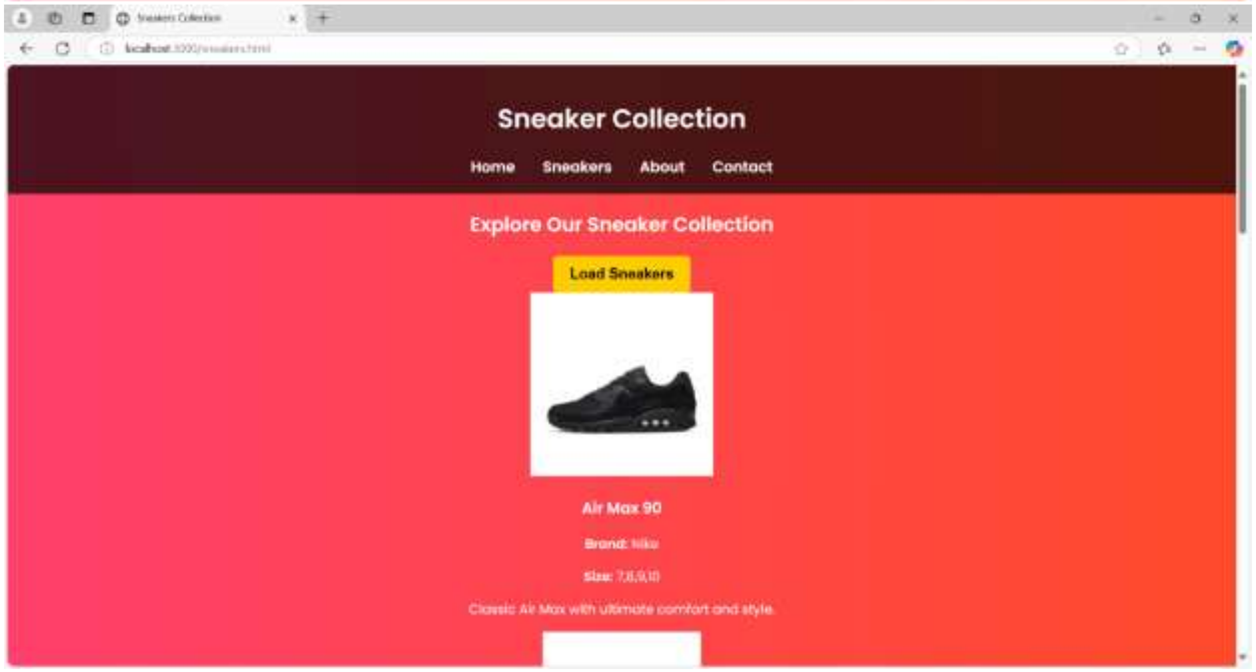
localhost:3000/sneakers.html

Sneaker Collection

Home Sneakers About Contact

Explore Our Sneaker Collection

Load Sneakers



Sneakers Collection

localhost:3000/sneakers.html

Sneaker Collection

Home Sneakers About Contact

Explore Our Sneaker Collection

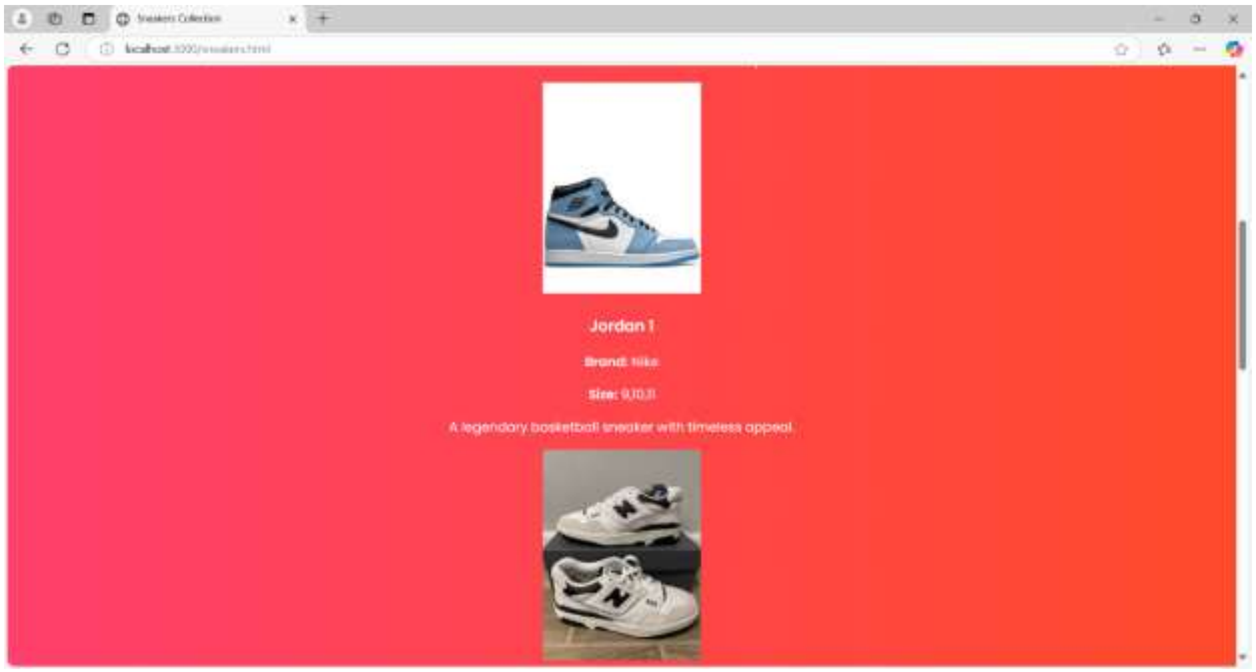
Load Sneakers

Air Max 90

Brand: Nike

Size: 7,8,9,10

Classic Air Max with ultimate comfort and style.

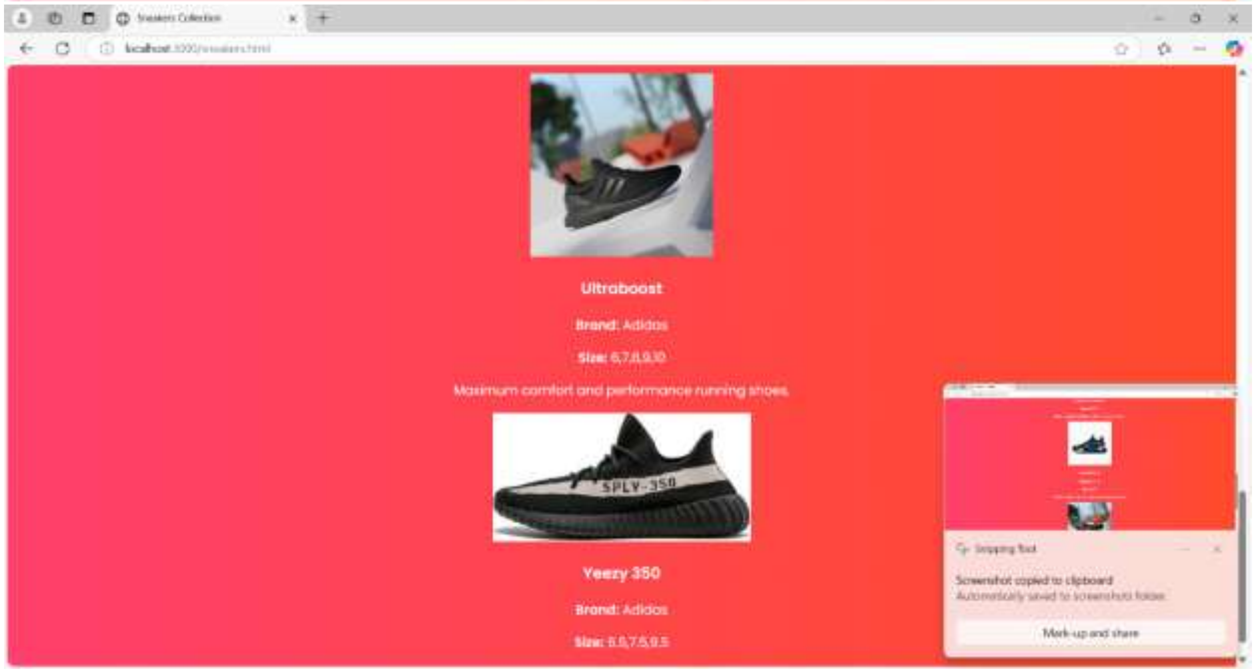


Jordan 1

Brand: Nike

Size: 9,10,11

A legendary basketball sneaker with timeless appeal.



Ultraboost

Brand: Adidas

Size: 6,7,8,9,10

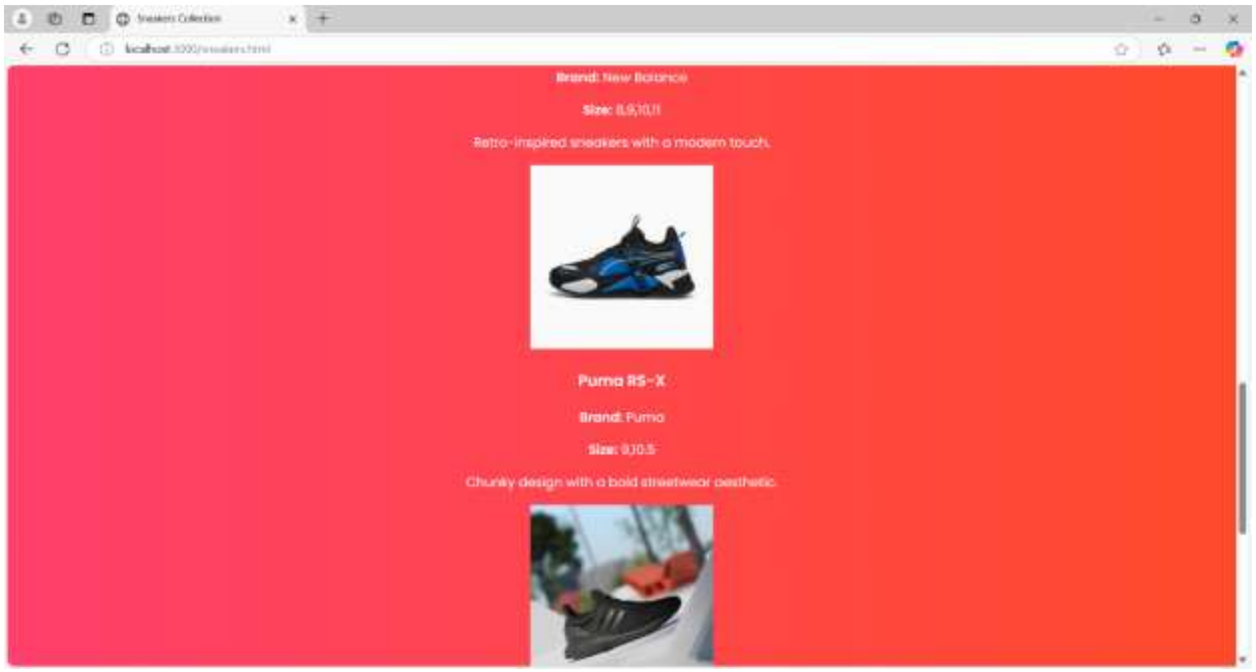
Maximum comfort and performance running shoes.

Yeezy 350

Brand: Adidas

Size: 6.5,7.5,9.5

Screenshot copied to clipboard
Automatically saved to screenshots folder



Brand: New Balance

Size: 8,9,10,11

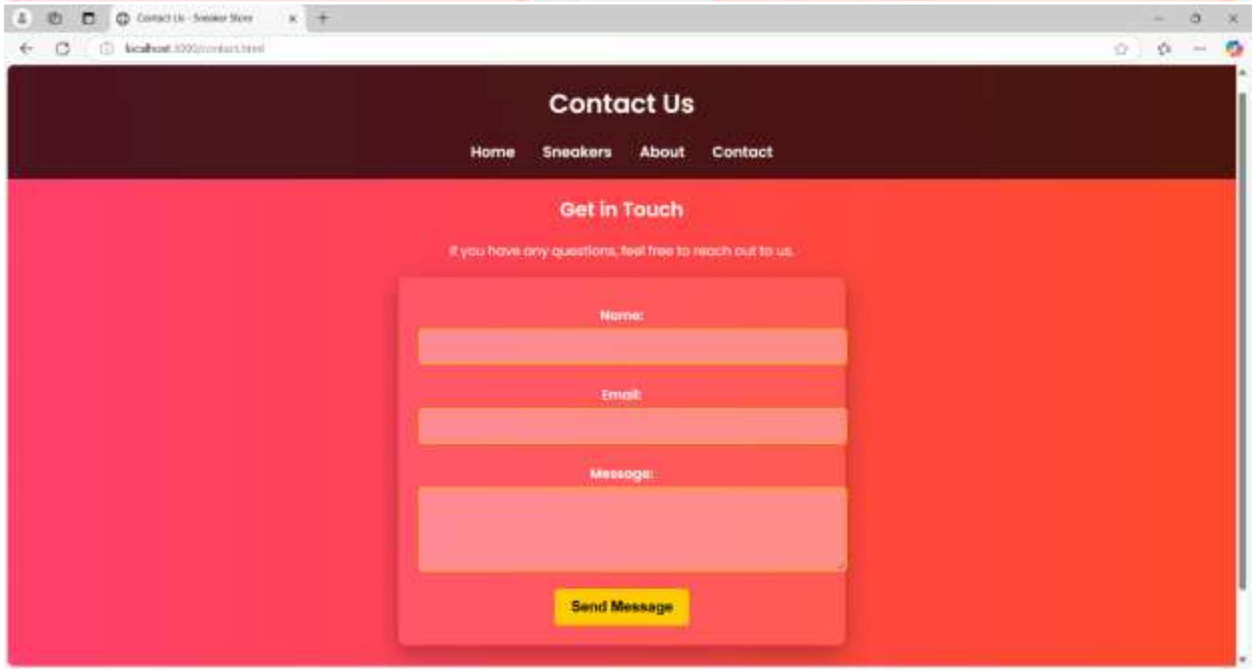
Retro-inspired sneakers with a modern touch.

Puma RS-X

Brand: Puma

Size: 9,10.5

Chunky design with a bold streetwear aesthetic.



Contact Us

Get in Touch

If you have any questions, feel free to reach out to us.

Name:

Email:

Message: