VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI-590018, KARNATAKA

2022-2023

A

Web Technology and its Applications Mini Project Report on [21CSL481]

"CLOTHING BRAND"

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE AND ENGINEERING

SAMITHA.M	**1AT21IS094**
RAKSHITHA.M	**1AT21IS083**
DEEKSHITHA.N	**1AT21IS031**

Under the Guidance of

Mrs. Meenatchi R
Assistant Professor
Dept of ISE
Atria I.T

Ms. Veena S
Assistant Professor
Dept of ISE
Atria I.T

ATRIA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University)
ASKB Campus, Anand Nagar,
Bengaluru-56002

Department of Information Science and Engineering

CERTIFICATE

Certified that the project work entitled **"CLOTHING BARND"** carried out by **SAMITHA.M(1AT21IS094),RAKSHITHA.M(1AT21IS083),DEEKSHITHA (1AT21IS031)**are Bonafide students of Department of Information Science and Engineering, ATRIA I.T., Bengaluru, in partial fulfilment for the award of Degree of **Bachelor of Engineering** in **Information Science & Engineering** of **Visvesvaraya Technological University, Belagavi,** during the academic year **2022-23**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Mrs. Meenatchi R Assistant Professor Department of ISE Atria I.T.	**Ms. Veena S** Assistant Professor Department of ISE Atria I.T.	**Dr. Shanthi Mahesh** Professor & HOD Department of ISE Atria I.T.	I/C Principal Atria I.T.

External Viva

Name of Examiners	**Signature with date**
1.	
2.	

DECLARATION

We,

SAMITHA.M(1AT21IS194),RAKSHITHA.M(1AT21IS083),DEEKSHITHA.N (1AT21IS031) students of 4th semester Bachelor of Engineering, Department of Information Science and Engineering, Atria Institute of Technology, Bengaluru, would hereby declare that the Web Technology and its Applications mini project entitled "CLOTHING BRAND" has been carried out by us at Atria Institute of Technology, Bengaluru, and submitted in partial fulfillment of the course requirement for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi, during the academic year 2022-23.

We further declare that, to the best of our knowledge and belief, the work embodied in this report has not been submitted to any other university or institution for the award of any other degree.

Place: Bengaluru

Signature of the student

Date:

SAMITHA.M
(1AT21IS194)

RAKSHITHA.M
(1AT21IS083)

DEEKSHITHA.N
(1AT21IS031)

ABSTRACT

Doctor Appointment Scheduling System is a software application designed to make it easier for patients to schedule appointments with doctors at convenient times. Getting an appointment involves going to the treatment center, searching for the best doctor for your condition, spending a lot of time, or making a phone call, or scheduling an appointment with a general doctor, allowing him or her to determine which specialists you ought to see is a long and tedious procedure. This framework makes use of Python as a back-end language, as well as a variety of other cloud resources. It helps to improve application growth by retrieving data for appointment scheduling. It has analytics, database, and messaging features that help users concentrate. This technology allows patients to schedule appointments and obtain time slots that are convenient for them and meet their needs, thus eliminating lengthy wait times. Patients may use this method to schedule appointments as well as explain the symptoms they are experiencing.

The Doctor Appointment Scheduling System is a software application designed to streamline and simplify the process of scheduling appointments with doctors. Traditionally, obtaining a medical appointment involved visiting a medical facility, searching for an appropriate doctor, spending a considerable amount of time, or making phone calls to schedule appointments. Sometimes, patients were required to schedule appointments with general doctors who would then determine which specialists the patients should see, leading to a time-consuming and tedious procedure.

the Doctor Appointment Scheduling System leverages technology to bridge the gap between patients and doctors. It offers a more convenient and efficient way for patients to access medical care, while also enhancing the overall workflow for medical facilities. The use of cloud resources, data analytics, and communication features contributes to the effectiveness of the system in providing a seamless appointment scheduling experience.

ACKNOWLEDGEMENT

We are grateful to our institution, **Atria Institute of Technology**, for having provided us with the facilities to successfully complete the Web Technology mini project on

Clothing brand

We thank **Dr. Aishwarya P, I/C Principal** and **Dr. Shanthi Mahesh, HOD, ISE** for providing us all the necessary facilities for the successful completion of our mini-project. Deadlines play a very important role in the successful completion of the academic project on time, efficiently and effectively.

We take this opportunity to express our deep sense of gratitude to our guide and coordinators **Mrs. Meenatchi R, & Ms. Veena S, Assistant Professor, Department of ISE** for their valuable guidance and help throughout the course of the academic mini-project. They have always been patient with us and helped immensely in completing the task on hand. We also thank them for their immense support, guidance, specifications & ideas without which seminar would have been completed without full merit.

Last but not least from the Department of Information Science and Engineering, teaching and non-teaching staffs for their constant encouragement, support, patience, and endurance shown during the preparation of this report were remarkable. We also thank the management.

Finally, we thank our parents and friends for their motivation, morale and material support.

SAMITHA.M
(1AT21IS194)

RAKSHITHA.M
(1AT21IS083)

DEEKSHITHA.N
(1AT21IS031)

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 **Introduction to Web Technology**

Web technology refers to the tools, software, protocols, and programming languages that are used to create and manage websites and web applications on the World Wide Web (WWW). It encompasses a wide range of technologies that work together to enable the functioning of websites, from the basic structure and design to the complex interactions and data processing.

Fig: 1.1.1 Web Technology

Web Technology can be classified into the following sections:

- World Wide Web (WWW): The World Wide Web is based on several different technologies: Web browsers, Hypertext Markup Language (HTML), and Hypertext Transfer Protocol (HTTP).

- Web Browser: The web browser is an application software to explore www (World Wide Web). It provides an interface between the server and the client and requests to the server for web documents and services.

- Web Server: Web server is a program which processes the network requests of the users and serves them with files that create web pages. This exchange takes place using Hypertext Transfer Protocol (HTTP).

- Web Pages: A webpage is a digital document that is linked to the World Wide Web and viewable by anyone connected to the internet has a web browser.
- Web Development: Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e., websites.

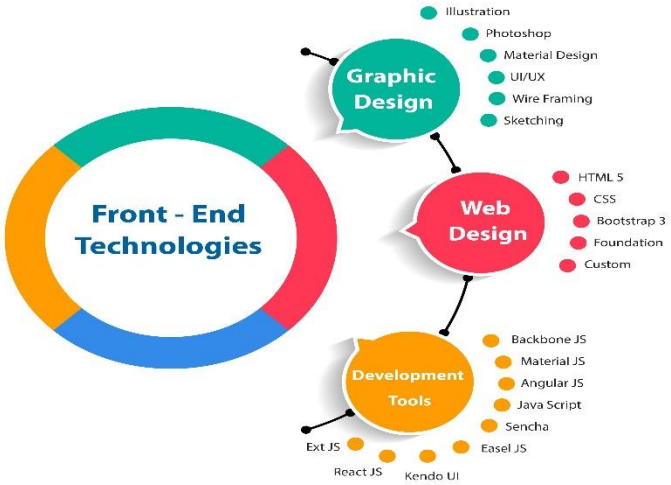


Fig:1.1.2 Types of Web Pages

Key Components of Web Technology:

- Hypertext Markup Language (HTML): HTML is the foundation of web technology. It is a markup language used to structure the content of web pages, defining headings, paragraphs, links, images, and other elements.

- Cascading Style Sheets (CSS): CSS is used to control the presentation and layout of HTML elements. It allows designers to define colors, fonts, spacing, and other visual aspects of a website.

- JavaScript: JavaScript is a programming language that adds interactivity and dynamic behavior to web pages. It enables features like animations, form validation, and real-time updates without requiring a page refresh.

- Web Browsers: Web browsers like Chrome, Firefox, Safari, and Edge are software applications that display web content. They interpret HTML, CSS, and JavaScript to render web pages to users.

- Web Servers: Web servers are software or hardware systems that store and deliver web content to users' browsers. They respond to requests from browsers and provide the necessary files to display web pages.

- HTTP and HTTPS: Hypertext Transfer Protocol (HTTP) is the foundation of data communication on the web. It defines how data is exchanged between web servers and browsers. HTTPS (HTTP Secure) adds a layer of security using encryption.

- Web Hosting: Web hosting involves storing web files on a server that is accessible over the internet. It allows websites to be available to users 24/7. There are various types of web hosting, including shared hosting, dedicated hosting, and cloud hosting.

- Web Development Frameworks: Frameworks like React, Angular, and Vue.js provide pre-built components and tools for building interactive web applications more efficiently.

- Content Management Systems (CMS): CMS platforms like WordPress, Joomla, and Drupal allow users to create, edit, and manage digital content on their websites without extensive coding knowledge.

- Responsive Design: With the proliferation of various devices (smartphones, tablets, desktops), responsive design ensures that websites adapt and display correctly on different screen sizes.

- APIs (Application Programming Interfaces): APIs allow different software applications to communicate and share data. Web APIs enables developers to integrate third-party services into their websites or build their own services.

- Web Security: Web technology includes security measures to protect user data and prevent unauthorized access. This includes features like user authentication, encryption, and secure coding practices.

- Web Standards and Accessibility: Web standards ensure consistency and compatibility across different browsers. Accessibility standards ensure that websites are usable by people with disabilities.

- Web Development Lifecycle: The process of creating a website involves various stages, including planning, design, development, testing, deployment, and maintenance.

1.2 Introduction to Service Oriented Architecture

Service-Oriented Architecture (SOA) is a design approach and architectural pattern that facilitates the creation, deployment, and management of software systems by breaking down complex applications into smaller, reusable, and interoperable services. These services are designed to perform specific business functions and can communicate with each other over a network, often using standardized protocols like HTTP, SOAP (Simple Object Access Protocol), or REST (Representational State Transfer).

The fundamental idea behind SOA is to promote modularity, reusability, and flexibility in software development. Instead of building monolithic applications where all functionalities are tightly integrated, SOA encourages the development of discrete, self-contained services that can be developed, maintained, and scaled independently. This modular structure offers several benefits:

- Reusability: Services can be developed once and reused across different applications, reducing development time and effort.

- Interoperability: Services can be developed using different technologies and platforms, allowing diverse systems to communicate effectively.

- Scalability: Services can be scaled independently based on demand, without affecting the entire application.

- Flexibility: New services can be added or existing ones modified without disrupting the entire system.

- Maintainability: Since services are isolated, maintenance and updates can be focused on specific components, making the overall system easier to manage.

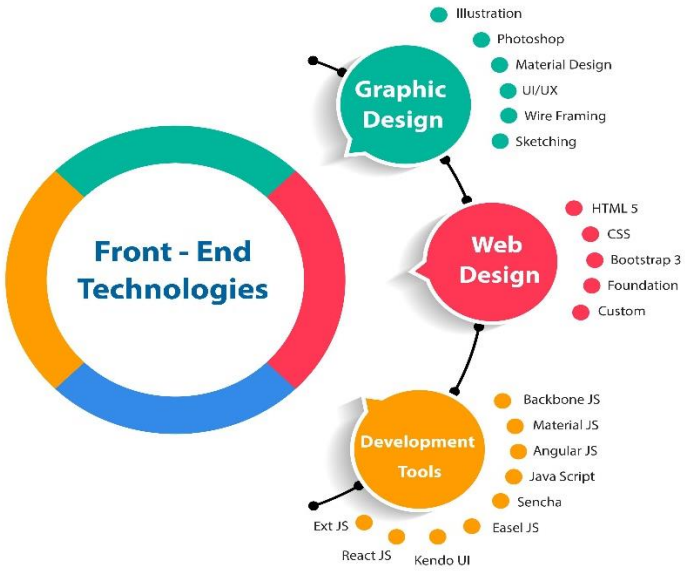


Fig:1.2.1 Service Oriented Architecture

Advantages of SOA:

- In SOA, applications are made from existing services. Thus, services can be reused to make many applications.
- As services are independent of each other they can be updated and modified easily without affecting other services.
- SOA allows making a complex application by combining services picked from different sources, independent of the platform.
- SOA facilities are easily available to anyone on request.

- SOA applications are more reliable because it is easy to debug small services rather than huge codes
- Services can run on different servers within an environment, this increases scalability

Disadvantages of SOA:

- High overhead: A validation of input parameters of services is done whenever services interact this decreases performance as it increases load and response time.
- High investment: A huge initial investment is required for SOA.
- Complex service management: When services interact they exchange messages to tasks. the number of messages may go in millions. It becomes a cumbersome task to handle a large number of messages.

1.4 Advantages of Web Technology

- Global Reach: The internet allows information and services to be accessible to a worldwide audience. This global reach enables businesses and individuals to connect with people from different parts of the world, expanding their potential customer base or audience.

- Accessibility: Web technologies enable people to access information and services from anywhere, using various devices like computers, smartphones, and tablets. This accessibility has made it easier for individuals to stay connected and informed on the go.

- Cost-Efficiency: Setting up and maintaining a web presence is often more cost-effective than traditional methods of communication or business operations. Online marketing, for example, can be more affordable and reach a larger audience compared to traditional advertising methods.

- Convenience: Web technology allows users to access services, shop, communicate, and perform various tasks without the need to physically visit a location. This level of convenience has led to the rise of e-commerce, online banking, remote work, and more.

- Information Sharing: The web facilitates easy sharing of information, whether it's through websites, social media, or online collaboration tools. This has led to the rapid dissemination of knowledge and ideas, fostering collaboration and innovation.

- Personalization: Websites and online services can tailor content and experiences based on user preferences and behavior. This personalization enhances user engagement and satisfaction, making interactions more relevant and meaningful.

- Scalability: Web applications and services can often be scaled up or down easily to accommodate changing user demands. Cloud computing and virtualization have further enhanced the scalability of web-based systems.

- Real-Time Communication: The web enables real-time communication through platforms like instant messaging, video conferencing, and social media. This is crucial for both personal interactions and business collaborations.

- Automation: Web technology has enabled automation of various tasks and processes, leading to increased efficiency and reduced human intervention. This is evident in areas like e-commerce order processing, customer service chatbots, and more.

- Analytics and Data Collection: Web technologies allow businesses and organizations to collect valuable data about user behavior, preferences, and interactions. This data can be analyzed to make informed decisions, improve user experiences, and optimize business strategies.

- Innovation and Creativity: The open nature of the web encourages innovation and creativity. Developers and content creators can experiment with new ideas and technologies, leading to the continuous evolution of the digital landscape.

- Flexibility and Customization: Web applications and websites can be customized to meet specific needs and requirements. This flexibility allows businesses to create unique online experiences that align with their brand and goals.

- Environmental Impact: The shift towards web-based services has the potential to reduce the environmental impact associated with physical products and transportation. Digital communication and transactions can contribute to a reduction in paper waste and carbon emissions.

- Learning and Education: Web technology has revolutionized the way education is delivered. Online courses, e-learning platforms, and educational resources are easily accessible, allowing people to learn at their own pace and from a variety of sources.

CHAPTER 2: SYSTEM REQUIREMENTS SPECIFICATION

2.1 Introduction

- A System Requirements Specification (SRS) is a crucial document in the field of web technology that outlines the detailed specifications, features, functionalities, and constraints of a software system or application.

- It serves as a blueprint that guides developers, designers, and stakeholders throughout the software development lifecycle, ensuring that everyone involved has a clear understanding of what needs to be built and how it should behave.

- Service-Oriented Architecture (SOA) is a fundamental concept in web technology and software architecture that focuses on designing and organizing software systems as a collection of interconnected and reusable services.

- These services are self-contained, modular components that perform specific functions and can be accessed over a network, typically the internet.

2.1.1 Purpose

- A System Requirements Specification (SRS) serves a crucial purpose as a comprehensive and formal document that outlines the specifications, features, functionalities, and constraints of a software system or application.

- The primary purpose of an SRS in web technology is to provide a clear and detailed roadmap for the development team, designers, stakeholders, and other involved parties.

- The System Requirements Specification (SRS) in web technology is an essential document that outlines the project's objectives, scope, functionalities, and constraints.

- It facilitates clear communication, guides development efforts, ensures alignment among stakeholders, and provides a foundation for successful project execution.

2.1.2 Scope

- The scope of a System Requirements Specification (SRS) refers to the boundaries and extent of what the software system or application will include, as well as what it will not include.

- Defining the scope clearly is essential to prevent scope creep, ensure alignment between stakeholders and the development team, and manage expectations regarding the final product.

- Defining the scope of a web technology project in the SRS is critical for ensuring that all stakeholders have a shared understanding of the project's objectives and limitations.

- It helps the development team focus on building the right functionalities and guides the project toward successful completion within the agreed-upon boundaries

3.2 Interface Description

- The interface description in an SRS specifically focuses on how users will interact with the system, including user interfaces, navigation, and user experience.

- The actual content and structure of the interface description will vary based on the specific project, its requirements, and the stakeholders' preferences.

- The goal is to provide a clear understanding of how users will interact with the web application and what their experience will be like.

3.2.1 Front End

- The front-end of a web application is responsible for creating the user interfaces and interactions as described in the System Requirements Specification (SRS) interface description.

- It’s the part of the application that users directly interact with.

- The front-end code is typically structured using HTML, CSS, and JavaScript.

- HTML: Defines the basic structure of the web page, including sections like header, main content, and footer. It also includes links to external stylesheets and scripts.

- CSS: Stylesheets define the visual appearance of elements, encompassing colors, fonts, spacing, and layout. This ensures a consistent and visually appealing design.

- JavaScript: Script files contain client-side interactions and functionality. They handle user input, make asynchronous requests to the back-end, and enhance the user experience.

- In a real-world scenario, frameworks and libraries such as React, Angular, or Vue.js are often used to efficiently manage and build the front-end.

- These tools help create reusable components, manage state, handle dynamic interactions, and ensure responsive designs for various screen sizes.

- The front-end’s actual code implementation involves detailed design elements, responsive layouts, and the realization of specific features outlined in the SRS interface description.

- It’s a crucial part of the web application development process, as it directly shapes how users perceive and interact with the system.

3.3 Specific Requirements

- Specific requirements in a System Requirements Specification (SRS) document outline detailed, clear, and unambiguous statements about the functionalities, behaviors, and characteristics of a software system.

- These requirements serve as a bridge between the high-level objectives and the technical implementation, providing developers and stakeholders with a comprehensive understanding of what the system needs to achieve.

3.3.1 Software requirements

In a System Requirements Specification (SRS) document, the software requirements section outlines the specific details that a software system must adhere to. These requirements are divided into several key components:

- Functional Requirements: Detailed descriptions of the specific functions or features the software must perform, such as user account creation or report generation.

- Non-Functional Requirements: Quality attributes like performance, security, and user experience expectations, including response times, scalability, and supported platforms.

- User Interface Requirements: Specifications for the appearance, layout, and behavior of the user interface, including design elements and navigation.

- System Architecture: High-level explanations of the software's structure, including components, modules, and interactions, often accompanied by diagrams.

- Data Requirements: Details about the data the software needs to handle, including formats, data flow, and constraints.

3.3.2 Hardware requirements

In the context of a System Requirements Specification (SRS) document, hardware requirements encompass the specifications and configurations needed for the optimal functioning of the software system on physical hardware.

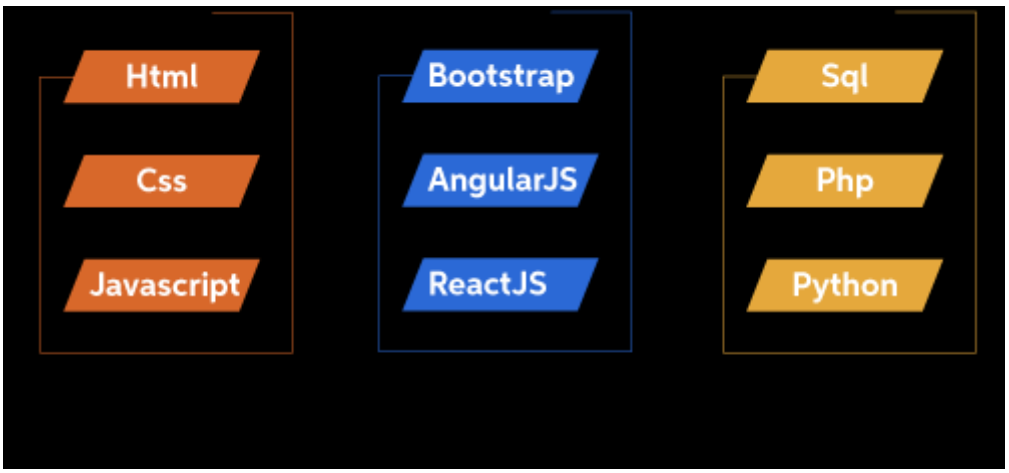
- Minimum and Recommended Specs: Clearly define the baseline hardware specifications required for running the software (minimum requirements) as well as the hardware that will provide an optimal experience (recommended requirements).

- Operating System Compatibility: Specify the operating systems and their versions that the software is compatible with.

- Processor, RAM, and Storage: Detail the necessary processor speeds, memory (RAM) amounts, and storage capacities for efficient software operation.

- Graphics and Display: If graphics play a role, outline the graphics card capabilities and display resolutions needed.

- Input/Output (I/O): Specify any required hardware interfaces or peripherals, such as printers or scanners.



CHAPTER 3: DESIGN

Designing web technology involves creating and optimizing the various components that make up the modern web ecosystem.

This includes everything from the underlying protocols and languages to the user interfaces and experiences. Here's an overview of the key aspects involved in designing web technology:

Web Protocols and Standards:

- HTTP/HTTPS: The Hypertext Transfer Protocol is the foundation of data communication on the web. HTTPS adds a layer of security using encryption.
- HTML: Hypertext Markup Language is used to structure content on web pages.
- CSS: Cascading Style Sheets are used to style and format the appearance of HTML elements.

Front-End Development:

- Responsive Design: Designing websites to adapt seamlessly to different devices and screen sizes.
- Web Accessibility: Ensuring that websites are designed to be usable by people with disabilities.
- Web Performance Optimization: Optimizing web assets (images, scripts, styles) for faster loading times.

Back-End Development:

- Server-Side Languages: Using languages like Python, Ruby, Java, or Node.js to build the logic and functionality behind web applications.
- Databases: Storing and retrieving data using databases like MySQL, PostgreSQL, MongoDB, or Redis.
- Server Management: Deploying and managing servers to host web applications, using platforms like AWS, Azure, or Google Cloud.

Web Security:

- Authentication and Authorization: Implementing secure login and access control mechanisms.
- Data Encryption: Protecting sensitive data during transmission and storage using encryption techniques.
- Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) Prevention: Mitigating common security vulnerabilities.

Web APIs:

- RESTful APIs: Designing APIs that follow the principles of Representational State Transfer for communication between client and server.
- Graph QL: An alternative to REST APIs that allows clients to request specific data structures, reducing over-fetching and under-fetching of data.

Web Standards and Compatibility:

- Browser Compatibility: Ensuring that websites work consistently across different web browsers (Chrome, Firefox, Safari, Edge, etc.).
- W3C Standards: Following guidelines set by the World Wide Web Consortium to ensure interoperability and consistent behavior.

3.1 High level design

- High-level design in web technology refers to the conceptual and architectural planning of a web application or system.

- It involves outlining the major components, interactions, and structures of the system before delving into detailed implementation. Here's an overview of the key aspects of high-level design in web technology:

1.System Architecture:

- Define the overall structure of the application, including the front-end and back-end components.

- Choose the architecture pattern, such as monolithic, microservices, serverless, or SPA (Single Page Application).

2.Front-End Design:

- Identify the user interface components and layout.
- Plan the user flows and navigation.

3.Back-End Design:

- Design the server-side components responsible for business logic, data processing, and storage.
- Choose the programming language (e.g., Python, Java, Node.js) and framework for the back-end.

4.Data Management:

- Define the data models and schemas for the application's data.
- Plan how data will be stored, retrieved, updated, and deleted.

5. Security Design:

- Identify potential security threats and vulnerabilities.
- Plan authentication and authorization mechanisms for user access.

6. Scalability and Performance:

- Design for scalability by considering how the system can handle increased load and user traffic.
- Plan strategies for load balancing, caching, and database optimization.

7. Integration and APIs:

- Plan the APIs that will enable communication between front-end and back-end components.
- Determine the data formats (JSON, XML) and authentication mechanisms for APIs.

8. Deployment and Hosting:

- Decide on the hosting environment (on-premises, cloud-based) and platform (AWS, Azure, Google Cloud).
- Consider scalability and redundancy in the deployment architecture.

9. Monitoring and Analytics:

- Plan how to monitor the application's performance, uptime, and user interactions.
- Integrate analytics tools to gather insights into user behavior and application usage.

10. User Experience (UX) Design:

- Ensure that the overall design aligns with a positive user experience.
- Plan user journeys, interactions, and feedback mechanisms.

CHAPTER 4: IMPLEMENTATION

HTML

```
<!doctype html>
<html lang="en">
 <head>
  <style>
        body {
            background-image: url('https://i.ibb.co/XzPGY4P/Picsart-23-09-06-23-12-06-255.jpg');
            background-size: cover;
            background-repeat: no-repeat;
            background-attachment: fixed;
        }
    </style>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Clothing Brand</title>
  <style>
        body {
            margin: 0;
            padding: 0;
            background-image: url('https://i.ibb.co/XzPGY4P/Picsart-23-09-06-23-12-06-255.jpg');
            background-size: cover;
            background-position: center;
            font-family: Arial, sans-serif;
        }
        .container {
            text-align: center;
            position: absolute;
            top: 30%;
            left: 48%;
```

HTML

```
        }
        .shop-button {
            padding: 10px 20px;
            background-color: #ff6600;
            color: #ffffff;
            text-decoration: none;
            font-weight: bold;
            border-radius: 5px;
            transition: background-color 0.3s ease;
        }
        .shop-button:hover {
            background-color: #ff4500;
        }
    </style>
</head>
<body>
 <div class="container">
  <h1><b></b>
   <p class="center-text"><b>Welcome to our clothing store</b></p></h1>
  <h3><b>Discover the latest fashion trends</b></h3><a class="shop-button" href="https://i.ibb.co/xCvyh9w/IMG-20230907-101043.png">Shop Now</a>
 </div>
 <style>
        /* Define CSS rules to center the text horizontally and change the text color */
        .center-text {
            text-align: center; /* Center the text horizontally */
            color: #ff6600; /* Change the text color */
        }
    </style>
```

HTML

```
<!doctype html>
<html>
 <head>
  <style>
  .container {
    position: relative;
    width: 100%;
    text-align: center;
  }

  .background-image {
    width: 100%;
    height: auto;
  }

  .left-half{
    width:40%;
    float:right;
  }

  .right-half{
    width:40%;
    float:left;
  }

  .shop-button {
    display: inline-block;
    margin-top: 50%;
```

HTML

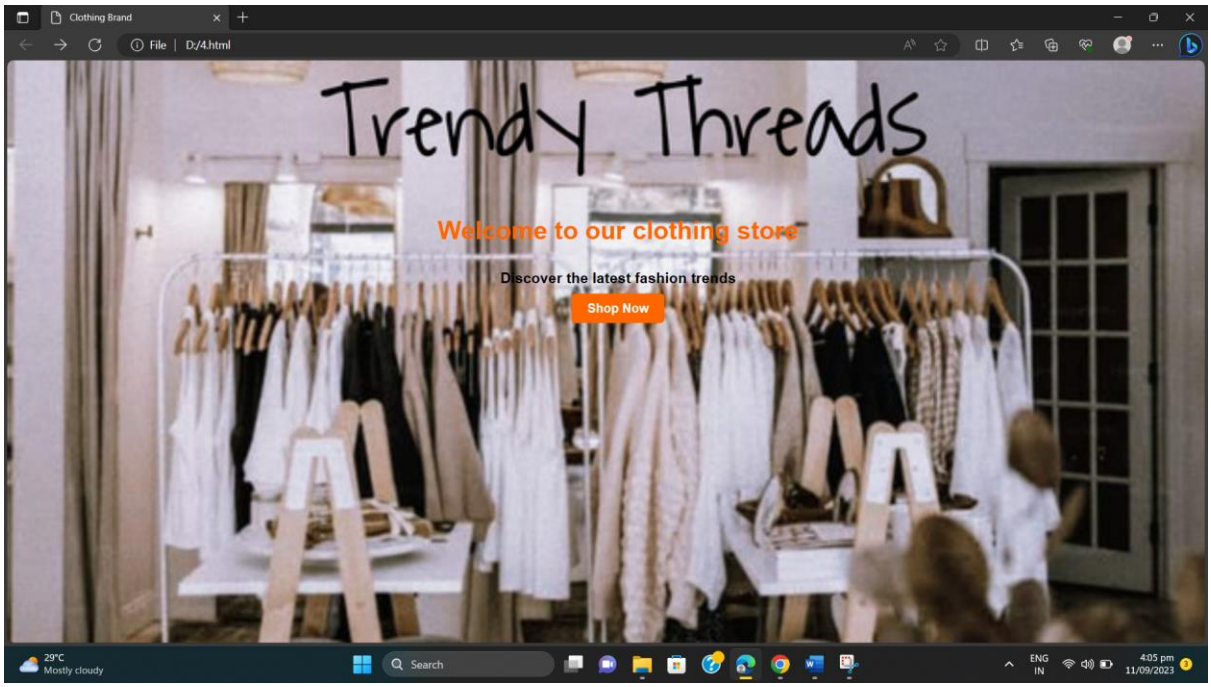
```
    transform: translateY(-1550%);
  }

  /* Additional styling for your shop buttons */
  .shop-button a {
    padding: 1px 5px;
    background-color: #ff6600;
    color: #fff;
    border: none;
    border-radius: 5px;
    text-decoration: none;
  }
</style>
 </head>
 <body>
  <div class="container">
   <img class="background-image" src="https://i.ibb.co/xCvyh9w/IMG-20230907-101043.png" alt="Background Image">
   <div class="left-half">
    <div class="shop-button">
     <a href="https://i.ibb.co/fqbLKS5/IMG-20230907-125220.jpg">Shop now</a>
    </div>
   </div>
   <div class="right-half">
    <div class="shop-button">
     <a href="https://i.ibb.co/1zpxF6h/IMG-20230907-125232.jpg">Shop now</a>
    </div>
   </div>
  </div>
```

HTML 15 unsaved changes

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Clothing Brand - Product Page</title>
    <style>
        body {
            background-color: beige;
        }
        .product-container {
            display: flex;
            justify-content: space-around;
            padding: 20px;
        }
        .product {
            background-color: white;
            border: 1px solid #ddd;
            padding: 20px;
            margin: 10px;
            text-align: center;
        }
        .product img {
            max-width: 100%;
            height: auto;
        }
    </style>
</head>
<body>
    <h1><center>Accessories</center></h1>
    <div class="product-container">
        <div class="product">
            <img src="https://i.ibb.co/9tNZJY9/IMG-20230907-123153.jpg" alt="Product 1">
        </div>
        <div class="product">
            <img src="https://i.ibb.co/23Sd3Fj/IMG-20230907-122759.jpg" alt="Product 2">
        </div>
        <div class="product">
            <img src="https://i.ibb.co/Q6KW8vQ/IMG-20230907-122842.jpg" alt="Product 3">
        </div>
        <div class="product">
            <img src="https://i.ibb.co/CbtNy2G/IMG-20230907-122738.jpg" alt="Product 4">
        </div>
    </div>
</body>
</html>
```

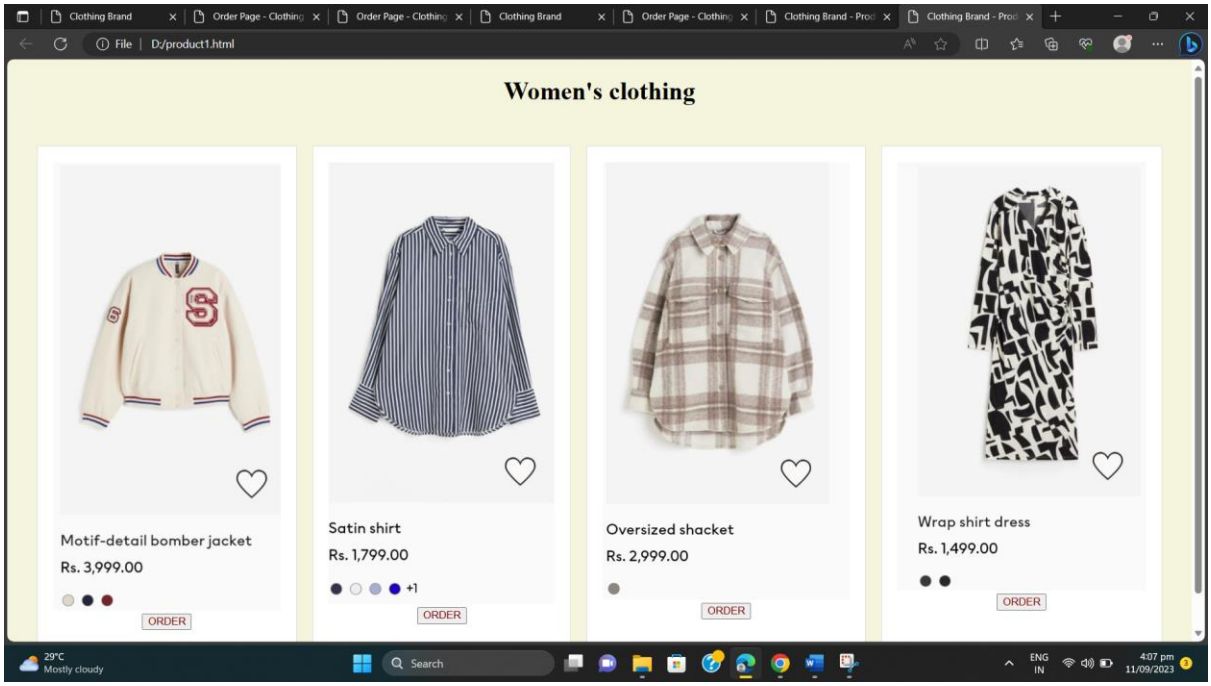
CHAPTER 5: RESULTS AND DISCUSSION

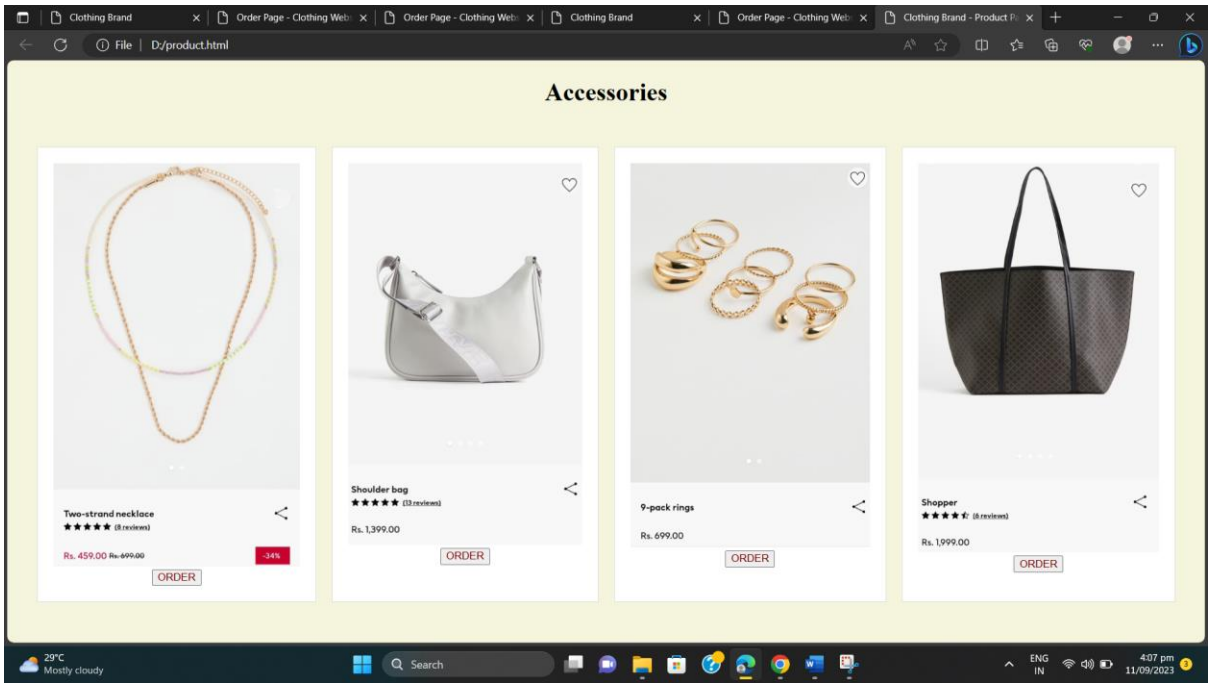


COVER PAGE:

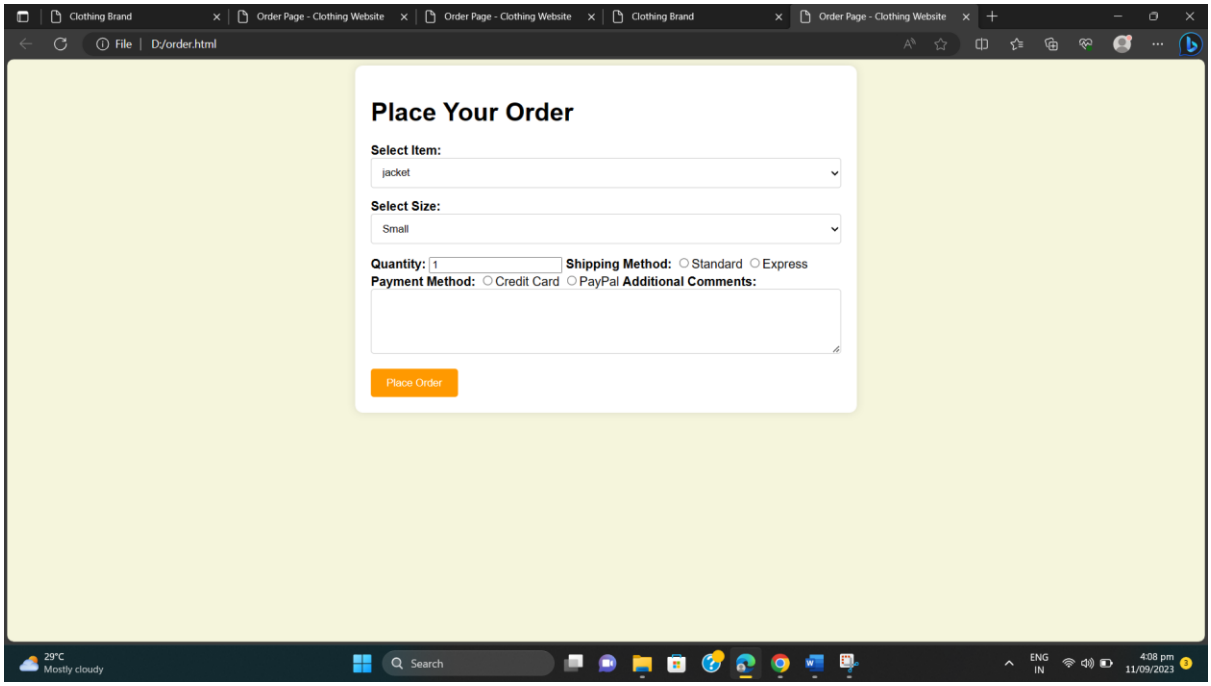
CATEGORY PAGE:

PRODUCT PAGES:





ORDER PAGE


Clothing Brand
Order Page - Clothing Website
Order Page - Clothing Website
Clothing Brand
Order Page - Clothing Website
File | D:/order.html
Place Your Order
Select Item:
jacket
Select Size:
Small
Quantity: 1
Shipping Method: Standard Express
Payment Method: Credit Card PayPal
Additional Comments:
Place Order
29°C
Mostly cloudy
Search
ENG
IN
4:08 pm
11/09/2023

CHAPTER 6: CONCLUSION AND REFERENCES

Fashion is important because it reflects the culture of a country. It makes our life colourful and changes our life with time. In a way it also adds variety to life, providing an opportunity in trying out something new.

In conclusion, fast fashion is a positive phenomenon that has changed the fashion industry for the bestt. It promotes a distorted consumer culture and encourages manufacturers to produce clothing of low-quality. Also, fast fashion has diminished the special status of luxury brands.

REFERENCES

- Bootstrap, ”Build fast, responsive sites with Bootstrap,”Bootstrap, [Online]. Available: https://getbootstrap.com
- https://www.sciencedirect.com/science/article/pii/
- https://medium.com/workfall/how-to-create-ci-cdworkflow-using-aws-codestar-fa25ca2ee091
- https://aws.amazon.com/getting-started/hands-on/createcontinuous-delivery-pipeline/module-one/
- Stahl, D., Bosch, J. (2017). Cinders: The continuous integration and delivery architecture framework. Information and Software Technology, 83, 76-93.