



**PROJECT NAME** - TREKIST E-COMMERCE

**GROUP NAME** - DYNAMIC DEVELOPER

CREATED BY  
2021 - 2024 BATCH



# SAHIBGANJ COLLEGE

SAHIBGANJ, JHARKHAND

A CONSTITUENT UNIT OF SIDO KANHU MURMU UNIVERSITY, DUMKA

A Project Document on

## E-Commerce Website

submitted to **Sahibganj College Sahibganj , JHARKHAND**

(Affiliated to S.K.M.U)

# Bachelor of Computer Application

Submitted by

**Fazil, Saurav, Aaishan, Asif, Prince, Subham, Manmohan & Shahnawaz**

**SESSION 2021 - 2024**

**Team Member's Name and Roll No.**

S.N	Name	Roll No.	Registration No.
1	Md Fazil Abu Bakar	216041342499	SKMU2021031653
2	Saurav Kumar	216041342518	SKMU2021031651
3	Aaishan Ahmad	216041342488	SKMU2021031654
4	Asif Raza	216041342498	SKMU2021031668
5	Prince Jha	216041342508	SKMU2021031641
6	Subham Kumar	216041342521	SKMU2021031667
7	Manmohan Reshi	216041342497	SKMU2021031315
8	Md Shahnawaz	216041342501	SKMU2021031656

**PROJECT GUIDE**

**SIGN OF INVIGILATOR**

## ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessings with his grace and taking our endeavor to a successful cumulative.

I extend my BCA and heartfelt thanks to our esteemed guide, **Prof. Rajiv Ranjan**, for providing me with the right guidance and advice at the crucial junctures and for showing me the right way.

I would like to thank the other faculty members also, at thus occasion. Last but not the least, I Would like to thank my friends and family for the support and encouragement they have given

TABLE OF CONTENTS:

## INTRODUCTION

The e-commerce website "**Trekist**" is a project aimed at providing an online platform for the sale of Trekist company shoes.

Project Title	Trekist A Ecommerce website
Project Description	Web Application, It is a website
Project Duration	2 Months
Project Guide	Rajiv Ranjan
Platform	Window 10
Technologies	HTML, CSS, JavaScript, Node Js & Mongodb
Tools Used	Microsoft Visual Studio, Chrome

The website is designed to be simple, user-friendly, and efficient, allowing customers to browse and purchase shoes without needing to create an account. The product listings are dynamically fetched from a MongoDB database, ensuring up-to-date information on the available products. Users can select a shoe, fill in their details on the provided form, and place an order. The backend, powered by Node.js, processes these orders and stores the details in the database. An order ID is generated for each purchase and returned to the user, allowing them to track their order status using local storage.

This project was developed over a duration of two months, under the guidance of **Rajiv Ranjan**, and aims to simulate the essential functionalities of a modern e-commerce platform. It uses a combination of technologies such as **HTML, CSS, JavaScript, Node.js**, and **MongoDB** to deliver a seamless user experience. The platform is built on **Windows 10**, and development was

carried out using tools such as **Microsoft Visual Studio** and **Google Chrome** for testing and debugging.

The project serves as a foundational step toward building more complex and feature-rich web applications in the future

### **Features:**

**No Login Required:** Users can purchase shoes without the need to create an account, allowing for a hassle-free and seamless shopping experience.

**Shoe Listings:** All available shoes are dynamically fetched from the MongoDB database and displayed on the website in the form of product cards. This ensures that users have access to up-to-date product information.

**Filter Options:** To enhance the user experience, the website offers filter options where users can narrow down their choices by shoe types, such as casual, formal, sneakers, etc., making it easier to find exactly what they're looking for.

**My Like:** Users can "like" shoes, and these liked shoes are saved in the My Like section, allowing users to easily revisit and access the shoes they are interested in purchasing later.

**Order Process:** Users can select any shoe and click on the "Buy" button. Upon selection, a form appears to collect essential buyer details. Once the form is submitted, the backend (powered by Node.js) processes the order and stores the data in MongoDB. An order ID is generated, sent back to the frontend, and stored in local storage. The user can then retrieve their order details and track the status using the order ID.

**Collections on Home Page:** The homepage also highlights collections of our best-selling, most popular shoes, and shoes categorized by type, such as sneakers, casual, and formal wear. This provides an intuitive and engaging way for customers to explore our top products.

## Scope of the Project: Trekist Website

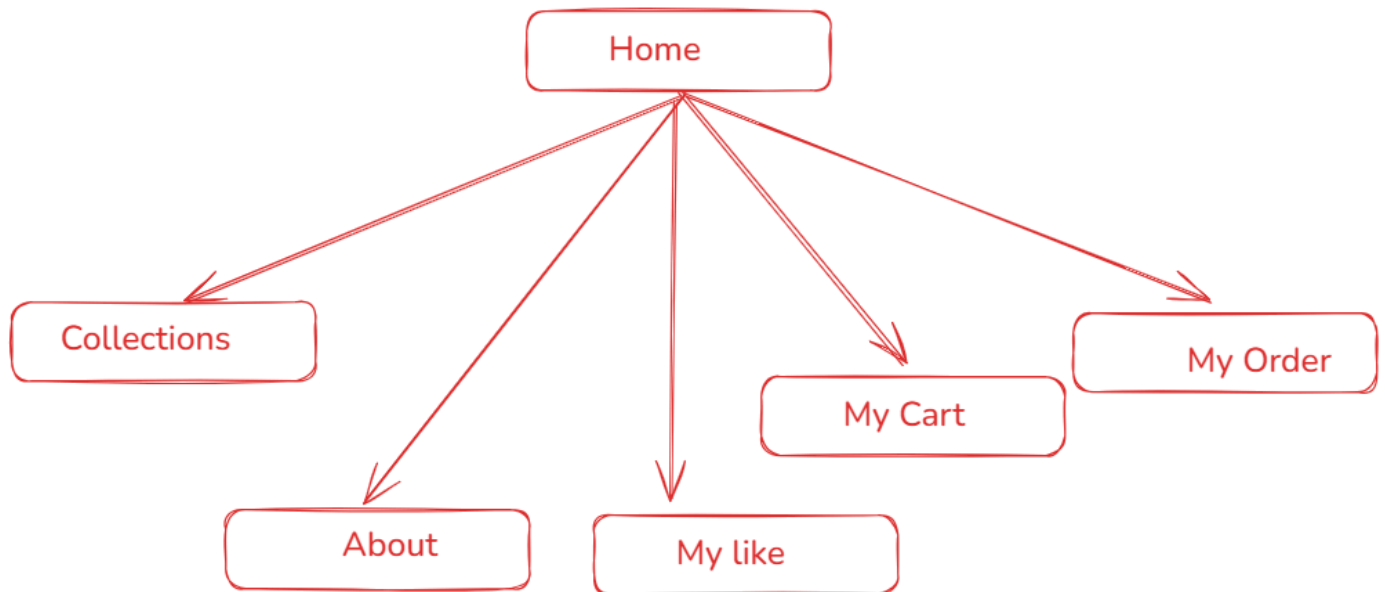
The **Trekist** e-commerce website project aims to create a streamlined, user-friendly platform for purchasing Trekist company shoes online. It provides an efficient shopping experience by simplifying the purchase process and allowing users to browse and buy shoes without the need for an account. The scope of this project covers several key functionalities and future development areas:

1. **Product Display and Filtering:** The website efficiently fetches shoe data from a MongoDB database and displays it in an organized manner. Users can filter shoes based on categories such as casual, formal, sneakers, and more, enabling them to quickly find what they need.
2. **Seamless Purchase Process:** The order placement process is smooth and intuitive, requiring users to fill out a simple form after selecting a product. Orders are processed in real-time, with details securely stored in the backend.
3. **Personalization:** Users can engage with the site by "liking" products and saving them in the **My Like** section for future reference, improving the overall shopping experience.
4. **Collections and Highlights:** The homepage showcases best-selling shoes, popular items, and shoes categorized by type. This increases customer engagement by offering featured collections.
5. **Future Expansion:** In the future, the scope of the project can expand to include additional features such as user authentication, payment gateway integration, and more detailed order history tracking.

## WORKING OF PROJECT

### 1. HOME PAGE: -

Visiting the home page on screen. On the top right side navigation bar including other options Home , Collection , about, MyLike(icon), Mycart(icon), Myorder(icon), and below the header main page contain shoes with filter option side bar .



### 2.Collection: -

Collection contain Most popular , best seller SHoes.

### 3.About: -

About the website and company also provided contact

### 4.MyLike: -

Can see your shoes Which you liked..

### 5.MyCart : -

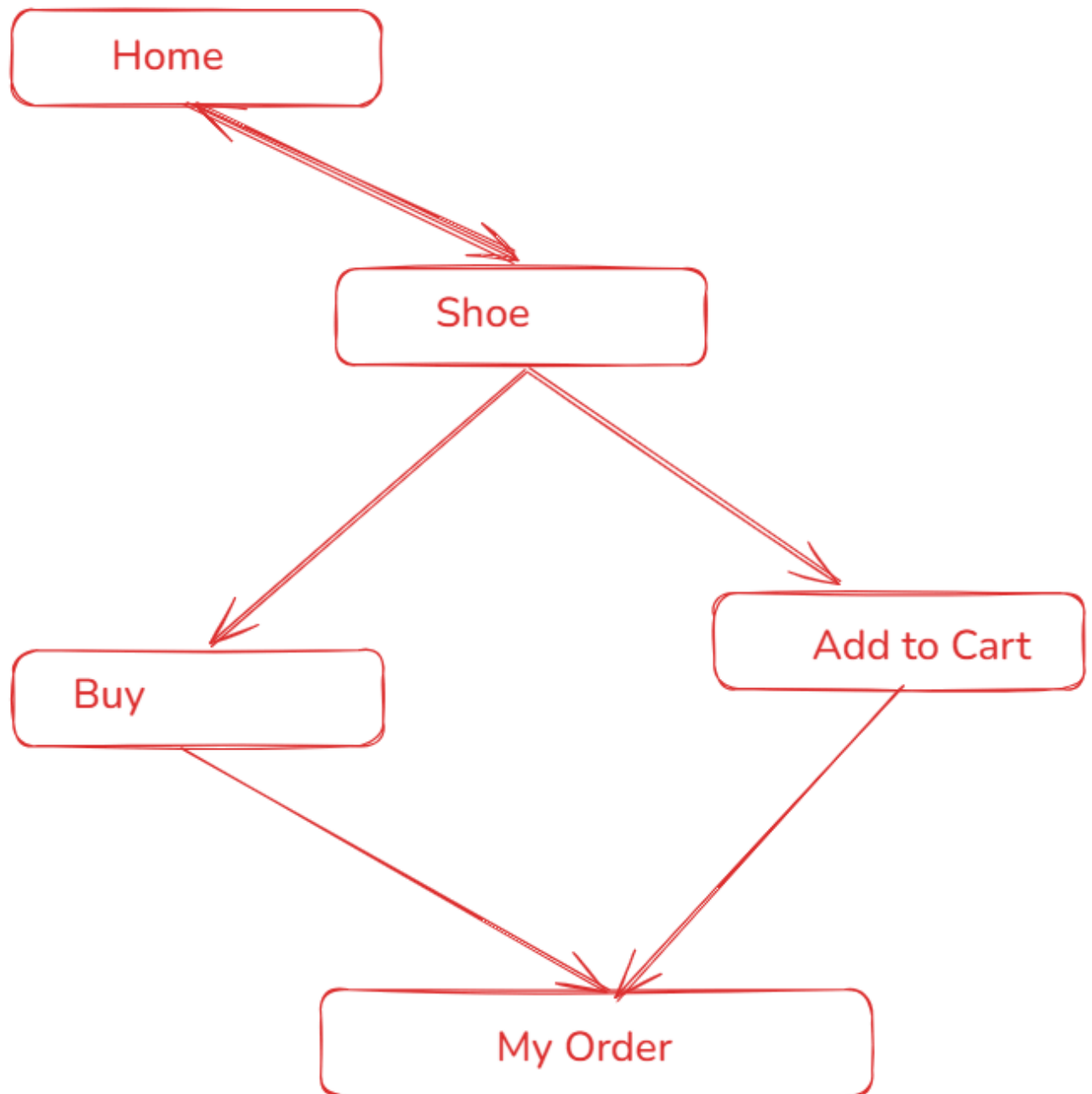
Can see your Shoes which a dded Cart .

### 6.MyOrder : -

See Your Order details

## 7. Shoes Profile :-

Shoes details and add to cart and Buy Option





# REQUIREMENT AND ANALYSIS

## 1. Functional Requirements

- **User Interface:** The website must provide a user-friendly interface that allows customers to browse, filter, and purchase shoes with minimal effort. Key sections include product listings, filtering options, and a simple order form for purchasing shoes.

- **Product Display:** Shoes should be dynamically fetched from the MongoDB database and displayed with their image, price, description, and a "Buy" button. Users should be able to filter shoes based on categories like casual, formal, sneakers, etc.

- **Order Placement:** Once a shoe is selected, the website should provide a form to collect customer details such as name, address, and shoe size. Upon submission, the order data must be processed, stored in the backend (Node.js and MongoDB), and an order ID should be generated and returned to the user.

- **Order Tracking:** The generated order ID should be stored in the browser's local storage, allowing users to track their order status by retrieving this data.

- **Collections:** The homepage should feature best-selling, most popular, and type-wise shoe collections to engage users and make the browsing experience more personalized.

- **My Like Feature:** Users should be able to "like" shoes, and these liked products should be stored and displayed in the **My Like** section for future reference.

## 2. Non-Functional Requirements

- **Performance:** The website must load efficiently and quickly fetch data from the MongoDB database. It should handle multiple users simultaneously without slowing down.

- **Security:** User data, including order details, must be securely handled by the backend, with secure communication between frontend and backend servers. Although no payment is required in this version, future enhancements should consider secure payment handling.

- **Reliability:** The website must be available and functional at all times, ensuring that orders are processed and stored accurately in the database without failures.

- **Usability:** The design must be intuitive, and the user experience should ensure that visitors can easily navigate the website, filter products, and place orders without confusion.

## 3. Tools and Technologies

- **Frontend:** HTML, CSS, JavaScript
  - The frontend is responsible for rendering product data, displaying the user interface, and handling form submissions.
- **Backend:** Node.js
  - The backend processes the orders and communicates with the MongoDB database to store and retrieve data.
- **Database:** MongoDB
  - MongoDB is used to store shoe details, customer orders, and any other relevant data for order processing.
- **Development Tools:** Microsoft Visual Studio, Chrome DevTools
  - These tools are used for development, debugging, and testing of the website.

#### 4. Analysis

The **Trekist** website is designed to meet the growing demand for online shopping, particularly focusing on shoes from the Trekist company. The project provides users with an easy-to-use platform for browsing and buying products, while keeping the technical complexity hidden behind a simple interface. The inclusion of features like product filtering, "My Like" section, and local storage-based order tracking enhances user engagement and satisfaction.

The use of Node.js for backend processing ensures that the system can scale, and MongoDB provides a flexible database solution that can handle large amounts of product and order data. The current design and functionality meet the requirements for a basic e-commerce website, with the potential to be expanded in the future to include more advanced features like user authentication, payment systems, and enhanced order history tracking.

## **Technologies and tools used in Project**

### **WINDOWS 10:**

Windows 10 is a widely used operating system developed by Microsoft. It is the successor to Windows 8.1 and was released in July 2015. Windows 10 is designed to work across a variety of devices, including PCs, tablets, smartphones, and embedded systems. It introduced several new features compared to its predecessors, such as the Cortana virtual assistant, the Microsoft Edge web browser, a revamped Start menu, and the ability to run Universal Windows Platform apps that can adapt to different device types. Windows 10 also includes regular updates and security patches to improve performance and address potential vulnerabilities.

Windows 10 has been well-received for its user interface improvements, performance enhancements, and focus on modern computing trends. It is widely used by consumers, businesses, and organizations around the world.

### **HTML:**

HTML, which stands for HyperText Markup Language, is the standard markup language used to create web pages. It is the basic building block of web development and is used to structure content on the web, such as text, images, videos, and links. HTML uses a system of tags to define the structure and layout of a web page, with each tag representing different elements like headings, paragraphs, lists, links, and more.

HTML documents are interpreted by web browsers to render the content and display it to users. Along with CSS (Cascading Style Sheets) for styling and JavaScript for interactivity, HTML forms the core technologies used to create websites and web applications.

It provides a set of tags that structure the content and layout of a web page, allowing developers to define headings, paragraphs, lists, links, images, and other elements. HTML documents are interpreted by web browsers to display the content to users.

### **CSS:**

CSS, which stands for Cascading Style Sheets, is a style sheet language used to

describe the presentation of a document written in HTML or XML (including XML dialects such as SVG or XHTML). It defines how elements are displayed on a screen, in print, or in other media. CSS separates the content of a web page (defined in HTML) from its presentation, allowing developers to control the layout, colors, fonts, and other visual aspects of a website.

CSS works by selecting HTML elements and applying style properties to them. For example, you can use CSS to set the color and size of text, define the layout of a page, create animations, and make the website responsive to different screen sizes.

CSS is an essential part of web development and is used alongside HTML and JavaScript to create visually appealing and interactive web pages.

## **JAVASCRIPT:**

JavaScript is a high-level programming language that is primarily used to create interactive effects within web browsers. It is one of the core technologies of the World Wide Web, along with HTML and CSS. JavaScript allows developers to add dynamic behavior to their websites, such as responding to user actions, manipulating the DOM (Document Object Model), and making asynchronous requests to web servers.

### **Key features of JavaScript include:**

1. **Client-Side Scripting:** JavaScript code is executed on the client's browser, which allows for dynamic content generation and manipulation without requiring server interaction.
2. **Object-Oriented:** JavaScript is an object-oriented language, which means it uses objects and their properties and methods to organize code.
3. **Event-Driven:** JavaScript is often used to create event-driven applications, where code is executed in response to certain events, such as a user clicking a button or submitting a form.
4. **Asynchronous Programming:** JavaScript supports asynchronous programming through features like callbacks, promises, and `async/await`, allowing developers to write non-blocking code that can handle multiple operations simultaneously.
5. **Cross-Platform:** JavaScript is supported by all major web browsers, making it a cross-platform language for web development.

JavaScript is a versatile language that is not only used for web development

but also for server-side development (Node.js), mobile app development (React Native, Ionic), game development, and more.

### **Node.js:**

Node.js is a runtime environment that allows developers to run JavaScript code on the server side. It is built on the V8 JavaScript engine and is widely used for building scalable, real-time web applications. Node.js allows for asynchronous programming, making it highly efficient for handling multiple simultaneous requests.

In the **Trekist** project, Node.js was used for backend development, processing user orders, handling form data, communicating with the MongoDB database, and returning the order ID to the frontend. Its lightweight, efficient architecture made it the ideal choice for the backend of the e-commerce site.

### **MongoDB:**

MongoDB is a NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). Unlike traditional relational databases, MongoDB uses collections and documents instead of tables and rows, making it easier to scale and handle large amounts of data.

In the **Trekist** project, MongoDB was used to store product details, such as shoe listings, and order information, including customer details and order statuses. Its flexibility and scalability make it suitable for e-commerce platforms that need to handle dynamic data efficiently.

### **Microsoft Visual Studio:**

Microsoft Visual Studio is an integrated development environment (IDE) used for coding, debugging, and managing projects. It supports multiple programming languages and frameworks, providing a rich set of tools for web development.

For the **Trekist** project, Visual Studio was used to write and manage the frontend and backend code, making it easier to develop and debug the website efficiently.

### **CHROME:**

Chrome, short for Google Chrome, is a popular web browser developed by Google. It was first released in 2008 and has since become one of the most widely used web browsers, competing with other major browsers like Mozilla Firefox, Microsoft Edge, and Apple Safari.

Key features of Google Chrome include:

1. **Speed and Performance:** Chrome is known for its fast performance, quick startup times, and efficient use of system resources.
2. **Synchronization:** Chrome allows users to sign in with a Google account, which enables synchronization of bookmarks, history, and settings across multiple devices.
3. **Security:** Chrome includes built-in features such as Safe Browsing, which helps protect users from phishing and malware, as well as automatic updates to ensure users have the latest security patches.
4. **Customization:** Chrome supports a wide range of extensions and themes, allowing users to customize their browsing experience with additional features and visual changes.
5. **Developer Tools:** Chrome includes powerful developer tools that help web developers debug and profile their web applications, inspect HTML and CSS, and analyze network performance.
6. **Cross-Platform:** Chrome is available on multiple platforms, including Windows, macOS, Linux, Android, and iOS, providing a consistent browsing experience across devices.

Google Chrome has a large market share in the web browser market due to its speed, simplicity, and integration with other Google services. It is also known for its support of modern web standards and its focus on providing a secure and stable browsing experience.

## Source Code

### Index.HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Trekist</title>
    <link rel="stylesheet" href="/index.css" />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
    <link
      href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400&display=swap"
      rel="stylesheet"
    />
    <link
      href="https://fonts.googleapis.com/css2?family=Matemasie&display=swap"
      rel="stylesheet"
    />
    <link
      href="https://fonts.googleapis.com/css2?family=Bungee+Tint&display=swap"
      rel="stylesheet"
    />

    <link rel="icon" type="image/x-icon" href="/icon.ico" />
  </head>
  <body>
    <!--header=====-->
    <header>
      <div>
        <h1>Trekist</h1>
```

```
</div>
<div>
  <nav>
    <li><a href="./index.html">Home</a></li>
    <li><a href="#collection">Collection</a></li>
    <li><a href="#about">About</a></li>
    <li>
      <a href="./mylike.html"
        ></a>
    </li>
    <li>
      <a href="./mycart.html"
        ></a>
    </li>
    <li>
      <a href="./myorder.html"
        ></a>
    </li>
  </nav>
</div>
</header>
```

```
<!--shoes=====-->
<div id="shoes">
  <div id="left">
    <li onclick="filter_apply('all')">All</li>
    <li onclick="filter_apply('casual')">casual</li>
    <li onclick="filter_apply('sports')">sports</li>
    <li onclick="filter_apply('formal')">formal</li>
```



```

    <li onclick="filter_apply('training')">training & Gym</li>
    <li onclick="filter_apply('sneakers')">sneakers</li>
    <li onclick="filter_apply('special')">special addition</li>
</div>

<div id="right" onclick="hello()">
    <div class="loader"></div>
</div>
</div>

<!--main=====-->
<main>
    <div>
        
    </div>
    <div>
        <h2>smart shoes</h2>
        <br />
        <p>
            Lorem ipsum dolor sit amet consectetur adipisicing elit. Impedit
            debitis, quae eum, magnam eos soluta nemo tempore illum rem doloremque
            aliquam repellat sed culpa sapiente aspernatur distinctio, eligendi
            nihil voluptates.
        </p>
    </div>
</main>

<!--collection=====-->
<div id="collection_heading">COLLECTION</div>
<div id="collection">
    <!--most_popular----->
    <!-- <div class="box">
<div class="heading">Most popular</div>
<div class="container" >

```

```
<li class="sub_box">
  
  <div class="collection_name_box">
    <p class="collection_name">Trekist Shoes </p>
    <p class="collection_price">MRP : 0000 ₹</p>

  </div>
</li>
<li class="sub_box">
  
  <div class="collection_name_box">
    <p class="collection_name">Trekist Shoes </p>
    <p class="collection_price">MRP : 0000 ₹</p>
  </div>
</li>
<li class="sub_box">
  
  <div class="collection_name_box">
    <p class="collection_name">Trekist Shoes </p>
    <p class="collection_price">MRP : 0000 ₹</p>
  </div>
</li>
<li class="sub_box">
  
  <div class="collection_name_box">
    <p class="collection_name">Trekist Shoes </p>
    <p class="collection_price">MRP : 0000 ₹</p>
  </div>
</li>
</div>
-->
```

</div>

<!--animation=====-->

<div

style="

width: 100%;

height: 200px;

overflow-x: hidden;

display: flex;

background-color: rgb(230, 230, 230);

"

>









```

```

```

```

```

```

```

```

```
</div>
```

```
<!--footer=====-->
```

```
<footer id="about">
```

```
<div id="footer-left">
```

```
<p>
```

Trekist – where innovation meets style. Our premium shoes are crafted  
for comfort and built to last. Step into excellence with every pair.

Explore our collection and elevate your journey."

```
</p>
```

```
</div>
```

```
<div id="footer-right">
  <li>trekist.shoes@gmail.com</li>
  <li>sahibganj , jharkhand ,816109</li>
  <li>give - feedback</li>
</div>

<div id="footer-bottom">
  <div id="footer-social">
    <a href="https://www.linkedin.com/in" class="footer-social-icon">
      </a>
    <a href="https://www.instagram.com/" class="footer-social-icon">
      </a>
    <a
      href="https://mail.google.com/mail/u/0/"
      class="footer-social-icon"
    >
      </a>
    <a href="https://x.com/" class="footer-social-icon">
      </a>
    </div>
    <div id="footer-copyright">
      Trekist pvt.limited all copyright © unresloved
    </div>
  </div>
</footer>

<!--javascript=====-->
```

```
<script src="./index.js" type="text/javascript"></script>
<script src="./shoes.js" type="text/javascript"></script>
</body>
</html>
```

### **index.js**

```
async function fetchCollections() {
  try {
    const response = await fetch("https://trekist.onrender.com/collections");
    if (!response.ok) {
      throw new Error("Network response was not ok");
    }
    const products = await response.json();

    return products;
  } catch (error) {
    console.error("There has been a problem with your fetch operation:", error);
  }
}

async function collection_load() {
  const data = await fetchCollections();

  data.forEach((element) => {
    console.log(element.name);
    var box = document.createElement("div");
    box.className = "box";
    box.innerHTML = `<div class="heading"> ${element.name}</div>`;
    var container = document.createElement("div");
    container.className = "container";
    box.appendChild(container);
```

```

for (let i = 0; i < 4; i++) {
    var sub_box = document.createElement("li");
    sub_box.className = "sub_box";
    sub_box.addEventListener("click", () => {
        localStorage.setItem("profile_data", JSON.stringify(element.shoes[i]));
        window.location.href = "./profile.html";
    });
    sub_box.innerHTML = `

    <div class="collection_name_box">
        <p class="collection_name">${element.shoes[i].name}</p>
        <p class="collection_price"> MRP : ${element.shoes[i].price}</p>

    </div>`;
    box.lastElementChild.appendChild(sub_box);
}

document.getElementById("collection").appendChild(box);
});
}

collection_load();

```

### index.css

```

@import url("./css/menu.css");
@import url("./css/header.css");
@import url("./css/shoes.css");
@import url("./css/main.css");
@import url("./css/collection.css");
@import url("./css/footer.css");

/*UNIVERSAL=====*/

```

```
* {  
  margin: 0px;  
  padding: 0px;  
  color: white;  
}
```

### menu.js

```
let switch_ = 0;  
  
function menu_switch() {  
  if (switch_ == 0) document.getElementById("menu").style.display = "flex";  
  else document.getElementById("menu").style.display = "none";  
}
```

### mycart.html

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8" />  
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
    <title>My Cart</title>  
    <link rel="stylesheet" href="./css/mycart.css" />  
    <link  
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400&display=swap"  
    rel="stylesheet"  
  />  
</head>  
<body>  
  <h2  
    style="
```



```

    color: red;
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
        sans-serif;
    padding: 15px;
    background-color: black;
    width: calc(100% - 30px);
    height: 40px;
    "
>
    MY CART
</h2>
<div id="cart">
    <div id="cart-products">
        <div id="cart-products-box"></div>
        <div id="cart-products-order">
            <button id="buy-cart">Place Order</button>
        </div>
    </div>
    <div id="cart-price">
        <div id="cart-price-box"></div>
    </div>
</div>
</body>

<script>
    let mycart = JSON.parse(localStorage.getItem("mycart"));
    let total_price = 0;

    if (mycart != null && mycart != [] && mycart.length != 0) {
        for (let i = 0; i < mycart.length; i++) {
            let cart_shoe = document.createElement("div");
            cart_shoe.className = "cart-shoe";

```

```

cart_shoe.innerHTML = `
<div class="cart-shoe-details">
    <div class="cart-shoe-name">${mycart[i].name}</div>
    <div class="cart-shoe-size">Size : ${mycart[i].set_size}</div>
    <div class="cart-shoe-price">MRP : ${mycart[i].price} ₹</div>
    <button class="cart-shoe-delete">remove </button>`;

/*cart remove event*/
cart_shoe.lastElementChild.addEventListener("click", () => {
    console.log("remove");
    let mycart_ = JSON.parse(localStorage.getItem("mycart"));
    for (let j = 0; j < mycart_.length; j++) {
        if (
            document.getElementById("cart-products-box").children[j] ==
            cart_shoe
        ) {
            for (let k = j; k < mycart_.length - 1; k++) {
                mycart_[k] = mycart_[k + 1];
            }
            mycart_.pop();
            localStorage.setItem("mycart", JSON.stringify(mycart_));
            window.location.href = "./mycart.html";

            return;
        }
    }
});

document.getElementById("cart-products-box").appendChild(cart_shoe);

```

```

    }

    mycart.forEach((element) => {
        total_price = total_price + element.price;
    });
} else {
    console.log("else");
    document.getElementById(
        "cart-products-box"
    ).innerHTML = `<p style="width:100% ; height:100%; display:flex; justify-content: center;
align-items: center; color:rgb(128, 128, 128);">Your cart is empty!</p>`;
}

/*cart price calculate*/

document.getElementById("cart-price-box").innerHTML = ` <li
    style="
        color: rgb(141, 141, 141);
        font-weight: bold;
        border-bottom: 0.2px solid rgb(174, 174, 174);
    "
    >
    PRICE DETAILS
</li>
<li>
    <p>Price (${mycart.length} items)</p>
    <p>${total_price} ₹</p>
</li>
<li>
    <p>Packaging Fee</p>
    <p>+ 0 ₹</p>
</li>

```

```

</li>

  <p>Delivery Charges</p>

  <p>+ 0 ₹</p>

</li>

<li>

  <p>Coupons for you</p>

  <p>- 0 ₹</p>

</li>

<li style="border-top: 2px dotted black">

  <p>Total Amount</p>

  <p>${total_price} ₹</p>

</li>`;

```

```

document.getElementById("buy-cart").addEventListener("click", () => {
  if (mycart != null && mycart.length != 0) {
    localStorage.setItem(
      "myorder",
      JSON.stringify({ products: mycart, amount: total_price })
    );
    window.location.href = "./order.html";
  }
});
</script>
</html>

```

### mylike.js

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

```

```

<title>My Like</title>
<link rel="stylesheet" href="./css/shoes.css" />
</head>
<body>
  <h2
    style="
      color: red;
      font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
        sans-serif;
      padding: 15px;
      background-color: black;
      width: calc(100% - 30px);
      height: 40px;
    "
  >
    MY LIKE
  </h2>
  <div class="frame" id="mylike"></div>
</body>

<script>
JSON.parse(localStorage.getItem("mylike")).forEach((element) => {
  let like_shoe = document.createElement("div");
  like_shoe.className = "shoe";
  like_shoe.innerHTML = `
  <p class="shoe_name">${element.name}</p>
  <p class="shoe_price">MRP : ${element.price} ₹</p>`;

```

```

like_shoe.addEventListener("click", () => {
  localStorage.setItem("profile_data", JSON.stringify(element));
  window.location.href = "./profile.html";
});
document.getElementById("mylike").appendChild(like_shoe);
});
console.log("hello");
</script>
</html>

```

### myorder.js

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>My Order</title>
  <link rel="stylesheet" href="./css/myorder.css" />
</head>
<body>
  <h2
    style="
      color: red;
      font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
        sans-serif;
      padding: 15px;
      background-color: black;
      width: calc(100% - 30px);
      height: 40px;
    "
  >

```

```

MY ORDER

</h2>

<div id="confirm-orders">


</div>

</div>

</body>

<script>

let orderconfirm = JSON.parse(localStorage.getItem("orderconfirm"));

//////////

if (orderconfirm != null && orderconfirm != 0) {


document.getElementById(

    "confirm-orders"

    ).innerHTML = `<div style="width:100% ; height:100%; display:flex; justify-content: center;
align-items: center; color:rgb(128, 128, 128);"><div class= "loader"></div></div>`;


let options = {
    method: "POST",
    headers: {
        "Content-type": "application/json; charset=UTF-8",
    },
    body: JSON.stringify(orderconfirm),
};


// Fetch request to the backend
//fetch("http://localhost:8000/order", options)
fetch("https://trekist.onrender.com/confirmorder", options)
.then((response) => {
    if (!response.ok) {
        console.error("Response error:", response);
        document.getElementById(

```

```

        "confirm-orders"

    ).innerHTML = `<p style="width:100% ; height:100%; display:flex; justify-content: center;
align-items: center; color:rgb(128, 128, 128);">some error occur! , try again</p>`;

    return;
}

// Parse the response JSON and return it to the next .then()

return response.json();
})
.then((result) => {
    // Log the result from the backend (e.g., { ids: [...] })

    if (result.result !== "error") {
        console.log("result successfull");
        let confirm_orders = document.getElementById("confirm-orders");
        confirm_orders.innerHTML = "";

        result.result.forEach(element => {

            let confirm_order = document.createElement("div");
            confirm_order.className = "confirm-order";

confirm_order.innerHTML = `
        <img
            src=${element.product.image}
            alt=""
        />
        <div class="confirm-order-details">
            <li>${element.product.name}</li>
            <li>${element.product.price} ₹</li>

```



```

        <li>size : ${element.product.set_size}</li>
        <li>status : ${element.status}</li>
    </div>
    <div class="confirm-order-address">
        <li>Order id : ${element._id}</li>
        <li>Delivery Expected : ${element.date}</li>
        <li>Add : ${element.address.state}, ${element.address.city} ,
        ${element.address.localadd}</li>
        <li>Mob : ${element.address.mob}</li>
    </div>

    confirm_orders.appendChild(confirm_order)

    });
    } else {
        console("fetch error");
        document.getElementById(
            "confirm-orders"
        ).innerHTML = `<p style="width:100% ; height:100%; display:flex; justify-content: center;
align-items: center; color:rgb(128, 128, 128);">fetch error!</p>`;
    }
    })
    .catch((error) => {
        console.error("Fetch error:", error);
    });
    } else {
        console.log("No order");
        document.getElementById(
            "confirm-orders"
        ).innerHTML = `<p style="width:100% ; height:100%; display:flex; justify-content: center;
align-items: center; color:rgb(128, 128, 128);">No Order!</p>`;
    }
}

```

```
</script>
</html>
```

## order.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Order</title>
    <link rel="stylesheet" href="./css/order.css" />
  </head>
  <body>
    <div id="order">
      <div id="order-form">
        <h2
          style="
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            color: rgb(58, 58, 58);
          ">
          ORDER FORM
        </h2>
        <div class="form-group">
          <label for="">Name</label>
          <input type="text" required name="name" class="form-data" />
        </div>
        <div class="form-group">
          <label for="">Mob no</label>
          <input type="number" required name="mob" class="form-data" />
        </div>
        <div class="form-group">
```

```
<label for="">State</label>
><select id="states" name="state" required class="form-data">
  <option value="andhra_pradesh">Andhra Pradesh</option>
  <option value="arunachal_pradesh">Arunachal Pradesh</option>
  <option value="assam">Assam</option>
  <option value="bihar">Bihar</option>
  <option value="chhattisgarh">Chhattisgarh</option>
  <option value="goa">Goa</option>
  <option value="gujarat">Gujarat</option>
  <option value="haryana">Haryana</option>
  <option value="himachal_pradesh">Himachal Pradesh</option>
  <option value="jharkhand">Jharkhand</option>
  <option value="karnataka">Karnataka</option>
  <option value="kerala">Kerala</option>
  <option value="madhya_pradesh">Madhya Pradesh</option>
  <option value="maharashtra">Maharashtra</option>
  <option value="manipur">Manipur</option>
  <option value="meghalaya">Meghalaya</option>
  <option value="mizoram">Mizoram</option>
  <option value="nagaland">Nagaland</option>
  <option value="odisha">Odisha</option>
  <option value="punjab">Punjab</option>
  <option value="rajasthan">Rajasthan</option>
  <option value="sikkim">Sikkim</option>
  <option value="tamil_nadu">Tamil Nadu</option>
  <option value="telangana">Telangana</option>
  <option value="tripura">Tripura</option>
  <option value="uttar_pradesh">Uttar Pradesh</option>
  <option value="uttarakhand">Uttarakhand</option>
  <option value="west_bengal">West Bengal</option>
  <option value="andaman_nicobar">Andaman and Nicobar Islands</option>
  <option value="chandigarh">Chandigarh</option>
```

```
<option value="dadra_nagar_haveli">
  Dadra and Nagar Haveli and Daman and Diu
</option>
<option value="delhi">Delhi</option>
<option value="lakshadweep">Lakshadweep</option>
<option value="puducherry">Puducherry</option>
</select>
</div>
<div class="form-group">
  <label for="">City</label>
  ><input type="text" required name="city" class="form-data" />
</div>
<div class="form-group">
  <label for="">Pincode</label>
  ><input type="number" required name="pincode" class="form-data" />
</div>
<div class="form-group">
  <label for="">Local Address</label>
  <textarea
    cols="30"
    rows="4"
    required
    name="localadd"
    class="form-data"
  ></textarea>
</div>

<button id="order-submit" type="submit">CONFIRM ORDER</button>
</div>
<div
  id="order-details"
  style="
```

```

        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        color: rgb(58, 58, 58);
        "
    ></div>

    <h1 id="test"></h1>
</div>
</body>

<script>
    const today = new Date();
    const deliveryDate = new Date(today);
    deliveryDate.setDate(today.getDate() + 5); // Add 3 days to today's date

    const formattedDeliveryDate = deliveryDate.toLocaleDateString();
    console.log("Delivery Date: ", formattedDeliveryDate);
    //////////////////////////////////////
    let order = JSON.parse(localStorage.getItem("myorder"));

    let items_name = [];

    order.products.forEach((element) => {
        items_name[items_name.length] = element.name;
    });

    document.getElementById("order-details").innerHTML = `
    <h2>ORDER DETAILS</h2>
    <li>
        <p>Total items</p>
        <p>${order.products.length}</p>
    </li>
    <li>

```

```

    <p>Items Name</p>
    <p>${items_name}</p>
  </li>
  <li>
    <p>Total Payable Amount</p>
    <p>${order.amount}₹</p>
  </li>
  <li>
    <p>Payment Method</p>
    <p>Cash On delivery</p>
  </li>
  <li>
    <p>Expected deliver date</p>
    <p>${formattedDeliveryDate}</p>
  </li>`;

```

```

///form data

```

```

let fi = document.getElementsByClassName("form-data");
document.getElementById("order-submit").addEventListener("click", () => {
  document.getElementById("order-submit").style.background = "white";
  document.getElementById("order-submit").style.color = "red";
  document.getElementById("order-submit").innerHTML = "wait a moment";

  setTimeout(() => {
    document.getElementById("order-submit").style.background = "red";
    document.getElementById("order-submit").style.color = "white";
    document.getElementById("order-submit").innerHTML = "CONFIRM ORDER";
  }, 2000);

  let name = fi[0].value;
  let mob = fi[1].value;
  let state = fi[2].value;

```

```
let city = fi[3].value;
let pincode = fi[4].value;
let localadd = fi[5].value;

console.log(name, mob, city, state, pincode, localadd);
console;
if (
  name !== "" &&
  mob !== "" &&
  city !== "" &&
  state !== "" &&
  pincode !== "" &&
  localadd !== ""
) {
  function getRecipes(products) {
    let data = {
      order: products, // Assuming `order` is defined elsewhere
      address: {
        name: name, // Assuming these are defined elsewhere
        mob: mob,
        city: city,
        state: state,
        pincode: pincode,
        localadd: localadd,
      },
      date: formattedDeliveryDate, // Assuming this is defined elsewhere
    };

    let options = {
      method: "POST",
      headers: {
        "Content-type": "application/json; charset=UTF-8",
```

```
},  
  body: JSON.stringify(data),  
};
```

```
// Fetch request to the backend  
//fetch("http://localhost:8000/order", options)  
fetch("https://trekist.onrender.com/order", options)  
  .then((response) => {  
    if (!response.ok) {  
      console.error("Response error:", response);  
      return;  
    }  
  
    // Parse the response JSON and return it to the next .then()  
    console.log("tyoe", typeof response);  
    console.log("do", response);  
    return response.json();  
  })  
  .then((result) => {  
    // Log the result from the backend (e.g., { ids: [...] })  
    console.log("Response from backend:", result);  
  
    if (result.result !== "error") {  
      result.result.forEach((element) => {  
        let orderconfirm = localStorage.getItem("orderconfirm");  
  
        if (orderconfirm == [] || orderconfirm == null) {  
          orderconfirm = [element];  
        } else {  
          orderconfirm = JSON.parse(orderconfirm);  
          orderconfirm[orderconfirm.length] = element;  
        }  
      });  
    }  
  })
```



```

        localStorage.setItem(
            "orderconfirm",
            JSON.stringify(orderconfirm)
        );
    });

    alert("Order Placed successfully");
    window.location.href = "./myorder.html";
} else {
    console.log("order error ");
    alert("error in order process , try again");
    window.location.href = "./order.html";
}
})
.catch((error) => {
    console.error("Fetch error:", error);
});
}

// Call the function

getRecipes(order.products);
} else {
    console.log("fill your form");
    alert("Fill the form properly");
}
});
</script>
</html>

```

### profile.html

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Shoes</title>
  <link rel="stylesheet" href="./css/header.css" />
  <link rel="stylesheet" href="./css/profile.css" />
  <link rel="preconnect" href="https://fonts.googleapis.com" />
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400&display=swap"
    rel="stylesheet"
  />
  <link
href="https://fonts.googleapis.com/css2?family=Matemasie&display=swap"
    rel="stylesheet"
  />
  <link
href="https://fonts.googleapis.com/css2?family=Bungee+Tint&display=swap"
    rel="stylesheet"
  />
</head>
<body>
  <!--header=====>
  <header>
    <div>
      <h1>Trekist</h1>
    </div>
    <div>
      <nav>
        <li><a href="./index.html">Home</a></li>

```

```
<li><a href="#reviews">Reviews</a></li>
<li>
  <a href="/mylike.html"
    ></a>
</li>
<li>
  <a href="/mycart.html"
    ></a>
</li>
<li>
  <a href="/myorder.html"
    ></a>
</li>
</nav>
</div>
</header>
```

```
<!--profile=====-->
<div id="profile">
  <div id="profile_image">
    
    
    
    
    
    <div id="pop-up">Added</div>

    <div id="slider_navigation">
        <div class="navigation_button" onclick="left_slider()">&#8249;</div>

        <div class="navigation_button" onclick="right_slider()">&#8250;</div>
    </div>
    <div id="slider_indicator">
        
  
  
  
  
  <div id="like_profile">
    
  </div>
</div>
</div>
```

```
<div id="profile_price">
  <p style="font-size: 25px; margin-top: 60px">Lorem</p>
  <p
    style="font-size: 20px; margin-top: -10px; color: rgb(114, 114, 114)"
  >
    sports
  </p>
  <p style="font-size: 22px">MRP : 0000 ₹</p>
  <p
    style="
      margin-top: 15px;
      font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
      sans-serif;
    "
  >
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Nihil, velit.
    Ea sed eius error eum autem. Eligendi error aspernatur omnis facilis
    est sint ab, accusamus corporis obcaecati eum maiores iure.
  </p>
  <div id="select-size">
    <div onclick="set_size(5)">5</div>
    <div onclick="set_size(6)">6</div>
    <div onclick="set_size(7)">7</div>
    <div onclick="set_size(8)">8</div>
    <div onclick="set_size(9)">9</div>
  </div>
  <button id="cart-button" onclick="cart_add_effect()">
    Add to cart
  </button>
  <button id="buy-button">Buy</button>
</div>
```

```

</div>

<!--reviwes=====-->
<!--suggestion=====-->
<script src="./profile.js" type="text/javascript"></script>
</body>
</html>

```

### profile.js

```

let show = 0;
let profile_image = document.getElementById("profile_image");
let slider_indicator = document.getElementById("slider_indicator");

let profile_data = localStorage.getItem("profile_data");
profile_data = JSON.parse(profile_data);

let profile_price = document.getElementById("profile_price");

profile_price.children[0].innerHTML = profile_data.name;
profile_price.children[1].innerHTML = profile_data.type;
profile_price.children[2].innerHTML = `MRP : ` + profile_data.price + ` ₹`;
profile_price.children[3].innerHTML = profile_data.description;

for (let i = 0; i < 5; i++) {
  profile_image.children[i].src = profile_data.profile_image[i];
  slider_indicator.children[i].src = profile_data.profile_image[i];
}

slider_indicator.firstElementChild.style.border = "1.3px red solid";

// add size in profile-->

```

```

function set_size(size) {
    let profile_data = localStorage.getItem("profile_data");
    let select_size = document.getElementById("select-size");
    profile_data = JSON.parse(profile_data);
    profile_data.set_size = size;
    localStorage.setItem("profile_data", JSON.stringify(profile_data));
    for (l = 0; l < 5; l++) {
        select_size.children[l].style.border = "white solid 1.6px";
        select_size.children[l].style.boxShadow =
            "1px 1px 1px 1px rgb(182, 181, 181)";
        ("0px 0px 0px 0px ");
        select_size.children[l].style.color = "black";
    }
    select_size.children[size - 5].style.border = "red solid 1.6px";
    select_size.children[size - 5].style.boxShadow = "0px 0px 0px 0px ";
    select_size.children[size - 5].style.color = "red";
}
set_size(5);
//<--- add size in profile

```

```

function change() {
    for (i = 0; i < 5; i++) {
        if (i == show) {
            profile_image.children[i].style.zIndex = "2";
            slider_indicator.children[i].style.border = "1.3px red solid";
        } else {
            profile_image.children[i].style.zIndex = "-1";
            slider_indicator.children[i].style.border =
                "1.3px solid rgb(226, 226, 226)";
        }
    }
}

```



```

function left_slider() {
    profile_image.childNodes[0].src =

    "https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/0f76f73e-2578-4d62-aba
b-c5563ea4f78c/NIKE+DUNK+LOW+RETRO.png";

    if (show == 0) show = 4;
    else show--;
    change();
}

```

```

function right_slider() {
    profile_image.childNodes[0].src =

    "https://static.nike.com/a/images/c_limit,w_592,f_auto/t_product_v1/0f76f73e-2578-4d62-aba
b-c5563ea4f78c/NIKE+DUNK+LOW+RETRO.png";

    if (show == 4) show = 0;
    else show++;
    change();
}

```

```

if (JSON.parse(localStorage.getItem("mylike")) != null) {
    JSON.parse(localStorage.getItem("mylike")).forEach((element) => {
        console.log(element);
        if (JSON.parse(localStorage.getItem("profile_data"))._id == element._id) {
            document
                .getElementById("like_profile")
                .firstElementChild.setAttribute(
                    "src",
                    "./source/menu_icons/like_shoe.svg"
                );
        }
    });
}

```

```
///like event
```

```
document.getElementById("like_profile").addEventListener("click", () => {  
  let like_profile = document.getElementById("like_profile");  
  let profile_data = localStorage.getItem("profile_data");  
  let mylike = localStorage.getItem("mylike");  
  
  console.log(localStorage.getItem("profile_data"));  
  if (mylike == null || JSON.parse(mylike) == null) {  
    mylike = [JSON.parse(profile_data)];  
    localStorage.setItem("mylike", JSON.stringify(mylike));  
    like_profile.firstElementChild.setAttribute(  
      "src",  
      "./source/menu_icons/like_shoe.svg"  
    );  
    console.log("nulllllll");  
  } else {  
    console.log("else start");  
    mylike = JSON.parse(mylike);  
    profile_data = JSON.parse(profile_data);  
  
    let profile_value = -1;  
    mylike.forEach((element, index) => {  
      if (element._id == profile_data._id) {  
        console.log("dislike");  
        profile_value = index;  
        return false;  
      } else {  
        console.log("like");  
  
        return true;  
      }  
    })  
  }  
})
```

```

});

console.log("else end");

//like
if (profile_value !== -1) {
  for (let j = profile_value; j < mylike.length - 1; j++) {
    mylike[j] = mylike[j + 1];
  }
  mylike.pop();
  localStorage.setItem("mylike", JSON.stringify(mylike));

  like_profile.firstElementChild.setAttribute(
    "src",
    "./source/menu_icons/dislike_shoe.svg"
  );
}

///dislike
else {
  mylike[mylike.length] = profile_data;
  localStorage.setItem("mylike", JSON.stringify(mylike));
  like_profile.firstElementChild.setAttribute(
    "src",
    "./source/menu_icons/like_shoe.svg"
  );
}
}

});

///cart event

document.getElementById("cart-button").addEventListener("click", () => {
  let profile_data = JSON.parse(localStorage.getItem("profile_data"));

```

```

let mycart = localStorage.getItem("mycart");

if (mycart == null || JSON.parse(mycart) == null) {
  localStorage.setItem("mycart", JSON.stringify([profile_data]));
} else {
  mycart = JSON.parse(mycart);
  mycart[mycart.length] = profile_data;
  localStorage.setItem("mycart", JSON.stringify(mycart));
}
});

// cart_add_effect
function cart_add_effect() {
  console.log("cart");
  document.getElementById("cart-button").style.color = "black";
  document.getElementById("cart-button").style.background = "white";
  document.getElementById("cart-button").innerText = "Added Successfully";

  window.setTimeout(() => {
    document.getElementById("cart-button").style.color = "white";
    document.getElementById("cart-button").style.background = "black";
    document.getElementById("cart-button").innerText = "Add to Cart";
  }, 1150);
}

//buy button event

document.getElementById("buy-button").addEventListener("click", () => {
  document.getElementById("buy-button").style.color = "black";
  document.getElementById("buy-button").style.background = "white";
  document.getElementById("buy-button").innerText = "wait..";
  let product = JSON.parse(localStorage.getItem("profile_data"));

```

```
let myorder = { products: [product], amount: product.price };
```

```
localStorage.setItem("myorder", JSON.stringify(myorder));
```

```
window.setTimeout(() => {  
  document.getElementById("buy-button").style.color = "white";  
  document.getElementById("buy-button").style.background = "black";  
  document.getElementById("buy-button").innerText = "Buy";  
  window.location.href = "./order.html";  
}, 1000);  
});
```

#### **shoes.js**

```
var type = "all";  
var all = true;
```

```
async function fetchShoes() {  
  try {  
    const response = await fetch("https://trekist.onrender.com/shoes");  
    if (!response.ok) {  
      throw new Error("Network response was not ok");  
    }  
    const products = await response.json();  
  
    return products;  
  } catch (error) {  
    console.error("There has been a problem with your fetch operation:", error);  
  }  
}
```

```
async function shoes_load() {  
  const data = await fetchShoes();
```

```

if (data != undefined) document.getElementById("right").innerHTML = "";

console.log(data);

let count = 0;

data.forEach((element) => {
  if (element.type == type || all == true) {
    count++;
  }
});

if (type != "all")
  document.getElementById("right").innerHTML = `<p class='shoes_heading'>${
    type[0].toUpperCase() + type.slice(1)
  } (${count})</p>`;

data.forEach((element) => {
  if (element.type == type || all == true) {
    console.log(element.name);
    var shoe = document.createElement("div");
    shoe.className = "shoe";
    shoe.innerHTML = `

    <p class='shoe_name'>
    ${element.name}
    </p>

    <p class='shoe_price'> MRP :
    ${element.price}
    ₹</p>`;
  }
});

```

```

shoe.addEventListener("click", () => {
    localStorage.setItem("profile_data", JSON.stringify(element));
    window.location.href = "./profile.html";
});

document.getElementById("right").appendChild(shoe);
}
});
}

shoes_load();

function filter_apply(type_) {
    (type = type_), (all = false);
    if (type_ == "all") {
        all = true;
        type = "all";
    }

    shoes_load();
    document.getElementById("right").scrollTop = 0;
}

```

### **collection.css**

```

#collection_heading {
    padding: 50px 0;
    width: 100%;
    height: 50px;
    background-color: rgb(230, 230, 230);
    color: rgb(0, 0, 0);
    font-weight: bold;
    font-size: 60px;
}

```

```
align-items: center;
text-align: center;
font-family: "Trebuchet MS", "Lucida Sans Unicode", "Lucida Grande",
  "Lucida Sans", Arial, sans-serif;
}

#collection {
  width: 100%;
  display: flex;
  flex-wrap: wrap;
  flex-direction: row;
  align-items: center;
  justify-content: center;
  background-color: rgb(230, 230, 230);
}
```

```
.box {
  display: inline-block;
  width: 300px;
  height: 400px;
  margin: 20px;
  background-color: rgba(255, 255, 255, 0.925);
}
```

```
.heading {
  margin: 0px 16px;
  margin-top: 5%;
  height: 5%;
  color: rgb(0, 0, 0);
  font-size: medium;
  font-family: Verdana, sans-serif;
  font-weight: bold;
  font-style: normal;
```



```
}
```

```
.container {  
  height: 90%;  
  list-style: none;  
  display: flex;  
  flex-wrap: wrap;  
  align-items: center;  
  color: black;  
  justify-content: space-around;  
}
```

```
.sub_box {  
  display: inline-block;  
  height: 45%;  
  width: 40%;  
  border: 1px rgb(255, 255, 255) solid;  
  background-color: rgb(255, 255, 255);  
}
```

```
.sub_box:hover {  
  transform: scale(1.06);  
  transition-property: "scale";  
  transition-duration: 0.4s;  
  box-shadow: 0.1px 0.1px 0.5px 0.5px rgb(208, 208, 208);  
}
```

```
.collection_image {  
  display: inline-block;  
  width: 100%;  
}
```

```
.collection_name_box {
    color: black;
    height: 30px;
    width: 100%;
    margin-left: 3px;
}

.collection_name {
    font-size: 11.4px;
    color: rgb(107, 107, 107);
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}

.collection_price {
    font-size: 13px;
    font-size: small;
    color: rgb(0, 0, 0);
    font-family: "poppins", sans-serif;
    margin-bottom: 3px;
    letter-spacing: 0.5px;
}

@media screen and (max-width: 600px) {
    #collection_heading {
        font-size: 50px;
        padding: 35px 0px;
    }
}
```

#### **footer.css**

```
#about {
    height: 300px;
    background-color: rgb(255, 0, 0);
    background-color: rgb(255, 255, 255);
}
```

```
display: flex;
flex-direction: row;
flex-wrap: wrap;
}
```

```
#footer-left {
width: 50%;
height: 200px;
display: flex;
justify-content: center;
align-items: center;
}
```

```
#footer-left > p {
width: 80%;
font-family: "poppins", "Gill Sans", "Gill Sans MT", Calibri, "Trebuchet MS",
    sans-serif;
font-weight: 200;
color: rgb(47, 47, 47);
}
```

```
#footer-right {
width: 50%;
height: 200px;
display: flex;
flex-direction: column;
justify-content: center;
}
```

```
#footer-right > li {
margin-left: 20%;
margin-top: 7px;
```

```
list-style: none;
color: rgb(47, 47, 47);
font-family: "poppins", Arial, Helvetica, sans-serif;
font-weight: 200;
font-size: 14px;
}
```

```
#footer-bottom {
width: 100%;
height: 100px;
}
```

```
#footer-social {
width: 100%;
height: 50%;
display: flex;
justify-content: center;
align-items: center;
}
```

```
.footer-social-icon {
width: 30px;
margin: 0 3%;
opacity: 90%;
filter: grayscale(1000%);
}
```

```
#footer-copyright {
color: rgb(47, 47, 47);
width: 100%;
text-align: center;
opacity: 50%;
font-family: "poppins";
```

```
font-weight: 300;
margin-top: 20px;
font-size: 13px;
}
```

```
@media screen and (max-width: 800px) {
  #about {
    height: 400px;
    flex-direction: column;
  }
}
```

```
#footer-left {
  width: 100%;
  height: 150px;
}
#footer-right {
  width: 100%;
  height: 150px;
}
#footer-right > li {
}
#footer-bottom {
  height: 100px;
}
}
```

## header.css

```
/*HEADER=====*/
header {
  width: 100%;
  min-height: 80px;
  height: auto;
```

```
display: flex;
flex-direction: row;
background-color: rgb(0, 0, 0);
}
```

```
header > div {
width: 50%;
height: auto;
display: flex;
align-items: center;
justify-content: space-around;
}
```

```
header > div > h1 {
margin: 10px;
display: flex;
flex-direction: row;
color: rgb(255, 0, 0);
font-family: "Matemasie", sans-serif;
font-weight: 400;
font-size: 40px;
}
```

```
header > div > nav {
display: flex;
flex-direction: row;
color: white;
}
```

```
header > div > nav > li {
font-family: "Poppins", sans-serif;
font-weight: 200;
font-size: small;
```

```
font-style: normal;
margin: 0 1.8vw;
cursor: pointer;
border-bottom: 1px rgb(0, 0, 0) solid;
list-style: none;
letter-spacing: 0.5px;
scale: 2px;
-webkit-user-select: none; /* Safari */
-ms-user-select: none; /* IE 10 and IE 11 */
user-select: none; /* Standard syntax */
}
header > div > nav > li > a > img {
  width: 20px;
}

header > div > nav > li:hover {
  transform: scale(1.09);
  transition: all 0.17s;
}
header > div > nav > li > a {
  text-decoration: none;
  color: rgb(255, 255, 255);
}
header > div > nav > li > a:hover {
  color: red;
}

@media only screen and (max-width: 600px) {
  header {
    flex-direction: column;
  }
}
```

```
header > div {  
  width: 100%;  
  height: 50%;  
}  
}
```

## **main.css**

```
/*MAIN=====*/  
main{  
  display: flex;  
  flex-direction: row;  
  width:100%;  
  height: 400px;  
  
  background-color: rgb(1, 1, 1);  
  flex-wrap: wrap;  
}  
  
main>div  
{  
  background-color: rgb(0, 0, 0);  
  height: 100%;  
  width: 50%;  
display: flex;  
align-items: center;  
justify-content: center;  
flex-direction: column;  
}  
main>div>img  
{  
  display: flex;
```



```
width: 100%;  
height: auto;  
overflow: hidden;  
}
```

```
main>div>h2
```

```
{  
  font-family: "Poppins", sans-serif;  
  font-weight: 400;  
  font-style:normal;
```

```
}
```

```
main>div>p
```

```
{  
  margin:0 12%;  
  
  font-family: "poppins", sans-serif;  
  font-weight: 100;  
  font-size: small;  
  background-color: rgb(0, 0, 0);  
  
}
```

```
/*-----*/
```

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

```
  main
```

```
{  
  flex-direction: column;
```

```
        height: auto;
    }
    main>div{
        width: 100%;
    }

    main>div>p
    {
        margin-bottom: 30px;
    }
}
```

#### **mycart.css**

```
* {
    margin: 0px;
    padding: 0px;
}

#cart {
    width: 100%;
    height: auto;
    display: flex;
    background-color: rgb(228, 228, 228);
}

/* products*/

#cart-products {
    width: 60%;
    height: calc(100vh - 58px);
    background-color: rgb(228, 228, 228);
}
```

```
}
```

```
#cart-products-box {  
  width: calc(100% - 200px);  
  margin-left: 150px;  
  height: calc(100% - 100px);  
  margin-top: 30px;  
  background-color: white;  
  border-radius: 5px;  
  overflow-y: scroll;  
  
  flex-wrap: wrap;  
  justify-content: center;  
  align-items: center;  
}
```

```
.cart-shoe {  
  height: 130px;  
  width: 100%;  
  margin: 20px 0;  
  display: flex;  
}
```

```
.cart-shoe-image {  
  height: 100%;  
  margin: 0 0 10px 20px;  
}
```

```
.cart-shoe-details {  
  height: 100%;  
  width: 200px;  
  margin-left: 10px;
```

```
display: flex;
flex-direction: column;
}
.cart-shoe-name {
font-size: 17px;
font-family: "poppins", "Lucida Sans", "Lucida Sans Regular", "Lucida Grande",
"Lucida Sans Unicode", Geneva, Verdana, sans-serif;
font-weight: 200;
}
.cart-shoe-size {
font-family: "poppins", sans-serif;
font-size: 14px;
margin: 7px 0;
color: rgb(102, 103, 103);
}

.cart-shoe-price {
font-family: "poppins", sans-serif;
font-size: 15px;
}
.cart-shoe-delete {
margin: 5px;
width: 100px;
background-color: black;
color: white;
height: 30px;
}

#cart-products-order {
width: calc(100% - 200px);
margin-left: 150px;
height: 60px;
```

```
box-shadow: 1px 1px 5px 0.1px rgb(155, 154, 154);
background-color: rgb(255, 255, 255);
border-radius: 2px;
display: flex;
align-items: center;
justify-content: right;
}
```

```
#cart-products-order > button {
width: 200px;
height: 80%;
background-color: red;
color: white;
font-family: "Gill Sans", "Gill Sans MT", Calibri, "Trebuchet MS", sans-serif;
margin-right: 20px;
border: 0;
font-size: 1.2rem;
box-shadow: 1px 1px 0.5px 0.5px rgb(87, 87, 87);
}
```

```
/* price*/
#cart-price {
width: 40%;
height: calc(100vh - 58px);
background-color: rgb(228, 228, 228);
}
```

```
#cart-price-box {
width: 70%;
margin-left: 5%;
height: calc(60% - 30px);
margin-top: 30px;
```

```
background-color: white;
border-radius: 5px;
}
```

```
#cart-price-box > li {
  list-style: none;
  width: calc(100% - 10px);
  padding-left: 10px;
  height: 42px;
  display: flex;
  align-items: center;
  margin-top: 3px;
  font-size: 0.9rem;
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
}
```

```
#cart-price-box > li > p {
  width: 50%;
  height: 100%;
  display: flex;
  justify-content: left;
  align-items: center;
}
```

```
@media screen and (max-width: 600px) {
  #cart-products {
    height: 90vh;
  }
  #cart-products-box {
    margin-left: 10px;
    width: calc(100% - 10px);
  }
}
```

```
.cart-shoe-details {  
  height: 100%;  
  width: 110px;  
  margin-left: 10px;  
  display: flex;  
  flex-direction: column;  
}  
.cart-shoe-name {  
  font-size: 12px;  
  font-family: "poppins", "Lucida Sans", "Lucida Sans Regular",  
    "Lucida Grande", "Lucida Sans Unicode", Geneva, Verdana, sans-serif;  
  font-weight: 300;  
}  
.cart-shoe-size {  
  font-family: "poppins", sans-serif;  
  font-size: 10px;  
  margin: 4px 0;  
  color: rgb(102, 103, 103);  
}  
  
.cart-shoe-price {  
  font-family: "poppins", sans-serif;  
  font-size: 11px;  
}  
.cart-shoe-delete {  
  margin: 5px;  
  margin-left: 0;  
  width: 50px;  
  background-color: black;  
  color: white;  
  font-size: 9px;
```

```
    height: 15px;
  }

#cart-products-order {
  margin-left: 10px;
  width: calc(100% - 10px);
}

#cart-products-order > button {
  width: 130px;
}

.cart-shoe {
  height: 80px;
  width: 100%;
  margin: 5px 0;
  display: flex;
}

#cart-price-box {
  width: 95%;
  margin-left: 5%;
  height: calc(60% - 30px);
  margin-top: 30px;
  background-color: white;
  border-radius: 5px;
}
}
```

### **myorder.css**

```
* {
  padding: 0;
  margin: 0;
}
```



```
#confirm-orders {  
  height: calc(100vh - 70px);  
  width: 100%;  
  
  flex-wrap: wrap;  
  overflow-y: scroll;  
}
```

```
.confirm-order {  
  margin-left: 20px;  
  height: 150px;  
  width: calc(100% - 4%);  
  display: flex;  
  flex-direction: row;  
  margin-top: 20px;  
  box-shadow: 0px 0px 1px 0px black;  
}
```

```
.confirm-order > img {  
  height: 100%;  
}
```

```
.confirm-order-details {  
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;  
  height: 100%;  
  display: flex;  
  justify-content: space-around;  
  flex-direction: column;  
  margin-left: 30px;  
}
```

```
.confirm-order-details > li {
```

```
list-style: none;
font-size: 0.9rem;
}

.confirm-order-address {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  height: 100%;
  display: flex;
  flex-direction: column;
  justify-content: space-around;

  margin-left: 30px;
}

.confirm-order-address > li {
  list-style: none;
  font-size: 0.9rem;
}

@media screen and (max-width: 600px) {
  .confirm-order {
    height: 120px;
  }

  .confirm-order-details {
    margin-left: 10px;
  }

  .confirm-order-details > li {
    font-size: 0.7rem;
  }

  .confirm-order-address {
    margin-left: 10px;
  }
}
```

```
.confirm-order-address > li {  
  font-size: 0.7rem;  
}  
}
```

```
/*loading*/  
.loader {  
  position: relative;  
  width: 48px;  
  height: 48px;  
  background: #ff0000;  
  border-radius: 50%;  
  animation: ellipseAnimation 2s linear infinite;  
}
```

```
@keyframes ellipseAnimation {  
  0% {  
    border-radius: 50%;  
  }
```

```
  12.5% {  
    border-radius: 0 50% 50% 50%;  
    transform: rotate(45deg);  
  }
```

```
  25% {  
    border-radius: 0 0 50% 50%;  
    transform: rotate(90deg);  
  }
```

```
  37.5% {  
    border-radius: 0 0 0 50%;
```

```
    transform: rotate(135deg);  
}
```

```
50% {  
    border-radius: 0;  
    transform: rotate(180deg);  
}
```

```
62.5% {  
    border-radius: 50% 0 0 0;  
    transform: rotate(225deg);  
}
```

```
75% {  
    border-radius: 50% 50% 0 0;  
    transform: rotate(270deg);  
}
```

```
87.5% {  
    border-radius: 50% 50% 50% 0;  
    transform: rotate(315deg);  
}
```

```
100% {  
    border-radius: 50%;  
    transform: rotate(360deg);  
}  
}
```

#### **order.css**

```
#order {  
    width: 100%;
```

```
display: flex;
flex-wrap: wrap;
justify-content: center;
}
```

```
#order-form {
width: 40%;
background-color: rgba(245, 155, 155, 0.406);
height: fit-content;
padding-bottom: 10px;
display: flex;
flex-direction: column;
align-items: center;
}
```

```
.form-group {
width: 70%;
margin: 10px 0;
display: flex;
flex-direction: column;
}
```

```
.form-group > label {
font-family: "Gill Sans", "Gill Sans MT", Calibri, "Trebuchet MS", sans-serif;
margin-bottom: 2px;
}
```

```
.form-group > input,
.form-group > select {
border: 0;
box-shadow: 0px 0px 1px 0px black;
height: 30px;
width: 80%;
}
```

```
padding-left: 2%;  
font-size: 0.9rem;  
color: rgb(61, 61, 61);  
border-radius: 2px;  
}
```

```
.form-group > textarea {  
border: 0;  
box-shadow: 0px 0px 1px 0px black;  
width: 80%;  
padding-left: 2%;  
font-size: 0.9rem;  
color: rgb(61, 61, 61);  
border-radius: 2px;  
}
```

```
.form-group > input:focus,  
.form-group > select:focus,  
.form-group > textarea:focus {  
outline: red 1px solid;  
}
```

```
#order-submit {  
margin: 5px 0;  
  
width: 40%;  
font-family: cursive;  
color: white;  
background-color: rgb(255, 0, 0);  
font-size: 1.2rem;  
border: 0;  
box-shadow: 0px 0px 2px 0px black;  
}
```

```
#order-details {  
  margin-left: 5%;  
  width: 40%;  
  background-color: rgba(245, 155, 155, 0.406);  
  display: flex;  
  flex-direction: column;  
  min-height: 500px;  
  justify-content: center;  
  align-items: center;  
  height: max-content;  
}
```

```
#order-details > li {  
  list-style: none;  
  width: 80%;  
  min-height: 50px;  
  height: fit-content;  
  display: flex;  
  justify-content: space-between;  
}
```

```
#order-details > li > p {  
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;  
  font-size: 17px;  
  width: 50%;  
}
```

```
@media screen and (max-width: 600px) {  
  #order-form {  
    width: 90%;  
  }
```

```
    margin: 10px 5%;
    height: fit-content;
    padding-bottom: 10px;
}

.form-group > input,
.form-group > select {
    width: 100%;
}

.form-group > textarea {
    width: 100%;
}

#order-details {
    width: 80%;
    padding: 5%;
    margin: 10px 5%;
    height: 200px;
}
}
```

### **profile.css**

```
* {
    padding: 0px;
    margin: 0px;
}

#profile {
    width: 100%;
    height: 600px;
    display: flex;
}
```



```
#profile_image {  
  width: 50%;  
  height: 100%;  
  position: relative;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

```
#like_profile {  
  width: 27px;  
  height: 27px;  
  color: black;  
  position: relative;  
  display: flex;  
  bottom: 430px;  
  right: -40px;  
}
```

```
#profile_price {  
  width: 50%;  
  display: flex;  
  flex-direction: column;  
  align-items: left;  
  justify-content: center;  
}
```

```
.slider {  
  height: 440px;  
  position: absolute;  
}
```

```
#slider_navigation {  
    height: 300px;  
    width: 440px;  
    position: absolute;  
    z-index: 2;  
    display: flex;  
    align-items: center;  
    justify-content: space-between;  
}
```

```
.navigation_button {  
    font-size: 50px;  
    font-family: "Gill Sans MT", Calibri, "Trebuchet MS", sans-serif;  
    cursor: pointer;  
    opacity: 10%;  
    margin: 0 5px;  
    height: 100px;  
  
    text-align: center;  
    border-radius: 10%;  
    -webkit-user-select: none; /* Safari */  
    -ms-user-select: none; /* IE 10 and IE 11 */  
    user-select: none; /* Standard syntax */  
}
```

```
.navigation_button:hover {  
    opacity: 30%;  
}
```

```
#slider_indicator {  
    width: 350px;
```

```
height: 50px;
position: relative;
z-index: 4;
top: 240px;
display: flex;
background-color: rgb(226, 226, 226);
align-items: center;
justify-content: center;
}
```

```
.slider_indicator_image {
margin-left: 10px;
border-radius: 10%;
border: 1.3px rgb(226, 226, 226) solid;
}
```

```
#profile_price > p {
height: auto;
font-family: "poppins";
font-size: 22px;
width: 80%;
}
```

```
#select-size {
width: 100%;
height: 50px;
display: flex;
align-items: center;
}
```

```
#select-size > div {
background-color: white;
```

```
height: 30px;
width: 30px;
margin-right: 20px;
text-align: center;
font-size: 1.1rem;
display: flex;
align-items: center;
justify-content: center;
border-radius: 2px;
font-weight: bold;
box-shadow: 1px 1px 1px 1px rgb(182, 181, 181);
cursor: pointer;
}
```

```
#profile_price > button {
width: 40%;
height: 60px;
margin: 12px;
background-color: black;
color: aliceblue;
border-radius: 40px;
}
```

```
@media screen and (max-width: 800px) {
#profile {
min-height: 1200px;
height: auto;
flex-direction: column;
}
```

```
#profile_image {
width: 100%;
```

```
min-height: 500px;  
height: auto;  
background-color: rgb(255, 255, 255);  
}
```

```
#like_profile {  
width: 27px;  
height: 27px;  
color: black;  
position: relative;  
display: flex;  
bottom: 390px;  
right: -10px;  
}
```

```
#profile_price {  
width: 100%;  
min-height: 600px;  
height: auto;  
background-color: rgb(255, 255, 255);  
align-items: center;  
}
```

```
#select-size {  
width: 80%;  
}
```

```
.slider {  
height: 360px;  
}
```

```
#slider_navigation {
```

```
height: 300px;
width: 360px;
position: absolute;
z-index: 2;
display: flex;
align-items: center;
justify-content: space-between;
}
}
```

```
/*pop-up*/
```

```
#pop-up {
position: absolute;
width: 300px;
height: 50px;
background-color: rgba(0, 0, 0, 0.584);
color: white;
border-radius: 10px;
top: 5px;
display: none;
justify-content: center;
align-items: center;
font-size: 1.2rem;
}
```

## shoes.css

```
* {
margin: 0;
padding: 0;
```

```
}  
  
#shoes {  
  width: 100%;  
  height: 600px;  
  background-color: white;  
  display: flex;  
  flex-direction: row;  
}
```

```
#left {  
  width: 20%;  
  height: 100%;  
  background-color: rgb(228, 228, 228);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  flex-direction: column;  
}
```

```
#left > li {  
  width: 80%;  
  background-color: rgb(228, 228, 228);  
  margin: 18px auto;  
  color: black;  
  text-align: center;  
  font-size: 20px;  
  list-style: none;  
  height: 40px;  
  border-radius: 10px;  
  display: flex;  
  align-items: center;  
  justify-content: center;
```

```
font-family: "poppins", sans-serif;
cursor: pointer;
-webkit-user-select: none; /* Safari */
-ms-user-select: none; /* IE 10 and IE 11 */
user-select: none; /* Standard syntax */
}
```

```
@media screen and (max-width: 800px) {
  #left {
    flex-direction: row;
    flex-wrap: wrap;
  }
  #left > li {
    margin: 0px 5px;
    width: 75px;
    height: 35px;
    font-size: 10px;
    background-color: white;
  }
}
```

```
#right {
  width: 80%;
  height: 100%;
  display: flex;
  flex-wrap: wrap;
  align-items: center;
  overflow-y: scroll;
  justify-content: center;
}
```

```
.shoes_heading {
```



```
width: calc(100% - 60px);  
color: black;  
margin-left: 60px;  
margin-top: 12px;  
font-family: "poppins", sans-serif;  
}
```

```
.shoe {  
width: 200px;  
height: 260px;  
margin: 15px 10px;  
display: flex;  
flex-direction: column;  
}
```

```
.shoe:hover {  
transition-duration: 0.2s;  
transform: scale(1.03);  
box-shadow: 0.1px 0.1px 1px 1px rgb(214, 213, 213);  
border-radius: 2px;  
}
```

```
.shoe_image {  
width: 100%;  
}
```

```
.shoe_name {  
color: rgb(139, 139, 139);  
font-family: Verdana, Geneva, Tahoma, sans-serif;  
font-size: 16.2px;  
margin: 5px 0px 6px 6px;  
}
```

```
.shoe_price {  
  color: black;  
  font-family: "poppins", sans-serif;  
  margin: -6px 6px;  
  font-size: 19px;  
}
```

```
@media screen and (max-width: 600px) {  
  #shoes {  
    flex-direction: column;  
  }  
}
```

```
#left {  
  width: 100%;  
  height: 20%;  
  background-color: rgb(182, 182, 182);  
}
```

```
#right {  
  width: 100%;  
  height: 80%;  
}
```

```
.shoes_heading {  
  width: calc(100% - 30px);  
  color: black;  
  margin-left: 30px;  
  margin-top: 10px;  
}
```

```
.shoe {  
  width: 150px;  
  height: 200px;
```

```
    margin: 15px 10px;
  }
}
```

```
.frame {
  display: flex;
  width: 100%;
  height: auto;
  flex-wrap: wrap;
  flex-direction: row;
}
```

```
/* loading animation */
```

```
/*loading*/
.loader {
  position: relative;
  width: 48px;
  height: 48px;
  background: #ff0000;
  border-radius: 50%;
  animation: ellipseAnimation 2s linear infinite;
}
```

```
@keyframes ellipseAnimation {
  0% {
    border-radius: 50%;
  }
```

```
  12.5% {
    border-radius: 0 50% 50% 50%;
    transform: rotate(45deg);
```

```
}
```

```
25% {  
  border-radius: 0 0 50% 50%;  
  transform: rotate(90deg);  
}
```

```
37.5% {  
  border-radius: 0 0 0 50%;  
  transform: rotate(135deg);  
}
```

```
50% {  
  border-radius: 0;  
  transform: rotate(180deg);  
}
```

```
62.5% {  
  border-radius: 50% 0 0 0;  
  transform: rotate(225deg);  
}
```

```
75% {  
  border-radius: 50% 50% 0 0;  
  transform: rotate(270deg);  
}
```

```
87.5% {  
  border-radius: 50% 50% 50% 0;  
  transform: rotate(315deg);  
}
```

```
100% {  
  border-radius: 50%;  
  transform: rotate(360deg);  
}  
}
```

### **index.js (backend)**

```
const express = require("express");  
const mongoose = require("mongoose");  
const cors = require("cors");  
const bodyparser = require("body-parser");  
const dotenv = require("dotenv");  
const { setTimeout } = require("timers/promises");  
dotenv.config();  
const PORT = process.env.PORT || 3030;  
  
const app = express();  
//<-----middleware----->  
app.use(bodyparser.json());  
app.use(cors());  
//<-----connect to mongodb----->  
  
main().catch((err) => console.log(err));  
  
async function main() {  
  // await mongoose.connect('mongodb://127.0.0.1:27017/');  
  await mongoose.connect(process.env.DB_KEY);  
  console.log("connected to mongodb");  
}  
  
// schema for the 'shoes' collection
```

```

const shoesSchema = new mongoose.Schema({
  name: String,
  price: Number,
  size: [Number],
  description: String,
  type: String,
  image: String,
  profile_image: [String],
});

// schema for the 'collection' collection
const collectionSchema = new mongoose.Schema({
  name: String,
  shoes: [Object], // Array of image URLs
});

// schema for the 'order' collection
const orderSchema = new mongoose.Schema({
  product: Object,
  address: Object,
  status: String,
  date: String,
});

// model for the 'shoes' collection
const Shoes = mongoose.model("shoes", shoesSchema);

// model for the 'collection' collection
const Collection = mongoose.model("Collection", collectionSchema, "collection");

// model for the 'order' collection
const Order = mongoose.model("Order", orderSchema);

//<-----send ----->

app.get("/", (req, res) => {

```

```
console.log("get /");
res.send("hello");
});

//<-----send shoes----->
app.get("/shoes", async (req, res) => {
  async function hello() {
    const data = await Shoes.find();
    console.log("get /shoes");
    res.json(data); // Send shoes data as JSON
  }
  hello();
});
```

```
//<-----send collections----->
app.get("/collections", async (req, res) => {
  async function hello() {
    const data = await Collection.find();
    console.log("get /collections");
    res.json(data); // Send shoes data as JSON
  }
  hello();
});
```

```
//<----- recive order----->
app.post("/order", async (request, response) => {
  console.log("post /order");
  async function hello(req, res) {
    console.log("put /order");

    const data = await req.body;
    console.log(data.order);
```

```

////////////////////////////////////
let result = Array(data.order).fill("null");
console.log("dataorder", typeof data.order);

setTimeout(
  data.order.forEach((element, index) => {
    console.log("====", index);
    const newOrder = new Order({
      product: element,
      address: data.address,
      status: "On the way",
      date: data.date,
    });
    // Save the new user to the database
    newOrder
      .save()
      .then((doc) => {
        console.log(`New user added: ${doc}`);
        result[index] = doc._id;

        function check() {
          for (let i = 0; i < data.order.length; i++) {
            if (result[i] == "null") return false;
            else return true;
          }
        }
        if (check() == true) res.json({ result: result });

        console.log("+++++++", result);
      })
      .catch((err) => {

```



```

        console.error("Error adding user:", err);
        res.json({ result: "error" });
    })
    .finally(() => {
        // Close the connection after the operation
        console.log("order process end");
    });
    },
    1000
);
}

```

```

    hello(request, response);
});

```

```

//<-----send confirm order----->

```

```

/*app.POST("/confirmorder", async (req, res) => {
    async function hello(request, response) {
        console.log(request.body);
    }

```

```

        console.log(results);
        console.log("get / order");
        res.json({ yo: 123 }); // Send shoes data as JSON
    }
    hello(req, res);
});
*/

```

```

app.post("/confirmorder", async (request, response) => {
    console.log("post /confirmorder");
    async function hello(req, res) {
        console.log("put /order");
    }
}

```

```

const data = await req.body;
console.log("data", data);

////////////////////////////////////

const ids = data;
const objectIds = ids.map((id) => new mongoose.Types.ObjectId(id));
Order.find({ _id: { $in: objectIds } })
  .then((doc) => {
    console.log(`confirm order list: ${doc}`);

    res.json({ result: doc });
  })
  .catch((err) => {
    console.error("Error adding user:", err);
    res.json({ result: "error" });
  })
  .finally(() => {
    // Close the connection after the operation
    console.log("confirm order list end");
  });
}

hello(request, response);
});

//<-----listen at the port 8000----->

app.listen(PORT, () => {
  console.log(`server started on port ${PORT}`);
});

```

**package.json**

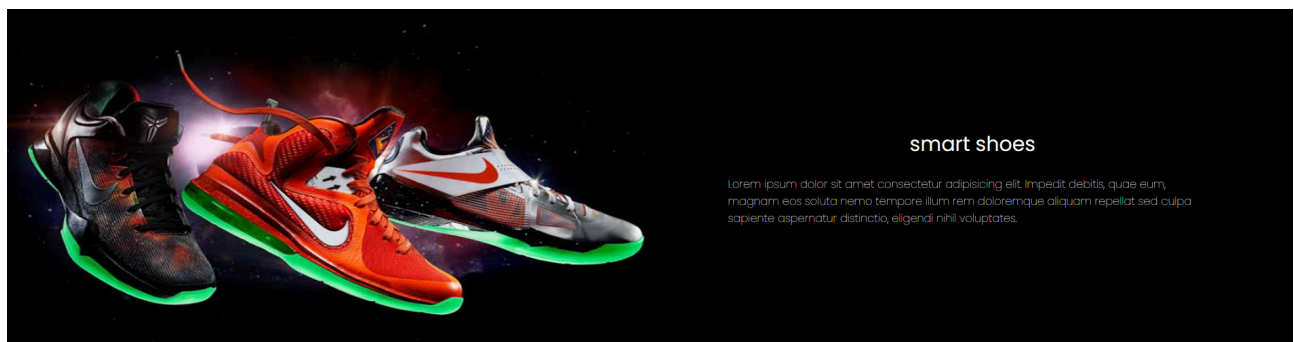
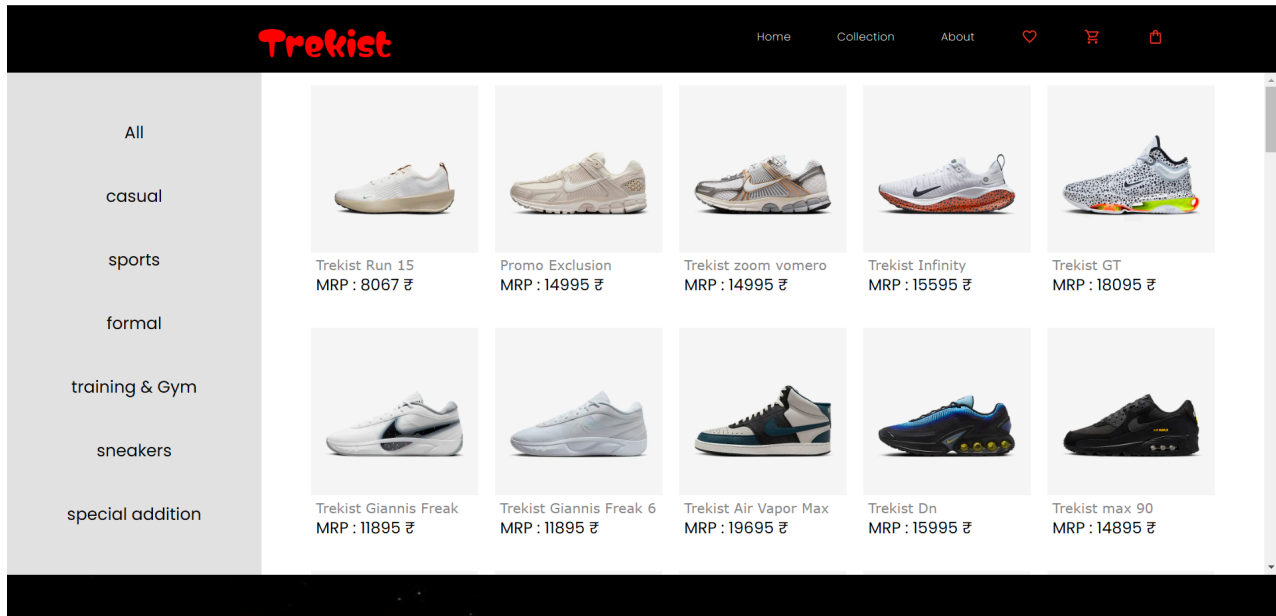
```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.3",
    "cors": "^2.8.5",
    "dotenv": "^16.4.5",
    "express": "^4.21.0",
    "mongoose": "^8.6.3"
  }
}
```

## **Deployment**

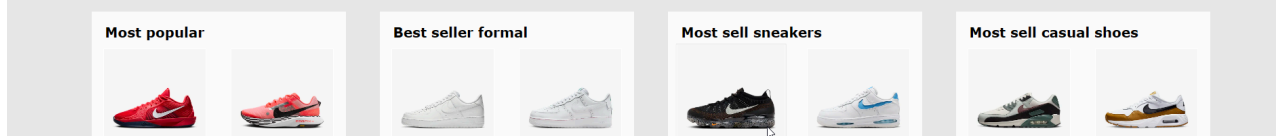
- The website is hosted on GitHub Pages: <https://ahsan-abc.github.io/trekist>
- The code repository is available on GitHub: <https://github.com/ahsan-abc/trekist>

## Project Snapshot

### Home Page:



## COLLECTION



# COLLECTION

## Most popular



Sabrina 2 EP  
MRP : 11895



Trekist Ultrafly  
MRP : 21695



Trekist Vaporfly 2  
MRP : 20695



Trekist Pegasus Dow  
MRP : 11895

## Best seller formal



Trekist Air Force 1  
MRP : 10795



Trekist Air Force  
MRP : 8097



Trekist Water Force  
MRP : 12578



Trekist Tanjun  
MRP : 3145

## Most sell sneakers



Trekist vapromax  
MRP : 19595



Trekist G.T Low  
MRP : 12795



Trekist Spinzik u  
MRP : 14095



Trekist P-12  
MRP : 10795

## Most sell casual shoes



Trekist P-Max  
MRP : 12440



Trekist Max SC  
MRP : 6790



Trekist Tanny Et  
MRP : 10090



Trekist Royal-K  
MRP : 7495

## Most sell sports shoes



MRP : 17595



MRP : 15995



Trekist Motiva  
MRP : 9207



Trekist Run 15  
MRP : 8067



Trekist – where innovation meets style. Our premium shoes are crafted for comfort and built to last. Step into excellence with every pair. Explore our collection and elevate your journey.\*


trekistshoes@gmail.com  
sahibganj, jharkhand, 816109  
give - feedback




Trekist pvtlimited all copyright © unresolved

MyLike:


MY LIKE



Trekist Precision  
MRP : 5407 ₹




Trekist pegasus+  
MRP : 16805 ₹




Trekist GT  
MRP : 18095 ₹

MyCart:

MY CART



Trekist GT  
Size : 9  
MRP : 18095 ₹  
remove



Trekist Air Vapor Max  
Size : 5  
MRP : 19695 ₹  
remove





PRICE DETAILS

Price (2 items)	37790 ₹
Packaging Fee	+ 0 ₹
Delivery Charges	+ 0 ₹
Coupons for you	- 0 ₹
Total Amount	37790 ₹

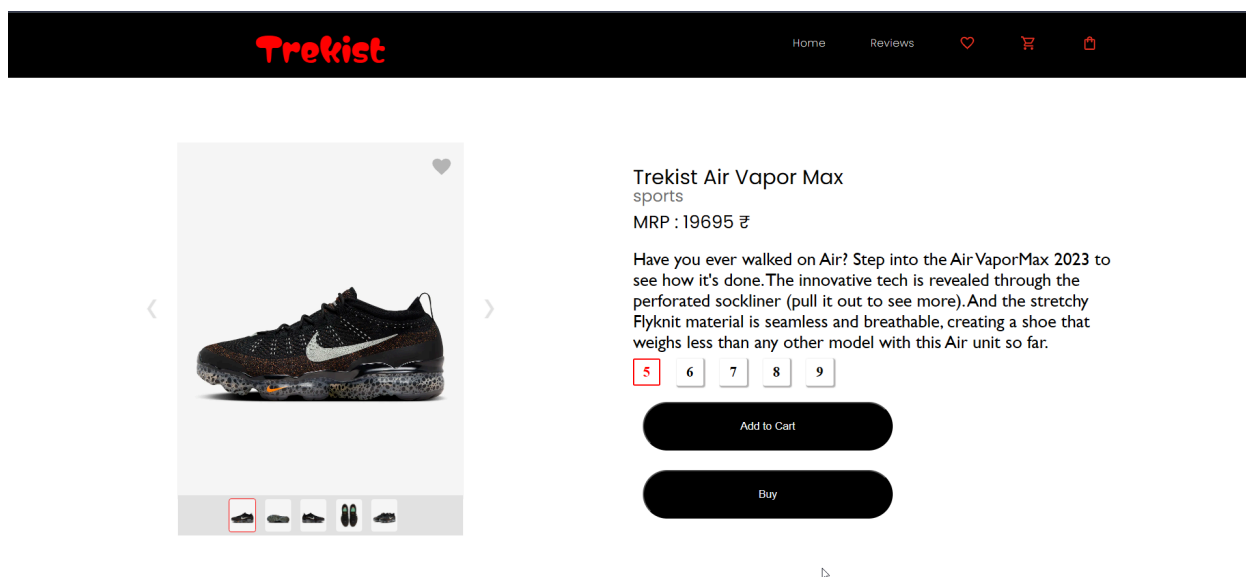
Place Order

### MyOrder:

## MY ORDER

	<p>Trekist Attack</p> <p>10257 ₹</p> <p>size : 5</p> <p>status : On the way</p>	<p>Order id : 66fa14313009219ee9ab187b</p> <p>Delivery Expected : 10/5/2024</p> <p>Add : jharkhand, sahibganj , Sahibgnaj jharkhand</p> <p>Mob : 1234</p>
	<p>Promo Exclusion</p> <p>14995 ₹</p> <p>size : 5</p> <p>status : On the way</p>	<p>Order id : 66fe5e5d8a41bad86f474150</p> <p>Delivery Expected : 10/8/2024</p> <p>Add : jharkhand, sahibganj , Sahibgnaj jharkhand</p> <p>Mob : 1234567</p>
	<p>Trekist Run 15</p> <p>8067 ₹</p> <p>size : 5</p> <p>status : On the way</p>	<p>Order id : 66fe5e5d8a41bad86f474152</p> <p>Delivery Expected : 10/8/2024</p> <p>Add : jharkhand, sahibganj , Sahibgnaj jharkhand</p> <p>Mob : 1234567</p>
	<p>Promo Exclusion</p> <p>14995 ₹</p> <p>size : 5</p> <p>status : On the way</p>	<p>Order id : 66fe5ebe081a857302a</p> <p>Delivery Expected : 10/8/2024</p> <p>Add : jharkhand, sahibganj , Sahibgnaj jharkhand</p> <p>Mob : 1234567</p>

**Shoe:**



# OrderForm:

ORDER FORM

Name

Mob no

State

Andhra Pradesh

City

Pincode

Local Address

CONFIRM ORDER

ORDER DETAILS

Total items

1

Items Name

Trekist Air Vapor Max

Total Payable Amount

19695₹

Payment Method

Cash On delivery

Expected deliver date

10/10/2024

# DataBase mongodb:

Atlas

AHSAN's Or...

Access Manager

Billing

All Clusters

Get Help

AHSAN

projectx

Data Services

Charts

Overview

DEPLOYMENT

Database

SERVICES

Atlas Search

Stream Processing

Triggers

Migration

Data Federation

SECURITY

Backup

Database Access

Network Access

Advanced

Goto

+ Create Database

Search Namespaces

portfolio

test

trekist

collection

orders

shoes

trekist.shoes

STORAGE SIZE: 112KB LOGICAL DATA SIZE: 91.43KB TOTAL DOCUMENTS: 73 INDEXES TOTAL SIZE: 34KB

Find

Indexes

Schema Anti-Patterns

Aggregation

Search Indexes

Generate queries from natural language in Compass

INSERT DOCUMENT

Filter

Type a query: { field: 'value' }

Reset

Apply

Options

QUERY RESULTS: 1-20 OF MANY

\_id: ObjectId('66efe8045d7043a09f8be597')

name: "Trekist Run 15"

price: 8967

size: Array (4)

description: "Styled for the '70s. Loved in the '80s. Classic in the '90s. Ready for..."

type: "Sports"

image: "https://static.nike.com/a/images/c\_limit,w\_592,f\_auto/t\_product\_v1/f6f..."

profile\_image: Array (5)

PREVIOUS

1-20 of many results

NEXT

System Status: All Good



## CONCLUSION

The **Trekist** e-commerce website project successfully delivers a streamlined and user-friendly platform for purchasing Trekist company shoes online. By utilizing modern web development technologies such as **HTML**, **CSS**, **JavaScript**, **Node.js**, and **MongoDB**, the project demonstrates a strong foundation in both frontend and backend development. The integration of features like product filtering, a simplified order process, and local storage for tracking orders enhances the user experience while maintaining ease of use.

The decision to avoid a login requirement ensures that customers can shop with minimal effort, making the platform accessible and efficient. Moreover, the use of a dynamic database for product listings and order management ensures scalability and adaptability for future growth.

The project showcases essential e-commerce functionalities, such as order processing, data storage, and product display, while leaving room for future enhancements such as payment gateway integration, user authentication, and more advanced product filtering options.

Overall, the **Trekist** website serves as an excellent starting point for developing robust, scalable e-commerce platforms, and it provides a valuable learning experience in full-stack web development. This project demonstrates practical skills that can be expanded further as the platform evolves.

