

S.no	observation	Marks	Signature
1	Personal webPage 25/8/24		d/b/24
2	clickable image link 25/8/24		
3	Navigation Bar 26/8/24		
4	Student slides 26/8/24		
5	Product information table 27/8/24		
6	Employee list 27/8/24		
7	Navigation 28/8/24		
8	Document with folder 28/8/24		
9	three column layout 29/8/24		
10	my resume 29/8/24		
11	Promotion Button 30/8/24		
12	welcome to website 30/8/24		

Observation-2

Scenario-1:

You're building a one-page online resume for yourself. This page is standalone, meaning it won't be linked to any other pages on the web. You want to style it with a specific color scheme, font family, and layout, but you want to keep everything within HTML.

Question:- How would you apply styles to this resume directly within the HTML file.

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>my Resume </title>
```

```
<style>
```

```
  body {
```

```
    font-family: Arial, sans-serif;
```

```
    color: #333;
```

```
    background-color: #f0f0f0;
```

```
    margin: 0;
```

```
    padding: 0;
```

```
  {
```

```
    container {
```

```
      width: 80%;
```

```
      margin: auto;
```

```
      background: #fff;
```

```
      padding: 20px;
```

```
      border-radius: 8px;
```



```

<h2> Experience </h2>
<p> Previous roles & accomplishments in web development &
design </p>
</div>
<div class="Footer">
<p> contact : Deekshitha deekshitha@gmail.com </p>
</div>
</div>
<body>
</html>

```

Output:-

My Resume.

Deekshitha - web developer/designer

G passionate web developer with experience.

Experience Previous roles & accomplishments.

Skills:- HTML, CSS.

contact:- Deekshitha@gmail.com.

Scenario 2:-

You are tasked with making quick adjust to a web page that already has a fully developed external style sheet. The marketing team requests that a single button on the home page be highlighted with different background color.

Question:-

How would you apply a specific background color to this button without an external stylesheet.

box-shadow: 0 0 0 0;

{ hi, ho, hie

color: #0070ff;

{ p {

line-height: 1.6;

h1 {

text-align: center;

margin-bottom: 20px;

{

section {

margin-bottom: 20px;

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<h1>maera </h1>

<p>web developer / designer </p>

</div>

<div class="section">

<h2>about me </h2>

<p>passionate web developer with experience</p>

</div>

<div class="section">

<h2>skills </h2>

HTML, CSS, Java Script

</div>

<div class="section">


```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Promotional Button</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Welcome to our website!</h1>
```

```
  <p>Check out our latest Promotional offer by clicking  
  the button below.</p>
```

```
  <button style="background-color: #ff5733;  
  color: white;
```

```
  padding: 10px 50px;
```

```
  border: none;
```

```
  border-radius: 5px;
```

```
  cursor: pointer;>
```

```
    Promotional Button
```

```
</button>
```

```
</body>
```

```
</html>
```

Output:-

Promotional Button:-

Welcome to our website.

check out our latest Promotional offer by
clicking the button below

Promotional Button

CSS code :-

root {

--primary-color: #3498db;

--secondary-color: #2ecc71;

--text-color: #333;

--background-color: #f6f6f6;

--font-Primary: Roboto, sans-serif;

}

body {

font-family: var(--font-Primary);

background-color: var(--background-color);

color: var(--text-color);

margin: 0;

padding: 20px;

}

header {

text-align: center;

padding: 20px;

background-color: var(--primary-color);

color: white;

}

header h1

font-family: var(--font-Secondary);

font-size: 2.5em;

margin: 0;

}

header p

font-size: 1.2em;

}

section {

margin: 20px 0;

}

Scenario-3

You are working on a large website with multiple pages, all of which need to have a consistent design. The site already uses an external css file styling.

Question:- How would you efficiently apply new styles across all pages of website.

HTML code

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title> my website </title>
```

```
<link rel="stylesheet" href="styles.css">
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <h1> welcome to my website </h1>
```

```
  </header>
```

```
  <div class="container">
```

```
    <p> this is a sample paragraph on homepage </p>
```

```
    <button> click me </button>
```

```
  </div>
```

```
</body>
```

```
</html>
```

button {

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

}

• btn-Primary {

background-color: var(--Primary-color);

color: white;

}

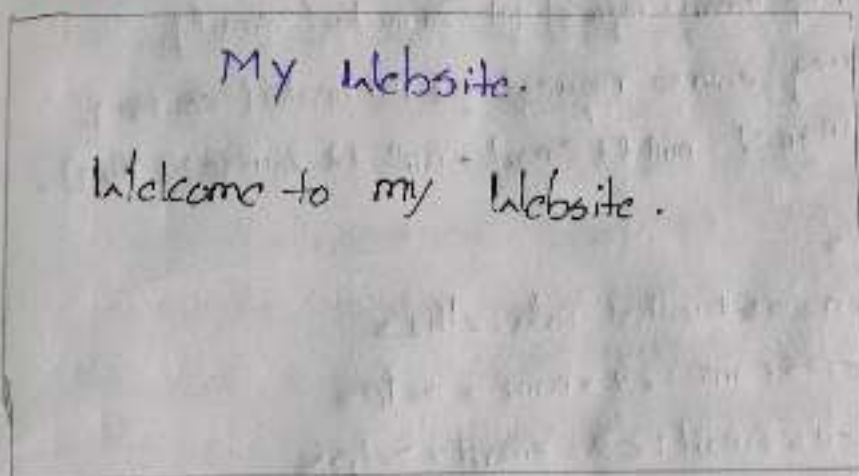
• btn-Secondary {

background-color: var(--Secondary-color);

color: white;

}

card:-



Scenario-4:- A university portal requires a JSP Page for students to register for Workshops by submitting their name, Email and course. Submitted information should display on the same Page.

Output:-

Submitted Data:

Name: Alice.

Email: alice@gmail.com.

Course: Java Programming.

Scenario-5:-

A library maintains its book catalog in XML format, requiring an XSD schema with specific constraints (ISBN must be 13 digits, Publication year is 4 digits, and availability is either "Available" or "checked out").

XML Document:-

```
<catalog>
```

```
  <book>
```

```
    <title>Effective Java</title>
```

```
    <author>Joshua Bloch</author>
```

```
    <isbn>9781034685991</isbn>
```

```
    <publication year>2018</publication year>
```

```
    <genre>Programming</genre>
```

```
    <availability>Available</availability>
```

```
  </book>
```

```
</catalog>
```

XSD Schema:-

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <xsd:element name="catalog">
```

```
    <xsd:complexType>
```

```
      <xsd:sequence>
```

```
        <xsd:element name="book" maxOccurs="unbounded">
```

Question:- Write a JSP Page that includes HTML.

JSP code:-

```
<%@ Page import="java.io.*", javax.servlet.*, javax.servlet.http.*"%>
```

```
<html>
```

```
<body>
```

```
    <form method="get">
```

```
        name: <input type="text" name="name"> <br>
```

```
        Email: <input type="text" name="Email"> <br>
```

```
        course: <input type="text" name="course"> <br>
```

```
        <input type="submit" value="Register">
```

```
    </form>
```

```
<%
```

```
    String name = request.getParameter("name");
```

```
    String Email = request.getParameter("Email");
```

```
    String course = request.getParameter("course");
```

```
    if (name != null && Email != null && course != null) {
```

```
    {
```

```
        <%
```

```
            <h2>Submitted Data: </h2>
```

```
            <p>Name: <% = name %> </p>
```

```
            <p>Email: <% = Email %> </p>
```

```
            <p>Course: <% = course %> </p>
```

```
        <%
```

```
    }
```

```
    <%
```

```
</body>
```

```
</html>
```


<x5:complexType>

<x5:sequence>

<x5:element name="title" type="x5:string"/>

<x5:element name="author" type="x5:string"/>

<x5:element name="isbn" type="x5:string" pattern="^\d{13}\$"/>

<x5:element name="publicationYear"

type="x5:int" minInclusive="100"

maxInclusive="9999"/>

<x5:element name="genre" type="x5:string"/>

<x5:element name="availability" type="x5:string"/>

<x5:simpleType>

<x5:restriction base="x5:string">

<x5:enumeration value="available"/>

<x5:enumeration value="checked"/>

<x5:restrictions>

<x5:simpleType>

<x5:elements>

<x5:sequence>

<x5:complexType>

<x5:elements>

<x5:sequence>

<x5:complexType>

<x5:elements>

<x5:schemas>

Output:-

Library Catalog:

Book Title: Effective Java

Author: Vishnu

ISBN: 982013425991

Publication year: 2018

Genre: Programming

Availability: Available

Scenario-6:- An online news website needs to display articles stored in XML format in a browser-friendly HTML format. Each article includes a title, author, date, summary and content.

XML Data:-

<articles>

<articles>

<title> Breaking news </title>

<author> Vishnu </author>

<date> 2024-09-11 </date>

<summary> summary of this article </summary>

<content> full content of the article </content>

</articles>

</articles>

XSLT code:-

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/xsl"

transform="version="1.0"

<xsl:template match="/articles">

<html>

<body>

<xsl:for-each select="article">

<h2 xsl:select="value-of select='title'/><xsl:

value-of select='date'/></h2>

<p xsl:select="value-of select='author'/></p>

<p xsl:select="value-of select='summary'/></p>

<p xsl:select="value-of select='content'/></p>

</xsl:for-each>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

Output:-

Breaking news - 2024-09-11

By: Vishnu

Summary: (This is the) summary of the article

Content: (This is the) full content of the article

Providing detailed information.

Scenario 3:- Write an XSLT file to transform the XML data into an HTML format.

XML Sample Data:-

<articles>

<article>

<title>Technology Advancements</title>

<author>Vishnu</author>

<date>2024-09-12</date>

<summary> Explores the latest trends in technology

summary

<content> From AI to blockchain, technology is evolving

</content>

<article>

</article>

XSTT code:-

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999-

xsl:transform" version="1.0">

<xsl:template match="articles">

<html>

<body>

<xsl:for-each select="article">

<h2><xsl:value-of select="title"/>

<xsl:value-of select="data"/>

</h2>

<p><xsl:value-of select="content"/>

</xsl:for-each>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

Output:-

Technology Advancements - 2024-09-12

By: Vishnu

Summary: Explores the latest trends in technology

Content: From AI to blockchain, technology is evolving.

Capstone Project

Online Laptop Sales & Services

Creating Websites - for online laptop sales & services, the number & types of pages vary based on the depth & content. Here's an outline of various webpages on e-commerce.

1. Home Page:-

- Overview of online store
- Highlights of featured laptops, special offers & services
- Navigation to other sections of the site.

2. Product Categories:

- **Laptop by Type:** Business, Gaming, ultrabooks etc.
- **Laptop by Brand:** Dell, HP, Apple, Lenovo etc.
- **Accessories** : chargers, bags, mouse, keyboard etc.
- **Custom Builds** : options for configuring laptops with specific components

3. Product Pages:-

- Detailed descriptions of each laptop model
- specifications, images and customer reviews
- options for customization or upgrades
- comparison tools between different models.

4. Services:-

- **Repair Services:** Details on laptop repair offerings
- **Warranty Services:** Information on warranty options and claims
- **OS Installation:** OS installation, antivirus etc.
- **Data Recovery** : services for recovering lost data.
- **Trade in Program:** Details on trading in old laptop.

5. Special offers:-

- **Discounts**: current promotions and discount codes
- **Bundles**: Deals on laptop and a necessary bundles
- **Seasonal offers**: special offers during holiday seasons

6. Customer services:-

- **F.A.Q.**: commonly asked questions regarding sales, shipping services
- **contact us**: -forms -for customer inquiries, live chat support
- **service Request form**: online form to request repairs or support

7. About:-

- **company history**: Information about business, mission & values
- **Team members**: Profiles of key team members
- **careers**: job openings and application forms

8. News:-

- **Tech News**: latest updates in the laptop & tech world
- **Buying guides**: Articles to help customers choose right laptop
- **How-to Articles**: Tutorials on maintaining & optimizing laptops

9. Reviews:-

- **customer reviews**: feedback from previous customers
- **case studies**: Detailed stories of how your services helped customers

10. Cart & Checkout:-

- **shopping cart**: overview of selected products and services
- **checkout process**: Payment and shipping information, confirmation page.

11. Account Pages:-

- **login**: user account creation and management
- **order history**: Details on Previous Purchases & order tracking
- **Wish list**: A list of Products the user is interested in buying later.

12. Shipping & Returns:-

- **Shipping Info**: Details on how delivery options and costs.
- **Returns Policy**: Guidelines on how to return Products and get refunds.

13. Legal Pages:-

- **Privacy Policy**: How Customer data is collected & used
- **Terms and conditions**: Rules & Regulations for using the site
- **Warranty Information**: Detailed terms of warranty coverage.

14. Affiliate Program:-

- **Join Program**: Information on how to become an affiliate.
- **Affiliate Dashboard**: Tools for affiliates to track their earnings.

15. Site map:-

- An overview of all Pages available on the site for easy navigation.

16. Location & Storefinder:-

- **find a store**: Information on physical store locations, if any.
- **Service centers**: List of service centers for repairs & support.

17. Events:-

- **Webinars**: Information on upcoming tech webinars or workshops.
- **Promotional Events**: Announcements about sales events or tech expos.

LAPTOP SALES AND SERVICES.

login Page.

user name:

~~Pass~~
Email/Phone:

Phone:

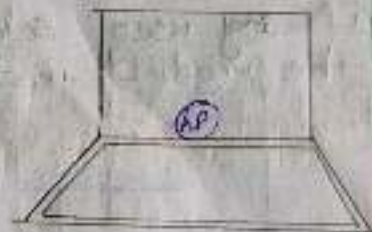
Password:

forget Password?

Don't have create new one?

Home Page :-

LAPTOP SALES



summer sales.
offers Available
up to 20%/-



Home



You



more



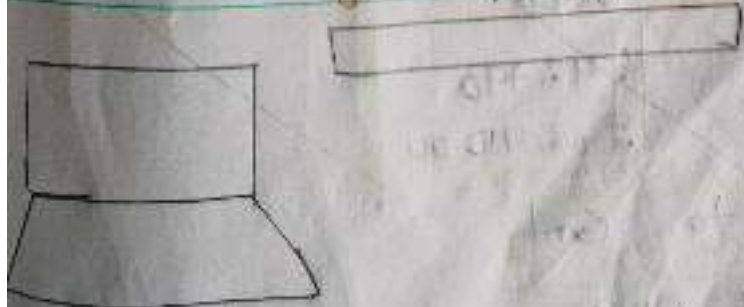
cart



menu

If user selects a Particular Product. Just for Example

Product Categories Features :-



18. Loyalty Program:

- **Membership Details:** Benefits of joining the loyalty Program
- **Points Redemption:** How to redeem Points for discounts/rewards

19. Community Forum:-

- A platform for customers to discuss Products, share tips, and get support from other users.

20. Partner Pages:-

- **Business Partnership:** Information for potential business Partners.
- **Reseller Program:** Details for those interested in reselling your laptops.

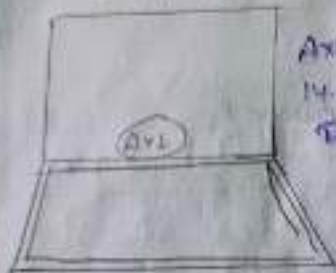
Product categories:



Dell 15 Thin light
laptop with Gen
Intel core i5
8GB, 512GB SSD
₹ 46,790

HP Laptop 250 G8
Gen Intel core i3,
15.6 inch (39.6 cm),
8GB, DDR4, 512GB SSD
₹ 38,790

Lenovo ThinkPad
E14 AMD Ryzen
(35.56 cm) 14 inch
IPS, 300 nits, Thin and
light (1.5 kg) 8GB
₹ 47,500

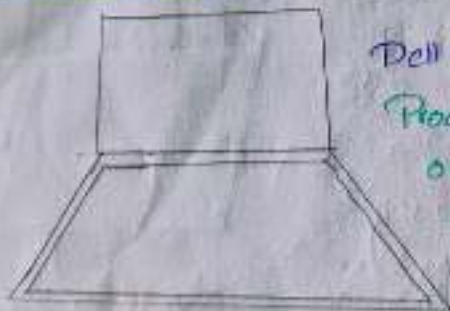


Axiova Book laptop
14.1 inch FHD IPS
Display
(4GB RAM, 128GB SSD)
₹ 12,797

Business laptops
Gaming laptops
Student laptops

If any user select a Particular Product, then how
the Product features and Qualities displays

Product Features:-



Dell 15 Thin + light laptop

Processor: 13th Gen Intel core i3-13500

OS: Windows 11

Color: Grey

Graphic Card: - FHD 120Hz 250 nits

Memory: - 8GB DDR4 + 512GB SSD

Display: - 15.6" (39.62 cm)

- 27% ₹ 36,490

MRP ₹ 49,935

Qty: - 1

Black.

₹ 48,790

₹ 66,410.00

Add to Cart

Buy Now

About a Product:



Dell Inspiron 3530 Thin Light
laptop, 9th Gen Intel Core i3-
13050 / 8GB / 512GB SSD
15.6" (39.62cm)
Windows 11

-231 240,990

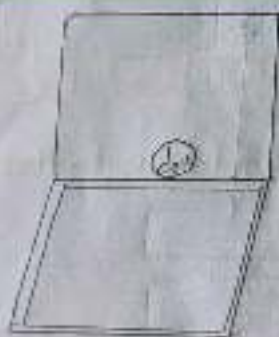
M.R.P - 243,000

3.8 ★★★★★ ♡ FREE delivery

Add to Cart

Any updates about the laptops and their series
in the mark will updates here:

updates about the product:



dell i3, 9th Gen Intel Core

i3-13050

8GB, 512GB SSD

Windows 11

-241 40,990

M.R.P = 253,658

Measurements

You might also like v
Sponsored.



8GB
512GB
SSD

dell 13th Gen
i3-13050



dell 13th Gen
i3-12150



Acer 13th
Gen i3.



Lenovo 13th Gen
Intel Core.

Service Page



① All offers & discounts

② Exchange your old Product
Save upto ₹ 14,000

③ Add Accessories

④ Protect your laptop

- ₹ 21,920

M.R.P. ₹ 25,255

EMI - from ₹ 21,924, No Cost EMI available

Start using Amazon Pay up!

- free Delivery

1 year warranty Care

Soft & Hard

Delivered.

customer Service:-

Any FAQs About Product:

customer

Contact num:-

Service_num:- xx xx xx 9921

- feedback:-

Service Request form:-

customer name:-

Contact num:-

Email Id :-

Problem of Product:-

Service help line num:-

Request form:-

Reviews :-



overall camera Battery Display value for money

4.6 ★

5

★★★★★

4

★★★★★

3

★★★

2

★★

1

★

Reviews

Reviewed

R₁

R₂

Product ID

P₁

P₂

Most helpful

SSD

25

Rating

3

4

credit AT

25th May 2018

4th June 2019

Cart and checkout Payments



Payments :-

Banking

Gift Card

Cash on delivery

Wallet

Coupons apply (20%)

Credit Card

debit Card

Credit Card details

Account

Bank name

Account Pages :-



Hello, Deekshitha ✓

Your orders

Your account

♥ Wishlist Items

Buy Again

Your lists

Order History orders

Legal Pages

Terms and Conditions :- eligibility
Product and service
ordering and payment
shipping and delivery
Returns and Refunds
warranty
Contact Information

Privacy Policy :-

warranty Policy :-

Shipping Policy :-

Disclaimer :- Accessibility statement :-

cookie Policy :-

contact us Page :-

Site Map:



Shop → categories
special offers

Services → Repair services
up grade services

About us

contact us

Home



you



more



cart



menu



Shipping orders below:-

one-time Password required at time of delivery

Shipping to:- Deekshitha, 1st E-106, Chennai, PopCity.

Items :- ₹ 240,990.00

Delivery :- ₹ 40.00

Total :- ₹ 41,030.00

Promotion Applied:- ₹ 40,00

Order total :- ₹ 40,990.00

Your savings:- ₹ 12,708.00 (24%)

- FREE Delivery
- Item discount

Paying with phonePe.

Valid Gift Card or Promo Code

Shipping order details

Select a delivery address:- Deekshitha, 1st E-Block-106,
Chennai, PopCity near
Suresh land

Deliver to this address

Edit Address

Add delivery instructions

Find a Pickup location near you.

Get it by :- 19 Sept Free delivery.



Deekshitha 13th Cross
13-13050, 86B, 5th Cross

Quantity:- 1 v.

Place your order and pay

Return and Refund

1. Overview.

2. Eligibility for Returns ✓

Time frame

conditions of items

Proof of purchase

Non-Returnable Items...

3. Return Process -

1. Contact Customer Support:-

Email:- _____

Phone num:- _____

Address:- _____

They will ship the Package and Return Confirmation

→ Refund Process.

Refund Method:-

Refund Timeframe

Partial Refunds

Shipping Costs.

→ Exchange Policy.

Eligibility for Exchange:- Exchanges are allowed within 30 day of the Purchase date for item.

Contact num:-

Email:-

Address:-



SET -2

1	You are updating a legacy web application to take advantage of modern web standards. The project involves transitioning from an earlier HTML version to HTML5. The goal is to leverage HTML5's new features to improve web development practices and enhance the functionality of web forms.	10	CO1
2	Designing a user registration form for a new web application. The form needs to capture essential information such as name, email, and a message from users. It is crucial to implement client-side validation to ensure that the data entered is accurate and complete before the form is submitted.	10	CO1
3	Designing a website that needs to function effectively across various devices and screen sizes. The design should include different types of layouts such as fixed, fluid, and responsive. To achieve this, you need to use CSS techniques like Flexbox or Grid to ensure that the layout adapts well to different screen sizes and maintains a consistent user experience.	10	CO1
	To create a webpage that offers a seamless user experience across devices, responsive design principles must be applied. This involves using fluid grids, flexible images, and media queries to ensure the layout adjusts gracefully to different screen sizes. For instance, on a desktop, the webpage might display multiple columns with detailed content, while on a smartphone, the same content could stack vertically for easy scrolling. Navigation menus should transform into a hamburger icon on smaller screens, ensuring they remain accessible without taking up too much space. Additionally, touch-friendly elements and optimized images ensure fast loading times and a smooth experience on all devices.	10	CO1
5	Given a scenario (e.g., creating a blog post, a product listing), design a webpage using appropriate elements, tables, lists, and images for optimal readability and user experience. Also, describe step-by-step the sequence of HTTP requests and responses that occur when a user accesses a webpage containing multiple resources (HTML, CSS, JavaScript, images)	10	CO

Assignment - 1

19/02/2023

You are updating a legacy web application to take advantage of Modern web standards. The Project involves transitioning from an earlier HTML version to HTML5. The goal is to leverage HTML5 new version improve web development Practices and Enhance the functionality of web forms.

HTML:-

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title> legacy web application </title>
```

```
<meta http-equiv="content-type" content="text/html; charset="
```

```
"UTF-8"/>
```

```
<style type="text/css">
```

```
• header & background- colour: #f0f0f0; padding: 10px; }
```

```
• content & padding: 10px; }
```

```
• footer & background- colour: #f0f0f0; padding: 10px; text-align: center; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="header">
```

```
<h1> welcome to our website </h1>
```

```
</div>
```

```
</div>
```

```
<div class="content">
```



```

<form action="/submit" method="post">
  <label> name: </label>
  <input type="text" name="name"/> </input>
  <label> Email: </label>
  </header>
  <main>
    <form action="/submit" method="post">
      <label for="name"> name: </label>
      <label for="birthdate"> Birthdate: </label>
      <input type="text" id="birthdate" name="birthdate"/>
      <input type="submit" value="submit"/>
    </form>
  </main>
  <footer>
    </footer>
  </body>
</html>

```

2. Designing a user registration form for a new web application the form needs to capture essential information such as name email and a message from users. It is crucial to implement client side validation to ensure that the layout adapts well to complete before form the submit:-

HTML:-

```

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

```

margin-bottom: 15px;

border-radius: 4px;

border-size: 1px;

in id type="submit"/>

background-color: #cccccc;

color: white;

border: none;

cursor: pointer;

transition: background-color 3s ease;

}

</style>

</head>

<body>

<form action="" submit method="post" novalidate>

<h2>user Registration</h2>

<label for="name">name</label>

<input type="email" id="email" name="email" required

placeholder="example@example.com">

minlength="10" max-length="500"

placeholder="Enter your message here">

<input type="submit" value="Register">

</form>

</script>

</body>

</html>


```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width
initial-scale=1.0">
<title>user Registration </title>
<style>
```

```
body {
```

```
font-family: crial, sans-serif;
```

```
margin: 0;
```

```
padding: 20px;
```

```
background-color: #f0f0f0;
```

```
} from {
```

```
background-color: #fff;
```

```
padding: 20px;
```

```
border-radius: 5px;
```

```
max-width: 400px;
```

```
margin: 0 auto;
```

```
box-shadow: 0 0 10px rgba(0,0,0,0.1);
```

```
} label {
```

```
display: block;
```

```
margin-bottom: 5px;
```

```
font-weight: bold;
```

```
} input, textarea {
```

```
width: 100%;
```

```
padding: 10px;
```

web
trial
from
valida
amp/

UP?

outPut:-

Registration form

Name : _____

Email : _____

Message : _____

- 3) Designing a Website that needs to function efficiently across various devices and screen sizes. The design should include different types of layouts such as fixed, fluid, and responsive. To achieve this, you need to use CSS techniques like flexbox or Grid to ensure the layout adapts well to different screen sizes and maintains a consistent user experience.

Implementation using CSS:-

1. Fixed Layout:-

- Define a container with a fixed width, typically using 'px'.
- Use CSS Properties like 'width', 'height', and 'margin' to align and size elements within the fixed width.
- Fixed container:

```
width: 1200px;  
margin: 0 auto;  
{
```

Fluid layout:-

- Use Percentage-based widths for elements, ensuring they expand or contract relative to their Parent Container.
- Fluid container:

```
width: 90%;  
margin: 0 auto;  
{
```

3. Responsive layout with flexbox.

* utilize flexbox to create a flexible grid system that adjust based on screen size.

- flex - container {
display: flex;
flex-wrap: wrap;

output:-

fixed layout

fixed layout

- flex - item {
flex: 1 200px 1

@media (max-width: 768px) {

- flex - item {

flex: 1 100%;

Responsive 1

Responsive 2

Responsive 3

Responsive 4

4. Responsive layout with Grid.

* use CSS grid to create a more complex layout that can adjust to different screen sizes.

- grid - container {
display: grid;
grid-template-columns: repeat(3, 1fr);
gap: 20px;

@media (max-width: 768px) {

- grid - container {
grid-template-columns: 1fr;

Assignment 2

4. To create a webpage that offers a seamless user experience across devices, responsive design principles must be applied. For instance, on a desktop, the webpage might display multiple columns with desktop content, while on a smartphone, the same content could stack vertically for easy scrolling. Additionally, touch-friendly elements and optimized images ensure fast loading times and a smooth experience on all devices.

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Responsive Webpage</title>
```

```
  <style>
```

```
    .container {
```

```
      display: grid;
```

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

```
      gap: 20px;
```

```
      max-width: 1200px;
```

```
      margin: auto;
```

```
    }
```

```
    .item {
```

```
      padding: 20px;
```

```
      background-color: #f0f0f0;
```

img {

max-width: 100%;

height: auto;

}

@media (max-width: 768px)

{

grid-template-columns: 1fr;

}
nav ul {

display: none;

}

nav.hamburger > > <div>

</nav>

<div class="container">

<div class="item">content Block 1</div>

<div class="item">content Block 2</div>

<div class="item">content Block 3</div>

</div>

</body>

</html>

Output:-

Responsive webpage

Home

about

Service

contact

column?

content goes here.

5. Given a scenario, design a webpage using appropriate elements, tables, lists, and images for optimal readability and user experience. Also, describe step-by-step sequence of HTTP requests and responses that occur when a user accesses a webpage containing multiple resources (HTML, CSS, JavaScript, Images).

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title> Blog Post with Product Listings </title>
```

```
<style>
```

```
body {
```

```
font-family: Arial, sans-serif;
```

```
line-height: 1.6;
```

```
{  
h1 {
```

```
color: #333;
```

```
{
```

```
content {
```

```
width: 70%;
```

```
max-width: 800px;
```

```
margin: 20px auto;
```

product-list {

list-style-type: none;

padding: 0;

}
product-list li {

border: 1px solid #ccc;

margin: 10px 0;

padding: 10px;

display: -flex;

align-items: center;

}
product-list img {

max-width: 150px;

margin-right: 20px;

}

</style>

</head>

<body>

<div class="content">

<h1>my latest blog post</h1>

<p>welcome to my blog! here are some of the
products I've been using lately:</p>

<ul class="product-list">

Output:

my awesome blog

• Home

• About

• Blog

• Contact

<h2>Product 1</h2>

<p>A great Product that I highly recommend</p>

</div>

<h2>

</h2>

</div>

<h2>Product 2</h2>

<p>This Product has changed my life</p>

</div>

</h1>

</div>

</body>

</html>

The title of my blog Post

Published on Aug 27, 2024 by author name a
disruptive ~~image~~ related to the blog post this
is the introduction to my blog Post

Subheading is:

Here is some detailed content.

subheading 2

more detailed about content.

comparison table:

feature	option A	option B
Price	\$100	\$150
Performance	Good	excellent
Support	24/7	Business hours

subheading 3:

- keypoint 1
- keypoint 2.
- keypoint 3.

conclusion! summarize the Blog Post

@2024 my awesome Blog. All rights reserved

Rubrics	split up	Marks obtained	Total marks
Form Design			
version design			
features design			
form design			
validation			
layout look			
layout design			
css background			
form design			
layout design			
navigation			
form design			
css			
layout design			
Java script			



Course Code /Title: CSA4399 – Internet Programming
Programme : Computer Science and Engineering

ASSIGNMENT 3 QUESTIONS

SET -5

S.No	Questions	Marks	CO	BTL
1	Implementing a feature in a web application that tracks the number of accesses by a client within a single session. You need to use Java Servlets to manage and monitor session data. The application should count how many times the client accesses the application during their session and retrieve information about the session, such as the session ID, creation time, and last accessed time.	20	CO4	3
2	Write a scenario where you had to use JSTL to solve a complex problem and how you went about it. Also, elaborate the function library in JSTL and how to create custom functions.	20	CO4	2
3	A page of stock market quotes uses script to refresh the page every five minutes in order to ensure the latest statistics remain available. 20 seconds before the five minute period expires, a confirm dialog appears asking if the user needs more time before the page refreshes. This allows the user to be aware of the impending refresh and to avoid it if desired.	20	CO5	3
4	You are developing an e-commerce application that needs to integrate with an external payment gateway service. This service is described using a WSDL file. How would you use the WSDL file to integrate the payment gateway service into your e-commerce application? Describe the steps involved in generating the client code, invoking the service, and handling any potential errors	20	CO6	3

Assignment-3

- ① Implementing a feature in a web application that tracks the number of accesses by a client within a single session using Java servlets using HTTP session object to manage and monitor session data.

servlet code:-

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

@WebServlet("/sessionTracker")

public class sessionTracker extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws

ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession(true);

String sessionId = session.getId();

long creationTime = session.getCreationTime();

long lastAccessedTime = session.getLast

AccessedTime();

Integer visitCount = (Integer) session.getAttribute("visitCount");

if (visitCount == null) {

```

Visit Count ++;
session.setAttribute("visit count", visitCount);
out.println("<html><body>");
out.println("<html>session Tracking Example</html>");
out.println("<p>sessionID: " + sessionId + "</p>");
out.println("<p>session created: " + new Date().toString() + "</p>");
out.println("<p>last accessed: " + new Date().toString() + "</p>");
out.println("<p>no of accesses in this session: visit count + "</p>");
out.println("</body></html>");
}
}

```

① Output:-

```

Session Tracking Example
Session ID: 12345ABCDE
session created: Mon Sep 09 12:00:00 IST 2024
last accessed: Mon Sep 09 12:01:05 IST 2024
no of accesses in this session: 1

```

- ② Write a scenario where you had to use JSTL to solve a complex problem and how you went about it. Also, elaborate the function library in JSTL and how to create custom functions.

JSP code using JSPC

```
<!--@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" -->
<!--@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" -->
```

```
<!--@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" -->
```

```
<html>
```

```
<head>
```

```
<title>order management</title>
```

```
</head>
```

```
<body>
```

```
<h2>order list</h2>
```

```
<form method="GET" action="order.jsp">
```

```
<label for="status" id="status">
```

```
<select name="status">filter by status</select>
```

```
<option value="All">All</option>
```

```
<option value="delivered">Delivered</option>
```

```
<option value="pending">pending</option>
```

```
</select>
```

```
<input type="submit" value="Filter">
```

```
</form>
```

```
<table border="1">
```

```
<thead>
```

```
<tr>
```

```
<th>order ID</th>
```

```
<th>date</th>
```

```
<th>status</th>
```

```
<th>Amount</th>
```

```
</tr>
```

```
</thead>
```

```
</table>
```

<short-name> <custom> <short-name>

<url> http://example.com/custom/short-name

<function>

example: reverse string class

<function> - class custom example - custom function the

- then class

<function> signature: java.lang.String

reverseString (java.lang.String)

<function> signature

<function>

<html>

Output:-

Custom function Example

Original: HelloWorld

Reversed: dlrow olleH

- ③ To implement the described functionality for refreshing a stock market quotes page every five minutes, with a confirmation dialog appearing 20 seconds before the refresh.

Javascript code snippet:

<!DOCTYPE html>

<html lang="en">

<head>

meta charset="UTF-8">

<title> Stock market quotes </title>

<script>

function refreshPage() {
location.reload();


```
<c:forEach var="order" items="${orders}">
```

```
<c:choose>
```

```
<c:when test="${param.status} == 'pending'">
```

```
<td>${order.status} </td>
```

```
</c:when>
```

```
<td>${order.id} </td>
```

```
<td>${order.date} </td>
```

```
<td>${order.status} </td>
```

```
<td>${order.amount} </td>
```

```
</td>
```

```
</c:when>
```

```
<c:choose>
```

```
<c:forEach>
```

```
<c:forEach>
```

```
</body>
```

```
</table>
```

```
</body>
```

```
</html>
```

Output:-

Order list

| Order ID | Date | Status | Amount |
|----------|------------|---------|--------|
| 1002 | 2024-09-03 | Pending | 150.00 |
| 1003 | 2024-09-09 | Pending | 300.00 |

JSTL function Library (fn)

④ Creating Custom Functions in JSTL

```
<taglib xmlns="http://java.sun.com/xml/ns/jstl" version="1.0">
```

```
</taglib>
```

on Timeout (case)
const confirmRefresh = confirm("The Page will refresh
in 20 seconds");

if (!confirmRefresh)

refreshPage();

}; else

alert("Page refresh cancelled");

};

</script>

</body>

</html>

</body>

</html>

Output:-

Page Display:-

Stock Market Quotes

- Apple Inc (AAPL): \$1500.00

- Microsoft Corp. (MSFT): \$2500.00

- Alphabet Inc (GOOGL): \$2800.00

Confirmation Dialog

The Page will refresh in 20 seconds

OK

Cancel

Alert

Page refresh cancelled

OK

Page Refresh.

(The Page reloads, displaying updated content)

Stock Market Quotes:

- Apple Inc. (AAPL) = \$152.00
- Microsoft Corp. (MSFT) = \$250.00
- Alphabet Inc. (GOOGL) = \$2850.00

4. To integrate an external Payment gateway service into your e-commerce application using a WSDL file

① Generate client code from WSDL

wsimport -keep-s src-d bin-p com.example.Payment-
v1.0.0 http://

Example: com.PaymentGateway.wsdl

② Integrate Generated code into Application

- Include Generated code
- Configure service Endpoint

③ Invoke the Payment Service

• create service Instance

PaymentService service = new PaymentService();

PaymentPortType port = service.getPaymentPort();

• Invoke Methods:

PaymentResponse response = port.processPayment
(paymentRequest);

④ Handle Response and Errors

• check Response

if (response.isSuccessful())

 // Handle successful Payment

else {

 // Handle Payment Failure

• Exception Handling

try {

 PaymentResponse response = port.processPayment
 (paymentRequest);

} catch (SOAPFaultException e) {

 // Handle SOAP fault (e.g. invalid request)

} catch (WebServiceException e) {

 // Handle connectivity or configuration errors

output:-

successful Payment:-

Payment successful. Transaction ID: 1987453210.

Payment failure (e.g. Invalid Card Details):

Payment failed due to a SOAP fault: Invalid
credit card details.

SOAP fault (e.g. Invalid Request)

Payment failed due to a SOAP fault: Invalid
Request format.

connectivity. Issue (eg: service unavailable)

Payment failed due to connectivity issues.

Could not connect to the Payment gateway.

Assignment Rubrics	splitup	Marks obtained	Total marks
Question-1	Code implemented - action ¹⁰ Session Data Accuracy ⁵ Efficiency and clarity ⁵ Explanation ⁵		
Question-2	Scenario Explanation - action ¹⁰ Function Library Explanation ⁵ Custom Functions ⁵ Clarity and organization ⁵		
Question-3	Script functionality ⁵ Dialog ⁵ User interaction ⁵ Design refresh ⁵ Code efficiency ⁵ Explanation ⁵		
Question-4	Node.js script ⁵ Functions ⁵ Understanding of SQL ⁵ Client code generation ⁵ Error handling ⁵ Clarity and depth ⁵		



Course Code /Title: CSA4399 – Internet Programming
Programme : Computer Science and Engineering

ASSIGNMENT 4 QUESTIONS

S.No	Questions	Marks	CO	BTI
1	From a developer's perspective, discuss why JDBC is essential in building database-driven applications. How to achieve JDBC CONNECTION pooling using JDBC Data Source and JNDI in Apache Tomcat Server. Provide examples of executing SQL queries using JDBC statements. Discuss the differences between Statement, Prepared Statement, and Callable Statement.	20	CO5	3
2	Describe the lifecycle phases of a JSP page. Explain the significance of each phase in the JSP execution process. Discuss the different ways to embed Java code within a JSP page with examples. Explain the advantages and disadvantages of using scriptlets, declarations, and expressions in JSP.	20	CO1	2
3	You need to develop a PHP program that generates a chessboard using HTML tables. The table should have a total width of 400px, and each cell should have a height and width of 30px. The chessboard should alternate colors between black and white for each cell to represent a typical chessboard layout. How would you write a PHP program using nested for loops to create a chessboard? The chessboard should be displayed using an HTML table with a total width of 400px, and each cell should have a height and width of 30px. Explain how you would use the nested for loops to alternate the cell colors and ensure the chessboard pattern is correctly displayed. Provide the code for this program	20	CO4	2
	You are developing a PHP application that reads content from a text file and uses regular expressions to extract specific patterns, such as email addresses and phone numbers. After extracting the data, the application should store the results in a new XML file following a defined schema for the data. Additionally, you need to compare and contrast DTD (Document Type Definition) and XML Schema for defining the XML structure. How would you create a PHP application that reads content from a text file and uses regular expressions to extract specific patterns (e.g.,	20	CO6	2



Assignment-4.

1. Why JDBC is essential in Building Database-Driven Applications.

JDBC is essential because it provides a standard API for Java applications to interact with databases.

Achieving JDBC Connection Pooling using JNDI Data source and JNDI in Apache Tomcat.

Configure DataSource

```
<Resource name = 'jdbc/myDB'
auth = 'Container'
type = 'javax.sql.DataSource'
maxTotal = '20'
maxIdle = '10'
maxWaitMillis = '10000'
username = 'dbuser'
password = 'dbPassword'
driverClassName = 'com.mysql.cj.jdbc.Driver'
url = 'jdbc:mysql://localhost:3306/mydatabase' />
```

Look up DataSource in Java code using Java.

```
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
import java.sql.Connection;

public class DatabaseUtil {
    public static Connection getConnection()
        throws Exception {
```

```

content initContent = new InitialContent();
DataSource ds = (DataSource) initContent;
return ds.getConnection();
}
}

```

Executing SQL queries using JDBC statement.

① Using a statement.

```

try (Connection conn = DatabaseUtil.getConnection();
    Statement stmt = conn.createStatement()) {
    String query = "SELECT * FROM users";
    Result set rs = stmt.executeQuery(query);
    while (rs.next()) {
        System.out.println("User ID: " + rs.getInt("id") + " +
        name: " + rs.getString("name"));
    }
}
}

```

② Using a Prepared statement.

```

try (Connection conn = DatabaseUtil.getConnection();
    PreparedStatement pstmt = conn.prepareStatement(
        "SELECT * FROM users WHERE id = ?")) {
    pstmt.setInt(1, 1);
    Result set rs = pstmt.executeQuery();
    while (rs.next()) {
        System.out.println("User ID: " + rs.getInt(
            "id") + " name: " + rs.getString("name"));
    }
}
}

```


3) using a callable statement - for stored Procedures

try (connection conn = DatabaseUtil.getConnection();
callableStatement cstmt = conn.prepareStatement("{
call getuserId(2)}")) {

• cstmt.executeQuery();

cstmt.close();

ResultSet rs = cstmt.executeQuery();

while (rs.next()) {

System.out.println("userId: " + rs.getInt(1) +
", name: " + rs.getString("name"));

}

Output:-

statement Example output:-

userId: 1, name: Vishnu

userId: 2, name: Priya

Prepared statement:-

userId: 1, name, Vishnu

callable statement:-

userId: 1, name Vishnu

8) Lifecycle Phases of a JSP Page.

① Translation Phase

② Compilation Phase

③ Initialization Phase

④ Request Processing Phase

⑤ Destruction Phase

① Disadvantages:- Can clutter JSP with Java code, leading to Poor separation of concerns.

Expressions:-

Advantages. Simplifies outputting dynamic content directly in JSP.

Disadvantages: limited to Expressions.

② Generates a chessboard using HTML tables with 400px (total) and each cell has a height and width of 30px.

PHP Code:-

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>chessboard </title>
```

```
<style>
```

```
table {
```

```
width: 400px;
```

```
border-collapse: collapse;
```

```
}  
td {
```

```
width: 30px;
```

```
height: 30px;
```

```
</style>
```

```
</head>
```

```
<body>
```


Embedding Java code in JSP

① Scriptlets

```
<% int sum = 5 + 10; %>
```

```
<%> The sum is: <%= sum %>
```

Output:

The sum is 15

② Declarations

```
<%: int add (int a, int b) { return a+b; } %>
```

```
<%> The result is : <%= add (3, 9) %>
```

Output:

The result is : 10

③ Expressions

```
<%> current time : <%= new java.util.Date().toString() %>
```

Output:-

current time : Mon Sep 09 18:30:00 PDT 2024

Scriptlets

Advantages:- Easy to use for embedding simple

Java logic

Disadvantages:- leads to messy code, difficult to maintain

Declarations

Advantages:- useful for declaring reusable methods and variables across multiple requests

```
<table>  
  <tr>  
    <td colspan=
```

folgt

echo' style = "background-color: black; color: white; text-align: center; padding: 5px; margin: 10px 0; font-weight: bold;">echo'

Echo "Litz"

522

 $\langle 1 \text{ table} \rangle$

body

 $\langle |h_{nm}| \rangle$

output :-

$\{ \} [+] [-] [+] [-]$

$$\{ \# \} [] \{ \# \} [] \{ \# \} [] \{ \# \} []$$

$[] [\#] [] [\#] [] [\#] [] [\#]$

$\{ \# \} [] [\#] [] [\#] [] [\#] []$

$$\{0\} \{1\} \{2\} \{3\} \{4\} \{5\} \{6\} \{7\} \{8\} \{9\}$$
$$[+][][\#][](+)[]\{ \# \}[]$$

output:

<data>

<emails>

<email> example@example.com </email>

<email> example@example.com </email>

</emails>

<phones>

<phone> +123-456-7890 </phone>

<phone> 987-654-3210 </phone>

</phones>

</data>

4.

PHP Application to Extract Data and Store in XML.

Steps:-

- ① Read Content from a Text file
- ② Extract Patterns using Regular
- ③ Create and store Results in an xml file
- ④ Define xml schema.

PHP Code:-

```
<?php
```

```
$filename = "input.txt";
```

```
$content = file_get_contents($filename);
```

```
$preg_match = all (/[a-zA-Z0-9-_.+~*]+@[a-zA-Z0-9-_.+~*]+/i, $content, $email(s));
```

```
$preg_match = all ([0-9-_.+~*]+@[a-zA-Z0-9-_.+~*]+/i, $content, $phone(s));
```

```
$xml = new SimpleXMLElement('data.xml', $options);
```

```
$emailElement = $xml->addChild('emails');
```

```
foreach ($email(s) as $email) {
```

```
    $emailElement->addChild('email', $email);
```

```
$phoneElement = $xml->addChild('phones');
```

```
foreach ($phone(s) as $phone) {
```

```
    $phoneElement->addChild('phone', $phone);
```

```
$xml->save('output.xml');
```

```
echo "Data extracted and saved to output.xml";
```


Assignment Rubrics	Split up	Marks obtained	Total Marks
Question 1	<p>30 Marks</p> <p>Explanation of Table ^{5M}</p> <p>Connection Pooling ^{5M}</p> <p>SQL Queries ^{5M}</p> <p>Indexing Types ^{5M}</p>		
Question 2	<p>Life Cycle Phases Expl- -nation ^{5M}</p> <p>Embedding Java code ^{5M}</p> <p>Advantages & Disadvan- -tages ^{5M}</p> <p>Clarity & Depth ^{5M}</p>		
Question 3	<p>Code Implementation ^{5M}</p> <p>HTML Table Structure ^{5M}</p> <p>Alternating colors ^{5M}</p> <p>Logic ^{5M}</p> <p>Explanation ^{5M}</p>		
Question 4	<p>Code Implementation ^{5M}</p> <p>- HTML ^{5M}</p> <p>Pattern Extraction ^{5M}</p> <p>XML File Generator ^{5M}</p> <p>DTD vs XML Schema comparison ^{5M}</p>		