# Introduction To Project

In an era characterized by the relentless march of technology, the way we communicate has undergone a profound transformation. Gone are the days when we relied solely on face-to face conversations or traditional methods like letters and telegrams to convey our thoughts and feelings. Today, communication has evolved into a dynamic, real-time, and digitized process that transcends geographical boundaries. At the heart of this revolution lies the world of instant messaging and web chat applications.

The rapid proliferation of the internet and the ubiquitous presence of smartphones have catalyzed the ascent of instant messaging as the primary mode of communication for billions of people worldwide. From casual conversations among friends to vital business exchanges, the ability to send and receive messages in real time has become an integral part of our daily lives. As a result, there is an increasing demand for intuitive, feature-rich, and responsive web chat solutions that can keep pace with our digital communication needs.

This project, titled "Developing a Real-time Web Chat Box using Socket.io," is a response to the pressing need for a state-of-the-art web chat solution that addresses the limitations of existing platforms. Traditional chat applications, while certainly functional, often fall short when it comes to delivering real-time, scalable, and interactive experiences.

Furthermore, security and privacy have become paramount concerns in the digital age. Users expect their personal data and conversations to be safeguarded against unauthorized access and data breaches. Traditional chat applications may lack robust authentication and encryption mechanisms, leaving sensitive information vulnerable to interception or misuse.

To address these challenges and fulfill the evolving demands of users, our project seeks to create a web chat box that is not only real-time and scalable but also rich in features and security. The project harnesses the power of Socket.io, a Node.js library renowned for its ability to enable real-time, bidirectional communication between clients and servers.

# Need Of Project

The need for such a project arises from the increasing demand for efficient and interactive online communication platforms. Traditional chat applications often lack real-time capabilities and struggle to handle concurrent users. Our project addresses these limitations by harnessing the power of Socket.io to create a responsive and scalable chat solution.

This project, titled "Developing a Real-time Web Chat Box using Socket.io," is a response to the pressing need for a state-of-the-art web chat solution that addresses the limitations of existing platforms. Traditional chat applications, while certainly functional, often fall short when it comes to delivering real-time, scalable, and interactive experiences.

Additionally, modern chat users expect more than just text-based messaging. The ability to share multimedia content, including images, videos, and documents, is now an essential feature. Many existing platforms struggle to provide seamless multimedia sharing, hindering the richness of communication that users desire.

Furthermore, security and privacy have become paramount concerns in the digital age. Users expect their personal data and conversations to be safeguarded against unauthorized access and data breaches. Traditional chat applications may lack robust authentication and encryption mechanisms, leaving sensitive information vulnerable to interception or misuse.

To address these challenges and fulfill the evolving demands of users, our project seeks to create a web chat box that is not only real-time and scalable but also rich in features and security. The project harnesses the power of Socket.io, a Node.js library renowned for its ability to enable real-time, bidirectional communication between clients and servers.

**Objectives of Project:**

**Develop a User-friendly Web Chat Box**: A paramount goal of our project is to create a user friendly interface that is intuitive and visually appealing. We understand that the success of a chat application heavily depends on its ease of use, and we are committed to delivering an interface that welcomes both novice and experienced users.

**Implement Real-time Messaging**: Leveraging the power of Socket.io, we aim to provide users with a real-time messaging experience characterized by minimal latency. Instant message delivery is a core feature of our chat box, ensuring that conversations flow seamlessly without noticeable delays.

**Support Multimedia Sharing**: In recognition of the multimedia-rich nature of modern communication, our chat box will empower users to share images, videos, and other multimedia content effortlessly. Whether it's a funny meme, an informative video, or a memorable photo, users will have the tools to express themselves fully.

**Responsive User Interface**: To accommodate the diverse array of devices used by today's users, we are committed to creating a responsive and adaptable user interface. Whether users access the chat box on a desktop computer, tablet, or smartphone, they can expect a seamless and enjoyable experience.

# Data Collection and Sampling

## Data Collection

**1) User Profile Data:**

* Description: During the registration process, the chat box will collect user profile data, including usernames, email addresses, profile pictures, and potentially other optional user details.
* Purpose: User profile data is essential for personalizing the chat experience and facilitating user identification.
* Method: Users will provide this information during account creation, and it will be stored securely on the server.

**2) Message Data:**

* Description: The chat box will collect data related to messages, including text content, timestamps, sender and recipient identifiers, and message status (e.g., sent, delivered, read).
* Purpose: Message data is crucial for real-time message delivery and chat history.
* Method: Messages will be sent by users and received by the server in real-time. The data will be stored temporarily in server memory for processing and then stored in a database for chat history.

**3) Server Logs:**

* Description: Server logs will capture various server activities, such as message processing, user authentication, and system health metrics (e.g., CPU usage, memory usage).
* Purpose: Server logs aid in monitoring, debugging, performance optimization, and security monitoring.
* Method: The server will generate logs for different activities, and these logs will be stored securely, ensuring access only to authorized personnel.

## Data Sampling

**1) Performance Sampling:**

* Description: To monitor server performance, periodic sampling of system metrics will be conducted, including CPU usage, memory consumption, and network latency.
* Purpose: Performance sampling helps identify trends, anomalies, and potential bottlenecks that may affect the chat application's responsiveness.
* Method: Automated scripts or monitoring tools will sample performance data at regular intervals and generate reports or alerts based on predefined thresholds.

**2) User Analytics:**

* Description: Periodic sampling of user behavior data, such as login times, session durations, and feature usage, will provide insights into user engagement.
* Purpose: User analytics inform feature improvements, user experience enhancements, and marketing strategies.
* Method: Analytics tools will collect and aggregate user behavior data anonymously, allowing for trend analysis without compromising user privacy.

**3) Security Monitoring:**

* Description: Sampling of security-related data, such as login attempts, authentication failures, and unusual user activities, will be conducted to detect and respond to security threats.
* Purpose: Security monitoring helps identify and mitigate potential security risks.
* Method: Security systems and logs will periodically sample data to identify suspicious patterns or events that may indicate security breaches.

**HARDWARE & SOFTWARE**

**SPECIFICATION OF PROJECT**

**Hardware:**

**1) Development Computer**:

* Description: A standard personal computer or laptop will be used for the development and testing phases of the project.
* Purpose: This hardware will serve as the primary development environment for coding, debugging, and testing the web chat box application.

**Software:**

**1) Operating System:**

**•** Compatibility with major operating systems such as Windows, macOS, or Linux, depending on the project's target audience.

**2) Node.js:**

* Description: Node.js is a runtime environment that allows the execution of JavaScript on the server-side. It provides the foundation for building the chat application.
* Purpose: Node.js will be the core technology for server-side development, enabling real-time communication through Socket.io.

**3) Socket.io:**

* Description: Socket.io is a JavaScript library that facilitates real-time, bidirectional communication between the server and connected clients. It plays a pivotal role in enabling instant messaging capabilities.
* Purpose: Socket.io will be the primary communication framework, ensuring realtime message delivery and responsiveness.

**4) HTML, CSS, and JavaScript:**

* Description: These are fundamental web technologies used for creating the user interface (UI) of the chat box. HTML provides the structure, CSS handles styling, and JavaScript adds interactivity.
* Purpose: These technologies will be used to design and develop the chat box's userfriendly and responsive UI.

**5) Integrated Development Environment (IDE):**

* Description: An IDE, such as Visual Studio Code or any preferred development environment, will be used for coding, debugging, and project management tasks.
* Purpose: The IDE provides a comprehensive development environment, aiding in code editing, debugging, version control, and project organization.

**6) Web Server:**

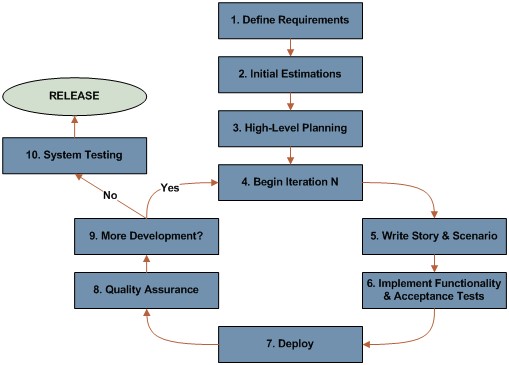
* Description: A web server software like Nginx or Apache may be used for serving the web application to users on the internet.
* Purpose: The web server will handle incoming HTTP requests and route them to the web chat box application for client access.

**7) Database System (e.g., MongoDB, MySQL, or PostgreSQL):**

* Description: A database management system will be used to store and manage data related to user profiles, message history, and application configurations.
* Purpose: The database system will ensure data persistence and retrieval for the chat application.

# Model will be implemented on project

## Agile Development Model Implementation



**Description:**

The Agile development model is a dynamic and iterative approach to software development that emphasizes collaboration, flexibility, and customer feedback throughout the project’s lifecycle. Unlike traditional linear models, such as Waterfall, Agile allows for incremental development, enabling the project team to adapt to changing requirements and deliver functional increments of the chat box application at regular intervals.

**Purpose:**

The adoption of the Agile development model for the chat box project serves several essential purposes:

1. Flexibility: Agile allows the project team to respond to changing requirements, priorities, and emerging user needs effectively. This flexibility is particularly valuable in dynamic and evolving projects like web chat applications.

1. Customer-Centricity: Agile places a strong emphasis on involving stakeholders and end-users throughout the development process. This ensures that the chat application aligns with user expectations and delivers value from the outset.

1. Continuous Improvement: Agile promotes continuous improvement through regular feedback and iterative development cycles. This approach enables the project team to identify and address issues promptly and refine features based on user input.

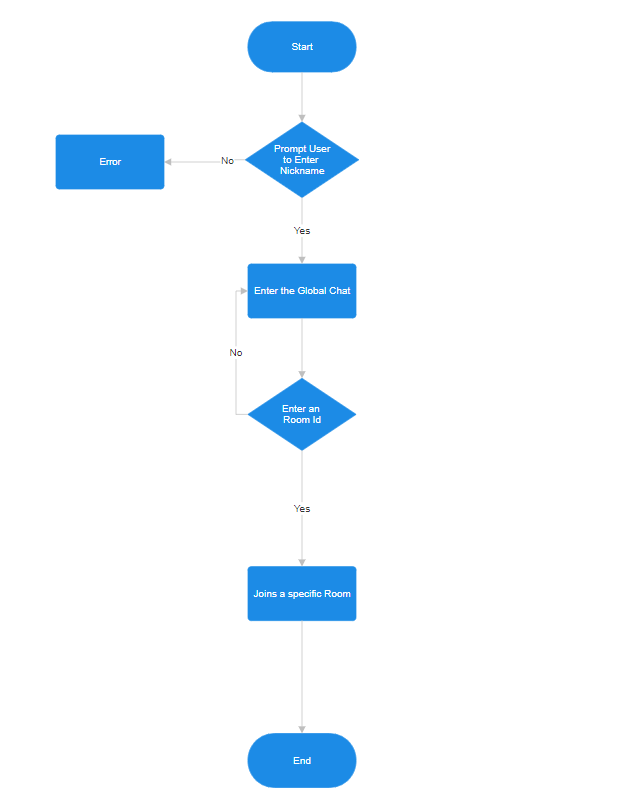
1. Efficiency: Agile's incremental development approach ensures that functional increments of the chat application are delivered at frequent intervals. This allows for rapid testing, validation, and deployment of features, maximizing efficiency.

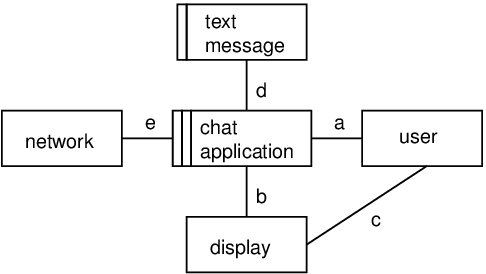
1. Collaboration: Agile fosters collaboration among team members, stakeholders, and users. Frequent communication and transparency are key principles, leading to better understanding and alignment among all parties involved.

**Iterative Development:**

Agile's iterative development approach allows for the continuous enhancement of the chat box application. Each sprint results in a potentially shippable product increment, meaning that valuable features and improvements are delivered regularly, ensuring the project remains responsive to evolving needs.

**Flow Chart Of Project**

****

**Use Case Diagram**

**Testing of Project**

**Unit Testing:**

* Client-Side Logic:

Test the client-side logic responsible for interacting with the user interface, handling user input, and managing the chat display.

* Server-Side Logic:

Unit test the server-side logic, including functions responsible for handling user commands, managing rooms, and broadcasting messages.

**Integration Testing:**

* User Interaction:

Test the integration between the client and server to ensure that user interactions, such as joining rooms and sending messages, work seamlessly.

* Room Management:

Verify that the application correctly manages different chat rooms, allowing users to join and leave as expected.

**End-to-End Testing:**

* Scenario Testing:

Perform end-to-end tests simulating common user scenarios, including opening the application, entering a username, joining the main chat room, sending messages, and joining/leaving rooms.

* Cross-Browser Testing:

Test the application on different web browsers to ensure compatibility.

**Performance Testing:**

* Load Testing:

Simulate a large number of users to assess how the application performs under heavy load.

* Message Handling:

Test the application's ability to handle a high volume of messages in the chat rooms.

**Cross-Platform Testing:**

* Device Compatibility:

Test the chat application on different devices (desktops, tablets, and mobile phones) to ensure responsiveness.

# Future of Project

## Language Translation Integration

* Description: The future of the chat box project envisions the integration of language translation capabilities. This feature will enable users from diverse linguistic backgrounds to communicate seamlessly, breaking down language barriers.

* Purpose: Language translation is crucial for expanding the chat box's global reach and fostering cross-cultural communication. It enhances inclusivity and user engagement by making the application accessible to a wider audience.

* Implementation Method: Advanced language translation APIs or libraries will be integrated into the chat box. Users will have the option to translate messages in real time, facilitating conversations between individuals who speak different languages.

## Enhanced Security and Privacy

* Description: The project's future roadmap includes a continued focus on enhancing security and privacy features. Robust encryption mechanisms, stricter access controls, and privacy settings will be implemented to safeguard user data and communications.

* Purpose: Maintaining user trust is paramount. By bolstering security and privacy, the chat box will ensure that user data remains confidential and protected from potential threats and breaches.

* Implementation Method: End-to-end encryption will be explored to provide the highest level of message confidentiality. Two-factor authentication, user authorization controls, and periodic security audits will be part of the ongoing security strategy.

## Video and Voice Calls

* Description: The chat box's future expansion includes the addition of video and voice calling functionalities. Users will be able to seamlessly transition from textbased messaging to video or voice calls, enhancing the richness of communication.

* Purpose: Video and voice calls add depth to conversations, allowing users to connect on a more personal level. These features are particularly valuable for remote collaboration, customer support, and staying connected with friends and family.

* Implementation Method: WebRTC (Web Real-Time Communication) technology will be leveraged to enable real-time video and voice calls within the chat box. Users will have the option to initiate calls directly from the chat interface.

## Innovative Features: Stickers and Chat Bots

* Description: To keep the chat box engaging and entertaining, the future of the project includes the introduction of innovative features such as stickers and chat bots. Stickers will add a fun and expressive dimension to messages, while chat bots will provide automated assistance and interactive experiences.

* Purpose: Stickers allow users to express emotions and sentiments creatively, enhancing the overall chat experience. Chat bots offer valuable services like answering frequently asked questions, providing recommendations, or assisting with tasks.

* Implementation Method: A library of stickers will be integrated into the chat interface, allowing users to select and send stickers as part of their messages. Additionally, chat bots with predefined functionalities will be deployed, with the potential for customization based on user needs.

## User-Driven Development

* Description: The future of the project emphasizes a user-driven development approach. Users will play a central role in shaping the direction of the chat box through feedback channels and feature requests.

* Purpose: By actively involving users in the development process, the chat box will continuously adapt to evolving user preferences and needs. User feedback will guide feature prioritization and enhancements.

* Implementation Method: User feedback mechanisms, such as in-app surveys, feedback forms, and user forums, will be established to collect insights and suggestions. Regular feedback analysis sessions will inform development priorities.

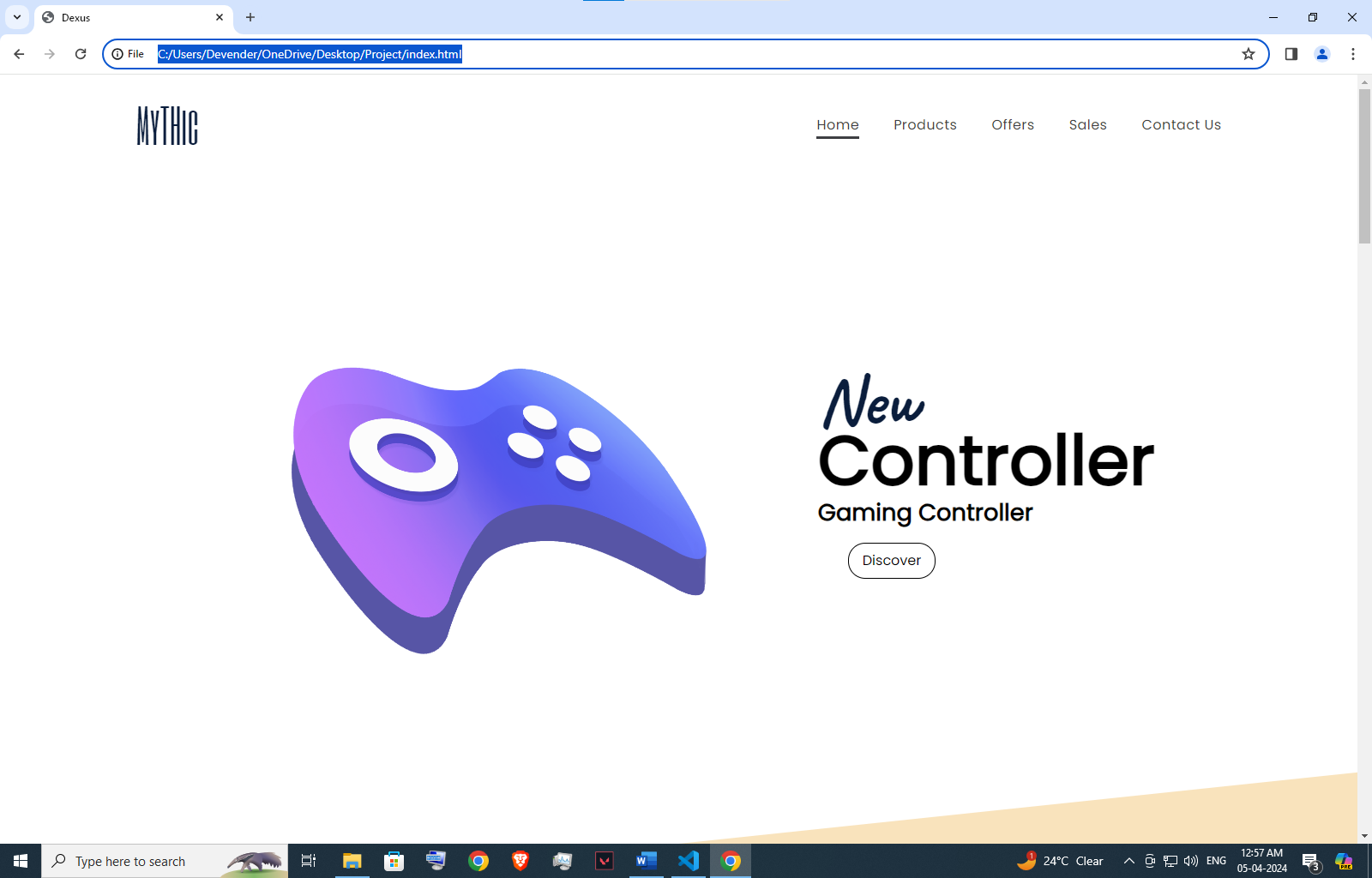
## Monetization and Sustainability

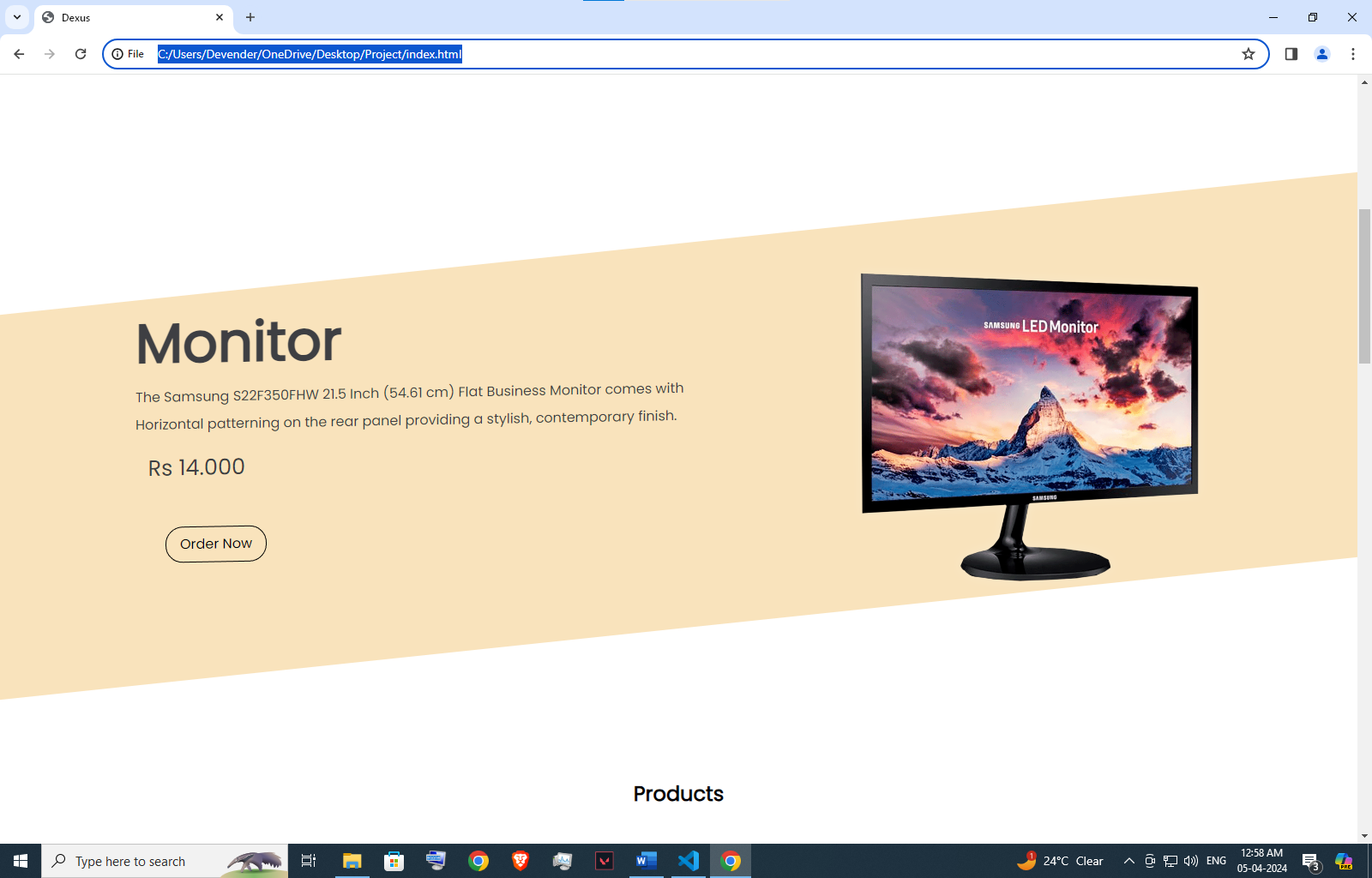
* Description: To ensure the long-term sustainability of the project, monetization strategies will be explored. These may include premium features, subscription models, or partnerships with relevant businesses.

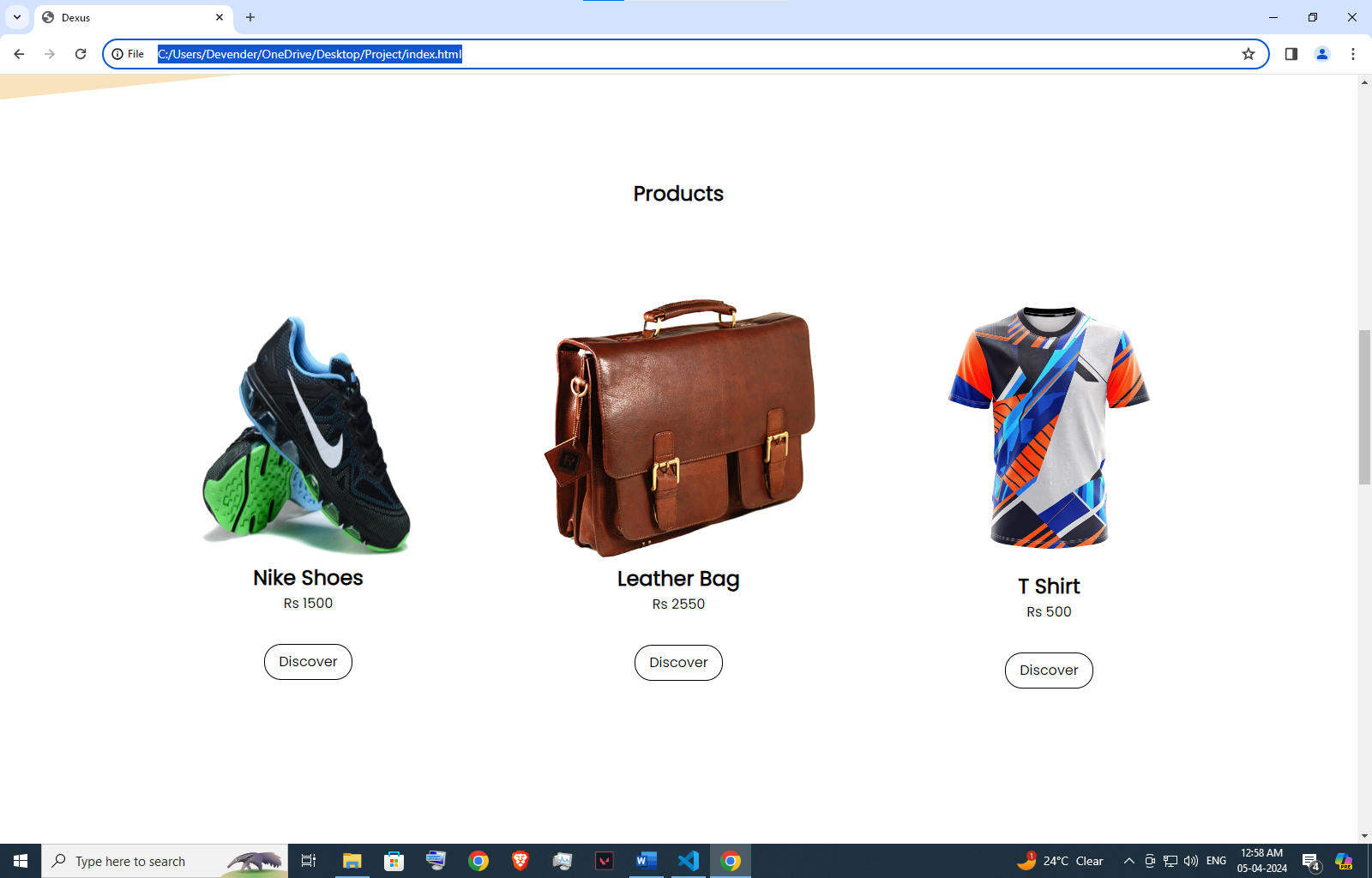
* Purpose: Sustainable revenue streams are essential for maintaining and improving the chat box, supporting ongoing development, and delivering a high-quality user experience.

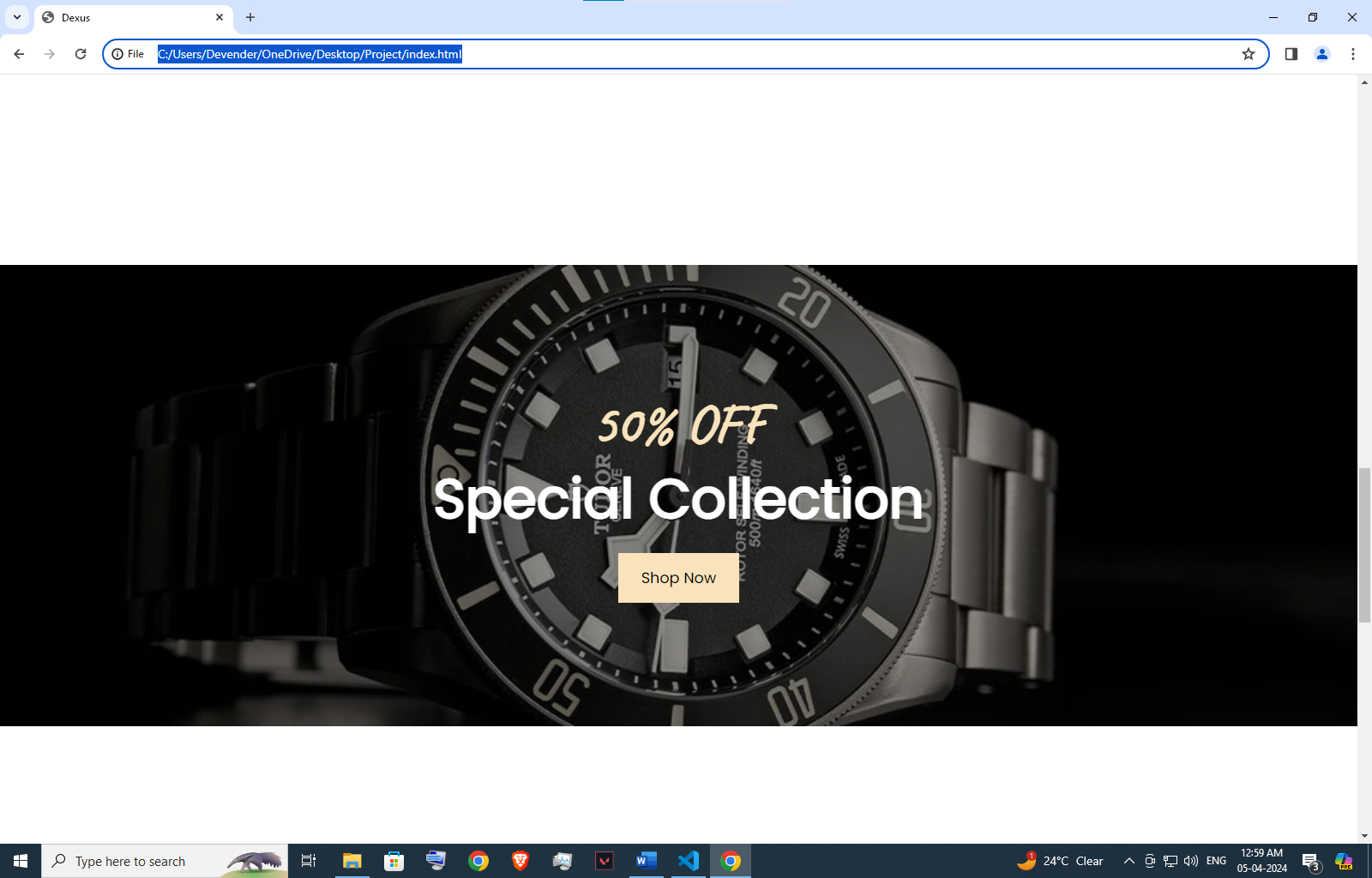
* Implementation Method: Monetization models will be evaluated and implemented thoughtfully, considering user value and affordability. Transparent communication with users regarding monetization strategies will be a priority.

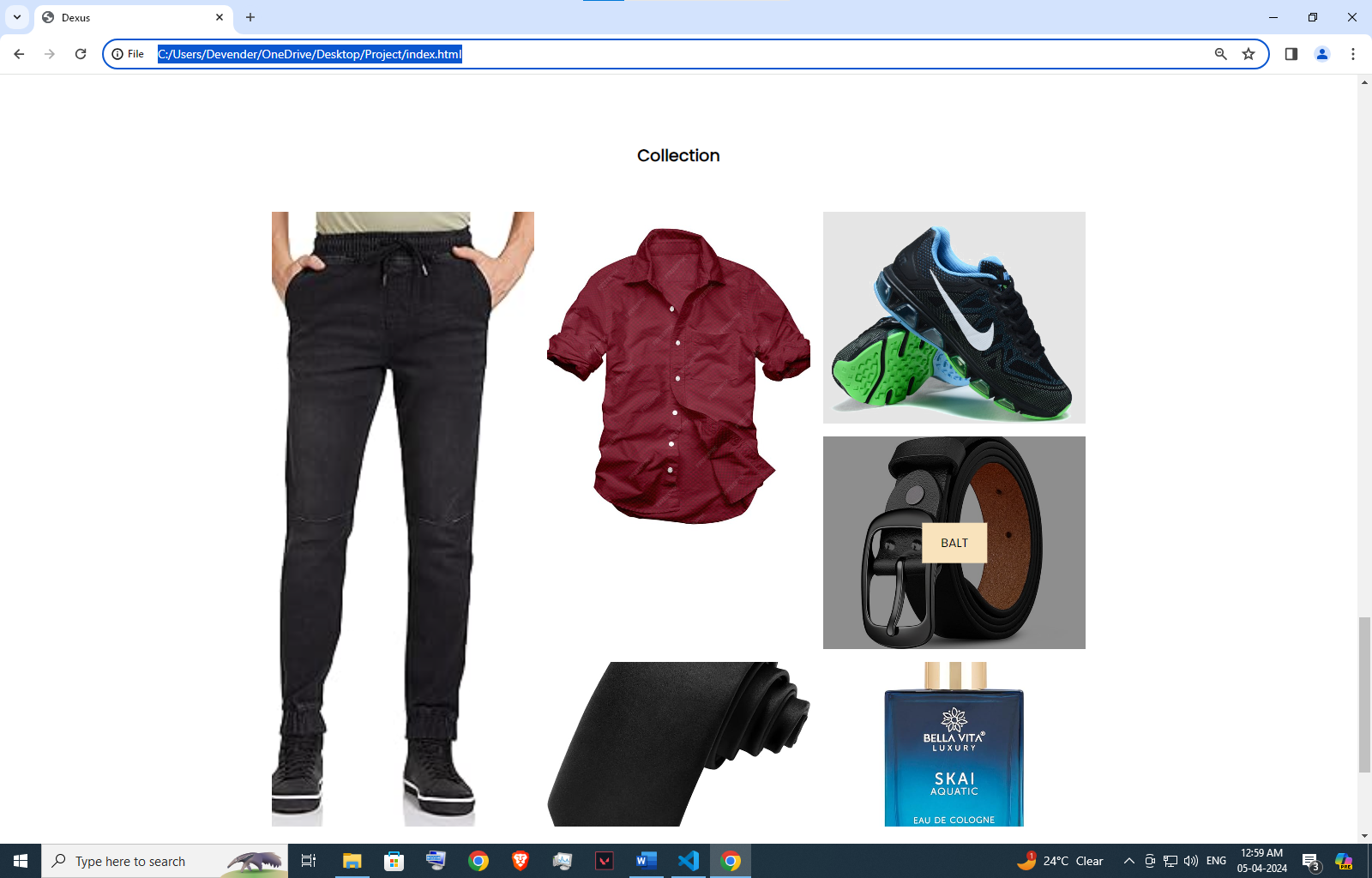
**SCREENSHOTS AND SOURCE CODE OF PROJECT**

****

****

****

****

****

**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>Dexus</title>

    <link rel="stylesheet" href="style.css">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css" integrity="sha512-Avb2QiuDEEvB4bZJYdft2mNjVShBftLdPG8FJ0V7irTLQ8Uo0qcPxh4Plq7G5tGm0rU+1SPhVotteLpBERwTkw==" crossorigin="anonymous" referrerpolicy="no-referrer" />

</head>

<body>

    <script src="main.js"></script>

    <!---section One--->

    <section class="section-one section">

        <nav>

            <div class="logo">MyTHic</div>

            <ul class="navlinks">

                <li>

                    <a href="" class="active">Home</a>

                </li>

                <li>

                    <a href="">Products</a>

                </li>

                <li>

                    <a href="">Offers</a>

                </li>

                <li>

                    <a href="">Sales</a>

                </li>

                <li>

                    <a href="">Contact Us</a>

                </li>

            </div>

            </ul>

            <div class="hamburger">

                <div class="line"></div>

                <div class="line"></div>

                <div class="line"></div>

            </div>

        </nav>

        <header>

            <div class="image-container">

                <img src="Image/Controller.png" class="header-image">

            </div>

            <div class="content">

                <h1 class="header-heading">Controller</h1>

                <h2 class="h2">Gaming Controller</h2>

                <div class="btn-container">

                    <button class="button">Discover</button>

                </div>

            </div>

        </header>

    </section>

    <!--Section Two-->

    <section class="section2 Section">

        <div class="section2-container">

            <div class="section2-inner-container">

                <div class="content">

                    <h1 class="section2-heading">Monitor</h1>

                    <p class="section2-p1">The Samsung S22F350FHW 21.5 Inch (54.61 cm) Flat Business Monitor comes with Horizontal patterning on the rear panel providing a stylish, contemporary finish.</p>

                    <p class="section2-cost">Rs 14.000</p>

                    <button class="section2-btn">Order Now</button>

                </div>

                <div class="image-container">

                    <img src="Image/Monitor.png" class="section2-image">

                </div>

            </div>

        </div>

    </section>

    <!-- ---section3--- -->

    <section class="section3 section">

        <br>

        <br>

        <br>

        <br>

        <br>

        <br>

        <br>

        <h1 class="section3-heading">Products</h1>

        <div class="card-container">

            <div class="card">

                <img src="Image/Shoes.png" class="card-image" height="300px" width="700px">

                <h2>Nike Shoes</h2>

                <p class="section3-Cost">Rs 1500</p>

                <button class="section2-btn">Discover</button>

            </div>

            <div class="card">

                <img src="Image/Leather Bag.png" class="card-image">

                <h2>Leather Bag</h2>

                <p class="section3-Cost">Rs 2550</p>

                <button class="section2-btn">Discover</button>

            </div>

            <div class="card">

                <img src="Image/T Shirt.png" class="card-image">

                <h2>T Shirt</h2>

                <p class="section3-Cost">Rs 500</p>

                <button class="section2-btn">Discover</button>

            </div>

        </div>

    </section>

    <!-- ---section4--- -->

    <section class="section4 section">

        <div class="section4-container">

            <div class="content">

                <h1 class="section4-off">50% off</h1>

                <h1 class="section4-sc">Special Collection</h1>

                <button class="section4-btn">Shop Now</button>

            </div>

        </div>

    </section>

    <!-- ---Section5-- -->

    <section class="section5 section">

        <h1  class="section3-heading">Collection</h1>

        <div class="grid-card-container">

            <div class="grid-card" Collection="Jeans"></div>

            <div class="grid-card" Collection="Shoes"></div>

            <div class="grid-card" Collection="Shirt"></div>

            <div class="grid-card" Collection="Tai"></div>

            <div class="grid-card" Collection="Balt"></div>

            <div class="grid-card" Collection="PerFume"></div>

        </div>

    </section>

    <!-- ---footer--- -->

    <footer>

        <p class="footer-Dvndr">Desiged by Dvndr

    </footer>

</body>

</html>

**Style.CSS**

@import url('https://fonts.googleapis.com/css2?family=Six+Caps&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Caveat:wght@700&display=swap');

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    font-family: "Poppins", sans-serif;

}

:root{

    --dark-blue : #0A1D3C;

    --light-grey: #E5E5E5;

    --white: #fff;

    --light-peach: #f9e3bc;

    --dark-grey: #3e3f43;

    --light-black: #1d1d1d;

}

/\* ---Section One--- \*/

.section{

    min-height: 100vh;

    width: 100%;

}

nav{

    width: 80%;

    height: 13vh;

    display: flex;

    align-items: center;

    justify-content: space-between;

    margin: auto;

}

header{

    width: 80%;

    height: calc(100vh - 13vh);

    display: flex;

    align-items: center;

    justify-content: space-evenly;

    margin: auto;

}

.navlinks{

    display: flex;

}

.header-image{

    width: 40rem;

}

.logo{

    font-size: 3.3rem;

    color: var(--dark-blue);

    font-family: "Six Caps", sans-serif;

    font-weight: 400;

    font-style: normal;

}

.navlinks li{

    list-style: none;

}

.navlinks li a{

    text-decoration: none;

    margin-left: 2.5rem;

    color: var(--dark-grey);

    font-size: 1rem;

    letter-spacing: .03rem;

    position: relative;

}

.header-heading{

    font-size: 5rem;

    position: relative;

}

.header-heading::before{

    content: "New";

    position: absolute;

    top: -60px;

    color: var(--dark-blue);

    font-family: "Caveat", cursive;

}

.h2{

    font-size: 1.7rem;

    line-height: 1px;

}

.button{

    outline: 0;

    grid-gap: 5px;

    align-items: center;

    background: 0 0;

    border: 1px solid #000;

    border-radius: 20px;

    cursor: pointer;

    display: inline-flex;

    flex-shrink: 0;

    font-size: 16px;

    gap: 5px;

    justify-content: center;

    line-height: 1;

    overflow: hidden;

    padding: 12px 16px;

    text-decoration: none;

    text-overflow: ellipsis;

    transition: all .14s ease-out;

    white-space: nowrap;

    outline: 0;

    grid-gap: 8px;

    align-items: center;

    margin: 35px;

}

.button:hover {

    box-shadow: 4px 4px 0 #000;

    transform: translate(-4px,-4px);

}

.button:focus-visible{

    outline-offset: 1px;

}

.navlinks li a::after, .navlinks li .active::after{

    content: "";

    background: var(--dark-grey);

    position: absolute;

    height: .15rem;

    width: 100%;

    left: 0;

    bottom: -5px;

    transform: scaleX(0);

    transition: transform(0.215);

    transform: right;

}

.navlinks li a:hover::after, .navlinks li .active::after{

    transform: scaleX(1);

    transform-origin: right;

}

.line{

    background: var(--dark-grey);

    width: 2.5rem;

    height: .2rem;

    margin: .38rem;

    border-radius: 2rem;

}

.hamburger{

    cursor: pointer;

    display: none;

}

/\* ---Section Two--- \*/

.section2{

    display: flex;

    align-items: center;

}

.section2-container{

    background: var(--light-peach);

    width: 100%;

    height: 50vh;

    transform: skewY(-6deg);

}

.section2-image{

    width: 28rem;

}

.section2-inner-container{

    transform: skewY(5deg);

    width: 80%;

    margin: auto;

}

.section2-container, .section2-inner-container{

    display: flex;

    justify-content: center;

    align-items: center;

}

.section2-heading{

    color: var(--dark-grey);

    font-size: 4rem;

}

.section2-p1{

    font-size: 1rem;

    color: var(--dark-grey);

    line-height: 2rem;

    max-width: 80%;

}

.section2-cost{

    font-size: 1.6rem;

    color: var(--dark-grey);

    margin: .9rem;

}

.section2-btn{

    padding: .9rem 1.7rem;

    border: .1rem solid var(--light-black);

    font-size: 1,1rem;

    background: transparent;

    transition: tranform .5s ease;

}

.section2-btn{

    outline: 0;

    grid-gap: 5px;

    align-items: center;

    background: 0 0;

    border: 1px solid #000;

    border-radius: 20px;

    cursor: pointer;

    display: inline-flex;

    flex-shrink: 0;

    font-size: 16px;

    gap: 5px;

    justify-content: center;

    line-height: 1;

    overflow: hidden;

    padding: 12px 16px;

    text-decoration: none;

    text-overflow: ellipsis;

    transition: all .14s ease-out;

    white-space: nowrap;

    outline: 0;

    grid-gap: 8px;

    align-items: center;

    margin: 35px;

}

.section2-btn:hover {

    box-shadow: 4px 4px 0 #000;

    transform: translate(-4px,-4px);

}

.section2-btn:focus-visible{

    outline-offset: 1px;

}

/\* Section Three \*/

.section3-heading{

    font-size: 2,5rem;

    text-align: center;

    margin-bottom: 4rem;

}

.card-container{

    display: flex;

    flex-wrap: wrap;

    justify-content: center;

    align-items: center;

    text-align: center;

    gap: 3rem;

}

.card-image{

    width: 20rem;

}

.card{

    padding: 2rem;

}

.section3-cost{

    font-size: 1.2rem;

    margin-top: .2rem;

    margin-bottom: 1.2rem;

}

/\* ---Section Four--- \*/

.section4{

    display: flex;

    align-items: center;

}

.section4-container{

    background-image: url("Image/Luxury-Watch-Brands-Like-Rolex-BANNER.jpg");

    background-position: center;

    background-size: cover;

    background-repeat: no-repeat;

    background-attachment: fixed;

    width: 100%;

    height: 60vh;

    display: flex;

    justify-content: center;

    align-items: center;

    text-align: center;

    position: relative;

}

.section4-off{

    font-size: 4rem;

    color: var(--light-peach);

    font-family: "Caveat", cursive;

    text-transform: uppercase;

}

.section4-sc{

    font-size: 4rem;

    color: var(--white);

    margin-bottom: 1rem;

}

.section4-btn{

    font-size: 1.1rem;

    padding: 1rem 1.7rem;

    background: var(--light-peach);

    border: none;

    transition: transform .5s ease;

}

.section4-btn:hover{

    transform: scale(1.1);

    cursor: pointer;

}

.section4-container::before{

    content: "";

    background: rgba(0, 0, 0,0.444);

    position: absolute;

    inset: 0;

}

.section4-container .content{

    z-index: 5;

}

/\* ---Section Five--- \*/

.grid-card-container{

    width: 60%;

    height: 80vh;

    margin: auto;

    display: grid;

    grid-template-columns: repeat(3, 1fr);

    grid-template-rows: repeat(2, 1fr);

    gap: 1.2rem;

    grid-auto-rows: 15rem;

}

.grid-card{

    background: var(--light-grey);

    position: relative;

    background-position: center;

    background-size: cover;

    background-repeat: no-repeat;

}

.grid-card:nth-child(1){

    grid-row: 1/4;

    background-image: url("Image/Jeans.jpg");

}

.grid-card:nth-child(2){

    background-image: url("Image/Shoes.png");

}

.grid-card:nth-child(3){

    grid-row: 1/3;

    background-image: url("Image/Shirt.jpg");

}

.grid-card:nth-child(4){

    grid-row: 3/4;

    background-image: url("Image/tai.jpg");

}

.grid-card:nth-child(5){

    background-image: url("Image/Balt.jpg");

}

.grid-card:nth-child(6){

    background-image: url("Image/Perfume.jpg");

}

.grid-card::before{

    content: "";

    position: absolute;

    background: rgba(0, 0, 0,0.444);

    inset: 0;

    transform: scaleX(0);

    transform-origin: right;

    transition: transform 1s ease-in-out;

}

.grid-card:hover::before{

    transform: scaleX(1);

    transform-origin: left;

}

.grid-card::after{

    content: attr(collection);

    position: absolute;

    text-transform: uppercase;

    background: var(--light-peach);

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

    padding: 1rem 1.7rem;

    font-size: 1.1rem;

    opacity: 0;

    visibility: hidden;

    transition: opacity 1s ease-in-out ;

}

.grid-card:hover::after{

    opacity: 1;

    visibility: visible;

}

/\* ---footer--- \*/

footer{

    background: var(--light-black);

}

.footer-Dvndr{

    color: white;

    font-size: 1.5rem;

    height: 20vh;

    width: 100%;

    margin-top: 5rem;

    display: flex;

    justify-content: center;

    align-items: center;

}

/\*---media Qeries---\*/

@media screen and (max-width: 500px){

    .header-image{

        display: none;

    }

    .navlinks{

        flex-direction: column;

        position: absolute;

        top: 0;

        background: var(--light-grey);

        width: 100%;

        height: 100%;

        justify-content: center;

        align-items: center;

        right: 100%;

        z-index: 2;

        gap: 2rem;

        transition: 1s ease-in-out;

    }

    .navlinks-active{

        right: 0;

    }

    .navlinks li a{

        font-size: 1.3rem;

        margin-left: 0;

    }

    .hamburger{

        display: block;

        z-index: 5;

    }

    .section .logo{

        z-index: 5;

    }

    .header-heading{

        font-size: 4rem;

    }

    .header{

        justify-content: flex-start;

    }

    /\*---Section Two---\*/

    .section2-image{

        display: none;

    }

    .section2-heading{

        font-size: 2.9rem;

    }

    .section2-p1{

        font-size: .93rem;

        max-width: 100%;

    }

    .section2-cost{

        font-size: 1,4rem;

    }

    .section2-btn{

        font-size: 1rem;

        padding: .7rem 1rem;

    }

    .section2-container{

        height: 60vh;

    }

    .section2-inner-container{

        width: 90%;

    }

    /\* --Section3-- \*/

    .card-image{

        width: 16rem;

    }

    /\* --Section-Four-- \*/

    .section4-heading{

        font-size: 2.8rem;

    }

    .section4-off{

        font-size: 3.3rem;

    }

    .section4-btn{

        font-size: 1rem;

    }

    /\* ---Section Five--- \*/

    .grid-card-container{

        grid-template-columns: repeat(2, 1fr);

        grid-template-rows: repeat(3, 1fr);

        width: 90%;

    }

    .grid-card:nth-child(1), .grid-card:nth-child(3){

        grid-row: 1/2;

    }

}