

ADWT PRACTICAL

**** SLIP 1_Q1 :** Write the HTML5 code for generating the form as shown below. Apply the internal CSS to the following form to change the font size of the heading to 6pt and change the color to red and also change the background color to yellow.

==> <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Project Management Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: white;

margin: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

flex-direction: column;

}

h1 {

font-size: 6pt;

color: red;

text-align: center;

margin-bottom: 20px;

text-decoration: underline;

}

.form-container {

background-color: yellow;

padding: 20px;

border-radius: 8px;

width: 350px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

label {

font-weight: bold;

margin-top: 10px;

}

input, select, textarea, button {

width: 100%;

margin-top: 5px;

padding: 8px;

border-radius: 4px;

}

input, select, textarea {

border: 1px solid #ccc;

}

.form-row {

display: flex;

justify-content: space-between;

align-items: center;

margin-top: 10px;

}

.form-row label {

```

    flex: 1;
    margin-right: 10px;
}
.form-row input, .form-row select {
    flex: 2;
}
.radio-group {
    display: flex;
    justify-content: flex-start;
    align-items: center;
    margin-top: 5px;
}
.radio-group label {
    margin-right: 10px;
}
button {
    width: 48%;
    cursor: pointer;
}
.submit-btn {
    background-color: #4CAF50;
    color: white;
}
.clear-btn {
    background-color: #2196F3;
    color: white;
}
</style>
</head>
<body>
<h1>Project Management</h1>
<div class="form-container">
    <form>
        <div class="form-row">
            <label for="projectName">Project Name</label>
            <input type="text" id="projectName" placeholder="Project name" required>
        </div>
        <div class="form-row">
            <label for="assignedTo">Assigned to</label>
            <select id="assignedTo" required>
                <option value="Er Merry Petision">Er Merry Petision</option>
                <option value="John Doe">John Doe</option>
            </select>
        </div>
        <div class="form-row">
            <label for="startDate">Start Date</label>
            <input type="date" id="startDate" required>
        </div>
        <div class="form-row">
            <label for="endDate">End Date</label>
            <input type="date" id="endDate" required>
        </div>
        <div class="form-row">
            <label>Priority</label>
            <div class="radio-group">
                <label><input type="radio" name="priority" value="High"> High</label>

```

```

        <label><input type="radio" name="priority" value="Average"> Average</label>
        <label><input type="radio" name="priority" value="Low"> Low</label>
    </div>
</div>
<div class="form-row">
    <label for="description">Description</label>
    <textarea id="description" rows="1" placeholder="Enter description" required></textarea>
</div><br>
<div>
    <button type="submit" class="submit-btn">Submit</button>
    <button type="reset" class="clear-btn">Clear</button>
</div>
</form>
</div>
</body>
</html>

```

** SLIP 2_Q1 : Create a container add row inside it and add 3 columns inside row using BootStrap.

==> <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap Container with Columns</title>

<!-- Bootstrap CSS -->

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-5">

<div class="row">

<!-- Column 1 -->

<div class="col-md-4">

<h2>Column 1</h2>

<p>This is the first column.</p>

</div>

<!-- Column 2 -->

<div class="col-md-4">

<h2>Column 2</h2>

<p>This is the second column.</p>

</div>

<!-- Column 3 -->

<div class="col-md-4">

<h2>Column 3</h2>

<p>This is the third column.</p>

</div>

</div>

</div>

<!-- Bootstrap JS and dependencies (optional) -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script

src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

** SLIP 3_Q1 : Write a bootstrap application to display thumbnails of the images.

==> <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Image Thumbnails</title>

<!-- Bootstrap CSS (via CDN) -->

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<!-- Container for the thumbnails -->

<div class="container my-5">

<h2 class="text-center mb-4">Image Thumbnails</h2>

<!-- Row for images -->

<div class="row">

<!-- First image thumbnail -->

<div class="col-md-4 mb-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Image 1</h5>

</div>

</div>

</div>

<!-- Second image thumbnail -->

<div class="col-md-4 mb-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Image 2</h5>

</div>

</div>

</div>

<!-- Third image thumbnail -->

<div class="col-md-4 mb-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Image 3</h5>

</div>

</div>

</div>

</div>

</div>

```

<!-- Bootstrap JS and dependencies -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

** SLIP 4_Q1 : Write a bootstrap program for the following “The .table class adds basic styling (light padding and only horizontal dividers) to a table” The table can have the first name, last name, and email id as columns

```

==> <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Table Example</title>
  <!-- Bootstrap CSS (via CDN) -->
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>

  <!-- Container for the table -->
  <div class="container my-5">
    <h2 class="text-center mb-4">User Information</h2>

    <!-- Bootstrap table with .table class -->
    <table class="table">
      <thead>
        <tr>
          <th scope="col">First Name</th>
          <th scope="col">Last Name</th>
          <th scope="col">Email ID</th>
        </tr>
      </thead>
      <tbody>
        <!-- Table Row 1 -->
        <tr>
          <td>John</td>
          <td>Doe</td>
          <td>john.doe@example.com</td>
        </tr>
        <!-- Table Row 2 -->
        <tr>
          <td>Jane</td>
          <td>Smith</td>
          <td>jane.smith@example.com</td>
        </tr>
        <!-- Table Row 3 -->
        <tr>
          <td>Emily</td>
          <td>Johnson</td>
          <td>emily.johnson@example.com</td>
        </tr>
      </tbody>
    </table>
  </div>

```

```

    </table>
</div>

<!-- Bootstrap JS and dependencies -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>
</html>

```

**** SLIP 5_Q1 :** Write a HTML code, which generate the following output [Apply border, border radius tags]

==> <!DOCTYPE html>

<html lang="en">

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>List of Persons</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 20px; }
    table { width: 100%; border-collapse: collapse; border-radius: 10px; box-shadow: 0 4px 8px
    rgba(0, 0, 0, 0.1); }
    th, td { border: 1px solid #000; padding: 8px; text-align: left; }
    th { background-color: #f2f2f2; }
    tr:hover { background-color: #f9f9f9; }
    hr { border: none; height: 2px; background-color: #000; margin: 20px 0; }
  </style>
</head>

```

```

<body>
  <hr>
  <table>
    <caption>List of Persons</caption>
    <thead>
      <tr>
        <th>Srno</th>
        <th>Person Name</th>
        <th>Age</th>
        <th>Country</th>
      </tr>
    </thead>
    <tbody>
      <tr><td>1</td><td>John Doe</td><td>30</td><td>USA</td></tr>
      <tr><td>2</td><td>Jane Smith</td><td>25</td><td>UK</td></tr>
      <tr><td>3</td><td>Emily Johnson</td><td>28</td><td>Canada</td></tr>
    </tbody>
  </table>
</body>
</html>

```

**** SLIP 6_Q1 :** Create a web page being rendered in the browser consists of many things - logo, informative text, pictures, hyperlinks, navigational structure and table.

==> <!DOCTYPE html>

<html lang="en">

<head>

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Sample Web Page</title>
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 20px;
    background-color: #f4f4f4; /* Corrected background color */
  }
  header {
    text-align: center;
    margin-bottom: 20px;
  }
  nav {
    background-color: #333; /* Corrected background color */
    color: white;
    padding: 10px;
  }
  nav a {
    color: white;
    margin: 0 15px;
    text-decoration: none;
  }
  nav a:hover {
    text-decoration: underline;
  }
  .content {
    display: flex;
    justify-content: space-between;
    margin-bottom: 20px;
  }
  .sidebar {
    width: 30%;
  }
  .main {
    width: 65%;
  }
  table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
  }
  table, th, td {
    border: 1px solid #333; /* Corrected border color */
  }
  th, td {
    padding: 8px;
    text-align: left;
  }
  th {
    background-color: #4CAF50; /* Corrected background color */
    color: white;
  }
  footer {
    text-align: center;
    padding: 10px;
  }
```

```
background-color: #333; /* Footer background */
color: white;
position: relative;
bottom: 0;
width: 100%;
}
</style>
</head>
<body>
  <header>
    
    <h1>Welcome to Our Website</h1>
  </header>

  <!-- Navigation Bar -->
  <nav>
    <a href="#home">Home</a>
    <a href="#about">About Us</a>
    <a href="#services">Services</a>
    <a href="#contact">Contact</a>
  </nav>

  <div class="content">
    <!-- Sidebar Content -->
    <div class="sidebar">
      <h2>Informative Text</h2>
      <p>This website is designed to provide you with the latest information and services we offer.
Explore our pages
to find out more!</p>
      <p><a href="https://example.com" target="_blank">Learn more about us</a></p>
    </div>

    <!-- Main Content -->
    <div class="main">
      <h2>Featured Images</h2>
      
      

      <h2>Our Services</h2>
      <table>
        <tr>
          <th>Service</th>
          <th>Description</th>
          <th>Price</th>
        </tr>
        <tr>
          <td>Consulting</td>
          <td>Expert advice for your business.</td>
          <td>$100/hr</td>
        </tr>
        <tr>
          <td>Design</td>
          <td>Creative design solutions.</td>
          <td>$500/project</td>
        </tr>
      </table>
    </div>
  </div>
</body>
</html>
```



```

        <td>Development</td>
        <td>Building web applications.</td>
        <td>$2000/project</td>
    </tr>
</table>
</div>
</div>

<!-- Footer -->
<footer>
    <p>&copy; 2024 Our Website. All rights reserved.</p>
</footer>
</body>
</html>

```

**** SLIP 7_Q1 :** Create a 3D text, apply appropriate font, style, color. Use : Hover in the style selector so that the 3D effects appear only when you hover over the text

```

==> <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>3D Text Hover Effect</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #f0f0f0;
            margin: 0;
        }

        .text {
            font-size: 60px;
            font-weight: bold;
            color: #333;
            text-transform: uppercase;
            transition: all 0.3s ease;
            cursor: pointer;
        }

        /* 3D effect on hover */
        .text:hover {
            color: #ff5733; /* Change color on hover */
            text-shadow:
                3px 3px 5px rgba(0, 0, 0, 0.4),
                -3px -3px 5px rgba(0, 0, 0, 0.2),
                5px 5px 10px rgba(0, 0, 0, 0.3);
            transform: translateZ(30px) scale(1.1); /* Creates the 3D effect */
        }
    </style>
</head>
<body>
    <div class="text">3D Text Effect</div>

```

```
</body>
</html>
```

**** SLIP 8_Q1 : Create a button with different style (Secondary, Primary, Success, Error, Info, Warning, Danger) using BootStrap**

==> <!DOCTYPE html>

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

```
<head>
```

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

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

```
  <title>Bootstrap Button Styles</title>
```

```
  <!-- Bootstrap CSS -->
```

```
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
```

```
</head>
```

```
<body class="container mt-5">
```

```
  <h2>Bootstrap Button Styles</h2>
```

```
  <p>Below are various buttons with different styles in Bootstrap:</p>
```

```
  <!-- Button Group -->
```

```
  <div class="btn-group" role="group" aria-label="Button group">
```

```
    <!-- Primary Button -->
```

```
    <button type="button" class="btn btn-primary">Primary</button>
```

```
    <!-- Secondary Button -->
```

```
    <button type="button" class="btn btn-secondary">Secondary</button>
```

```
    <!-- Success Button -->
```

```
    <button type="button" class="btn btn-success">Success</button>
```

```
    <!-- Danger Button -->
```

```
    <button type="button" class="btn btn-danger">Danger</button>
```

```
    <!-- Warning Button -->
```

```
    <button type="button" class="btn btn-warning">Warning</button>
```

```
    <!-- Info Button -->
```

```
    <button type="button" class="btn btn-info">Info</button>
```

```
    <!-- Light Button (alternative style) -->
```

```
    <button type="button" class="btn btn-light">Light</button>
```

```
    <!-- Dark Button -->
```

```
    <button type="button" class="btn btn-dark">Dark</button>
```

```
  </div>
```

```
  <!-- Bootstrap JS and dependencies -->
```

```
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
```

```
  <script
```

```
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
```

```
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

```
</body>
```

```
</html>
```

**** SLIP 9_Q1 : Write an HTML 5 program for student registration form for college admission. Use input type like search, email, date etc**

==> <!DOCTYPE html>

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

```
<head>
```

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

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

<title>College Admission - Student Registration</title>

<style>

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  margin: 0;
  padding: 0;
}
.container {
  width: 70%;
  margin: 50px auto;
  padding: 20px;
  background-color: white;
  border-radius: 8px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
h1 {
  text-align: center;
  color: #4CAF50;
}
form {
  display: flex;
  flex-direction: column;
}
label {
  margin-top: 10px;
  font-weight: bold;
}
input, select, textarea {
  margin-top: 5px;
  padding: 8px;
  border-radius: 4px;
  border: 1px solid #ccc;
}
input[type="submit"], button {
  background-color: #4CAF50;
  color: white;
  padding: 10px 15px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  margin-top: 20px;
}
input[type="submit"]:hover, button:hover {
  background-color: #45a049;
}
.form-row {
  display: flex;
  justify-content: space-between;
}
.form-row > div {
  width: 48%;
}
```

</style>

</head>

<body>

```
<div class="container">
  <h1>Student Registration Form</h1>
  <form>
    <!-- Student Name -->
    <label for="fullName">Full Name:</label>
    <input type="text" id="fullName" name="fullName" placeholder="Enter your full name"
required>

    <!-- Search for Course -->
    <label for="course">Search for Course:</label>
    <input type="search" id="course" name="course" placeholder="Search for a course"
required>

    <!-- Date of Birth -->
    <div class="form-row">
      <div>
        <label for="dob">Date of Birth:</label>
        <input type="date" id="dob" name="dob" required>
      </div>

      <!-- Gender -->
      <div>
        <label>Gender:</label>
        <div>
          <input type="radio" id="male" name="gender" value="male" required>
          <label for="male">Male</label>
        </div>
        <div>
          <input type="radio" id="female" name="gender" value="female" required>
          <label for="female">Female</label>
        </div>
      </div>
    </div>

    <!-- Email -->
    <label for="email">Email Address:</label>
    <input type="email" id="email" name="email" placeholder="Enter your email" required>

    <!-- Phone Number -->
    <label for="phone">Phone Number:</label>
    <input type="tel" id="phone" name="phone" placeholder="Enter your phone number"
required>

    <!-- Address -->
    <label for="address">Address:</label>
    <textarea id="address" name="address" rows="4" placeholder="Enter your address"
required></textarea>

    <!-- Preferred Admission Date -->
    <label for="admissionDate">Preferred Admission Date:</label>
    <input type="date" id="admissionDate" name="admissionDate" required>

    <!-- Terms and Conditions -->
    <div>
      <label for="terms">
```

```
<input type="checkbox" id="terms" name="terms" required>
```

```
I agree to the <a href="#">terms and conditions</a>.
```

```
</label>
```

```
</div>
```

```
<!-- Submit Button -->
```

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

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

** SLIP 10_Q1 : Create a web page that shows use of transition properties, transition delay and duration effect.

==> <!DOCTYPE html>

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

```
<head>
```

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

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

```
<title>CSS Transition Example</title>
```

```
<style>
```

```
/* Basic body and page styling */
```

```
body {
```

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

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

```
    margin: 0;
```

```
    padding: 20px;
```

```
    text-align: center;
```

```
}
```

```
h1 {
```

```
    color: #333;
```

```
    font-size: 30px;
```

```
}
```

```
/* Styling the box */
```

```
.transition-box {
```

```
    width: 200px;
```

```
    height: 200px;
```

```
    background-color: #3498db; /* Initial color */
```

```
    margin: 50px auto;
```

```
    border-radius: 10px;
```

```
    transition: background-color 1s ease, transform 0.5s ease 0.5s, border-radius 1s ease 1s;
```

```
}
```

```
/* Hover effects */
```

```
.transition-box:hover {
```

```
    background-color: #e74c3c; /* Change background color */
```

```
    transform: scale(1.2); /* Scale up the box */
```

```
    border-radius: 50%; /* Change the shape to a circle */
```

```
}
```

```
/* Add additional styling for clarity */
```

```
p {
```

```
    font-size: 18px;
```

```

        color: #555;
    }

    /* Container styling */
    .container {
        margin-top: 50px;
    }

</style>
</head>
<body>

    <div class="container">
        <h1>CSS Transition, Duration & Delay</h1>
        <p>Hover over the box to see the transitions with delay and duration effects.</p>

        <!-- Box to demonstrate transitions -->
        <div class="transition-box"></div>
    </div>

</body>
</html>

```

**** SLIP 11_Q1 :** Write a HTML code which will divide web page in three frames. First frame should consists of company name as heading. Second frame should consists of name of departments with hyperlink. Once click on any department, it should display information of that department in third frame.

```

==> <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Company Information</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            display: flex;
            flex-direction: column;
            height: 100vh;
        }
        header {
            background-color: #4CAF50;
            color: white;
            text-align: center;
            padding: 20px;
        }
        .content {
            display: flex;
            flex-grow: 1;
        }
        .sidebar {
            width: 20%;
            padding: 20px;
            background-color: #f4f4f4;

```

```

        border-right: 1px solid #ccc;
    }
    .main-content {
        width: 80%;
        padding: 20px;
    }
    iframe {
        width: 100%;
        height: 100%;
        border: none;
    }
    h2 {
        color: #333;
    }
    ul {
        list-style-type: none;
        padding: 0;
    }
    li {
        margin: 10px 0;
    }
    a {
        text-decoration: none;
        color: #007BFF;
        cursor: pointer;
    }
    a:hover {
        text-decoration: underline;
    }
</style>
</head>
<body>

    <header>
        <h1>ABC Corporation</h1>
    </header>

    <div class="content">
        <!-- Sidebar: Department List -->
        <div class="sidebar">
            <h2>Departments</h2>
            <ul>
                <li><a href="javascript:void(0);" onclick="loadDepartmentInfo('HR')">HR
Department</a></li>
                <li><a href="javascript:void(0);" onclick="loadDepartmentInfo('Sales')">Sales
Department</a></li>
                <li><a href="javascript:void(0);" onclick="loadDepartmentInfo('Engineering')">Engineering
Department</a></li>
                <li><a href="javascript:void(0);" onclick="loadDepartmentInfo('Finance')">Finance
Department</a></li>
            </ul>
        </div>

        <!-- Main Content Area: Department Info -->
        <div class="main-content">
            <h2>Department Information</h2>

```

```

        <iframe id="departmentInfo" src="about:blank"></iframe>
    </div>
</div>

<script>
    function loadDepartmentInfo(department) {
        let infoHTML = "";

        switch(department) {
            case 'HR':
                infoHTML = `
                    <h3>HR Department</h3>
                    <p>The HR Department handles recruitment, employee benefits, and employee
relations.</p>
                `;
                break;
            case 'Sales':
                infoHTML = `
                    <h3>Sales Department</h3>
                    <p>The Sales Department is responsible for generating revenue through customer
acquisition and retention.</p>
                `;
                break;
            case 'Engineering':
                infoHTML = `
                    <h3>Engineering Department</h3>
                    <p>The Engineering Department focuses on product development, maintenance,
and technology implementation.</p>
                `;
                break;
            case 'Finance':
                infoHTML = `
                    <h3>Finance Department</h3>
                    <p>The Finance Department manages the company's finances, including budgeting,
forecasting, and investments.</p>
                `;
                break;
            default:
                infoHTML = "<p>Please select a department to view details.</p>";
        }

        // Inject the department information into the iframe
        const iframe = document.getElementById('departmentInfo');
        const doc = iframe.contentWindow.document;
        doc.open();
        doc.write(`
            <!DOCTYPE html>
            <html lang="en">
            <head>
                <meta charset="UTF-8">
                <meta name="viewport" content="width=device-width, initial-scale=1.0">
                <title>${department} Department</title>
                <style>
                    body {
                        font-family: Arial, sans-serif;
                        margin: 20px;

```



```

        background-color: #f9f9f9;
    }
</style>
</head>
<body>
    ${infoHTML}
</body>
</html>
`);
doc.close();
}
</script>
</body>
</html>

```

**** SLIP 12_Q1 :** Design an appropriate HTML form for customer registration visiting a departmental store. Form should consist of fields such as name, contact no, gender, preferred days of purchasing, favorite item (to be selected from a list of items), suggestions etc. You should provide button to submit as well as reset the form contents.

==> <!DOCTYPE html>

<html lang="en">

```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Customer Registration</title>
    <style>
        body { font-family: Arial, sans-serif; background-color: #f4f4f4; padding: 20px; }
        .form-container { max-width: 400px; margin: auto; background: #fff; padding: 20px;
border-radius: 8px; box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); }
        h2 { text-align: center; margin-bottom: 20px; }
        label { display: block; margin: 10px 0 5px; }
        input, select, textarea { width: 100%; padding: 8px; margin-bottom: 15px; border: 1px solid
#ccc; border-radius: 5px; }
        button { width: 48%; padding: 10px; margin-right: 4%; border: none; cursor: pointer;
border-radius: 5px; }
        button[type="submit"] { background: #4CAF50; color: white; }
        button[type="reset"] { background: #f44336; color: white; }
        button:hover { opacity: 0.9; }
    </style>
</head>
<body>

```

```

<div class="form-container">

```

```

    <h2>Register</h2>

```

```

    <form>

```

```

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

```

```

        <input type="text" id="name" name="name" required>

```

```

        <label for="contact">Contact Number</label>

```

```

        <input type="tel" id="contact" name="contact" pattern="[0-9]{10}" required>

```

```

        <label for="gender">Gender</label>

```

```

        <select id="gender" name="gender" required>

```

```

            <option value="male">Male</option>

```

```

            <option value="female">Female</option>

```

```

    <option value="other">Other</option>
</select>

<label for="preferred-days">Preferred Days</label>
<select id="preferred-days" name="preferred-days[]" required>
    <option value="Monday">Monday</option>
    <option value="Tuesday">Tuesday</option>
    <option value="Wednesday">Wednesday</option>
    <option value="Thursday">Thursday</option>
    <option value="Friday">Friday</option>
    <option value="Saturday">Saturday</option>
    <option value="Sunday">Sunday</option>
</select>

```

```

<label for="favorite-item">Favorite Item</label>
<select id="favorite-item" name="favorite-item" required>
    <option value="clothes">Clothes</option>
    <option value="groceries">Groceries</option>
    <option value="electronics">Electronics</option>
    <option value="furniture">Furniture</option>
    <option value="toys">Toys</option>
</select>

```

```

<label for="suggestions">Suggestions</label>
<textarea id="suggestions" name="suggestions" rows="2"></textarea>

```

```

<div style="text-align: center;">
    <button type="submit">Submit</button>
    <button type="reset">Reset</button>
</div>

```

```

</form>

```

```

</div>

```

```

</body>

```

```

</html>

```

** SLIP 13_Q1 : Create a useful web with the following information and structure using HTML5 tags like: <header> , <footer>, <nav>, <aside>, <section>

```

==> <!DOCTYPE html>

```

```

<html lang="en">

```

```

<head>

```

```

    <meta charset="UTF-8">

```

```

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

```

```

    <title>Departmental Store</title>

```

```

    <style>

```

```

        body { font-family: Arial, sans-serif; margin: 0; padding: 0; background-color: #f4f4f4; }
        header, footer { background-color: #333; color: white; text-align: center; padding: 15px 0; }
        nav { background-color: #444; color: white; padding: 10px 0; }
        nav ul { list-style: none; padding: 0; margin: 0; display: flex; justify-content: center; }
        nav ul li { margin: 0 20px; }
        nav ul li a { color: white; text-decoration: none; font-weight: bold; }
        .container { display: flex; padding: 20px; }
        .main-content { flex: 3; margin-right: 20px; }
        .aside { flex: 1; background-color: #fff; padding: 20px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); }

```

```

    }

```

```
    footer { font-size: 0.8em; }
    section h2 { color: #333; }
    .footer-links { display: flex; justify-content: center; gap: 20px; margin-top: 10px; }
</style>
</head>
<body>

  <header>
    <h1>Welcome to Our Departmental Store</h1>
    <p>Your one-stop shop for all your needs</p>
  </header>

  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">Products</a></li>
      <li><a href="#">Sales</a></li>
      <li><a href="#">Contact</a></li>
      <li><a href="#">About Us</a></li>
    </ul>
  </nav>

  <div class="container">
    <main class="main-content">
      <section>
        <h2>Featured Products</h2>
        <p>Check out our latest collection of clothes, electronics, and groceries!</p>
        <!-- Content such as images or product descriptions can go here -->
      </section>

      <section>
        <h2>Store Hours</h2>
        <p>We are open every day from 9 AM to 9 PM.</p>
      </section>

      <section>
        <h2>Customer Reviews</h2>
        <p>Here's what our customers say:</p>
        <ul>
          <li>"Excellent service and great variety!"</li>
          <li>"I love shopping here. Friendly staff and good prices."</li>
          <li>"A fantastic shopping experience every time!"</li>
        </ul>
      </section>
    </main>

    <aside class="aside">
      <h3>Store News</h3>
      <p>Stay tuned for our seasonal sale next month!</p>
      <h3>Helpful Links</h3>
      <ul>
        <li><a href="#">Customer Support</a></li>
        <li><a href="#">Delivery Information</a></li>
        <li><a href="#">Return Policy</a></li>
      </ul>
    </aside>
```

</div>

<footer>

<p>© 2024 Departmental Store</p>

<div class="footer-links">

Privacy Policy

Terms of Service

FAQ

</div>

</footer>

</body>

</html>

**** SLIP 14_Q1 :** Design an HTML form to take the information of a customer for booking a travel plan consisting of fields such as name, address, contact no., gender, preferred season(Checkboxes), location type(to be selected from a list) etc. You should provide button to submit as well as reset the form contents

==> <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Travel Plan Booking</title>

<style>

body { font-family: Arial, sans-serif; background-color: #f4f4f4; padding: 20px; }

.form-container { max-width: 500px; margin: auto; background: #fff; padding: 20px;

border-radius: 8px; box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); }

label { display: block; margin: 10px 0 5px; }

input, select, textarea { width: 100%; padding: 8px; margin-bottom: 15px; border: 1px solid

#ccc; border-radius: 5px; }

button { width: 48%; padding: 10px; margin-right: 4%; border: none; cursor: pointer;

border-radius: 5px; }

button[type="submit"] { background: #4CAF50; color: white; }

button[type="reset"] { background: #f44336; color: white; }

button:hover { opacity: 0.9; }

.checkbox-group { display: flex; gap: 15px; margin-bottom: 15px; }

</style>

</head>

<body>

<div class="form-container">

<h2>Travel Plan Booking</h2>

<form>

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

<input type="text" id="name" name="name" required>

<label for="address">Address</label>

<textarea id="address" name="address" rows="4" required></textarea>

<label for="contact">Contact Number</label>

<input type="tel" id="contact" name="contact" pattern="[0-9]{10}" required>

<label for="gender">Gender</label>

<select id="gender" name="gender" required>

```

        <option value="male">Male</option>
        <option value="female">Female</option>
        <option value="other">Other</option>
    </select>

    <label>Preferred Season</label>
    <div class="checkbox-group">
        <label><input type="checkbox" name="season" value="summer"> Summer</label>
        <label><input type="checkbox" name="season" value="winter"> Winter</label>
        <label><input type="checkbox" name="season" value="monsoon"> Monsoon</label>
    </div>

    <label for="location-type">Location Type</label>
    <select id="location-type" name="location-type" required>
        <option value="beach">Beach</option>
        <option value="mountain">Mountain</option>
        <option value="forest">Forest</option>
        <option value="city">City</option>
        <option value="desert">Desert</option>
    </select>

    <div style="text-align: center;">
        <button type="submit">Submit</button>
        <button type="reset">Reset</button>
    </div>
</form>
</div>

</body>
</html>

```

**** SLIP 15_Q1 : Create HTML web-page using Bootstrap as shown below**

```

==> <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        /* Making font bold for labels and button */
        .form-label {
            font-weight: bold;
        }
        .btn-warning {
            font-weight: bold;
        }
    </style>
</head>
<body>
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-md-6">

```

```

<div class="card shadow-lg">
  <div class="card-header text-center bg-light">
    <h3 class="mb-0">Registration Form</h3>
  </div>
  <div class="card-body">
    <form>
      <div class="mb-3">
        <label for="firstName" class="form-label">First Name</label>
        <div class="input-group">
          <span class="input-group-text"><i class="bi bi-person"></i></span>
          <input type="text" class="form-control" id="firstName" placeholder="First
Name" required>
        </div>
      </div>
      <div class="mb-3">
        <label for="lastName" class="form-label">Last Name</label>
        <div class="input-group">
          <span class="input-group-text"><i class="bi bi-person"></i></span>
          <input type="text" class="form-control" id="lastName" placeholder="Last
Name" required>
        </div>
      </div>
      <div class="mb-3">
        <label for="department" class="form-label">Department / Office</label>
        <div class="input-group">
          <span class="input-group-text"><i class="bi bi-building"></i></span>
          <select class="form-select" id="department" required>
            <option selected>Select your Department/Office</option>
            <option value="HR">HR</option>
            <option value="IT">IT</option>
            <option value="Admin">Admin</option>
          </select>
        </div>
      </div>
      <div class="mb-3">
        <label for="username" class="form-label">Username</label>
        <div class="input-group">
          <span class="input-group-text"><i class="bi bi-person-circle"></i></span>
          <input type="text" class="form-control" id="username"
placeholder="Username" required>
        </div>
      </div>
      <div class="mb-3">
        <label for="password" class="form-label">Password</label>
        <div class="input-group">
          <span class="input-group-text"><i class="bi bi-lock"></i></span>
          <input type="password" class="form-control" id="password"
placeholder="Password" required>
        </div>
      </div>
      <div class="mb-3">
        <label for="confirmPassword" class="form-label">Confirm Password</label>
        <div class="input-group">
          <span class="input-group-text"><i class="bi bi-lock-fill"></i></span>
          <input type="password" class="form-control" id="confirmPassword"
placeholder="Confirm Password" required>

```



```

        display: flex;
        background-color: #f8f9fa;
        border-radius: 8px;
    }
    .contact-info {
        background-color: #ff9800;
        color: solid black;
        padding: 20px;
        text-align: center;
        border-radius: 8px 0 0 8px;
        font-size: 1.2em;
        width: 250px;
        display: flex;
        flex-direction: column;
        justify-content: center;
    }
    .form-section {
        padding: 30px;
        flex-grow: 1;
    }
    .contact-form .form-group label {
        font-size: 1em;
    }
    .contact-form .form-control {
        font-size: 1em;
        padding: 10px;
    }
    .btn-submit {
        background-color: #2c3e50;
        color: white;
        font-size: 1em;
        padding: 10px 20px;
    }
</style>
</head>
<body>

<div class="form-wrapper">
  <div class="form-container">
    <div class="contact-info">
      <h4>Contact Us</h4>
      <p>We would love to hear from you!</p>
    </div>
    <div class="form-section">
      <form class="contact-form">
        <div class="form-group">
          <label for="firstName">First Name:</label>
          <input type="text" class="form-control" id="firstName" placeholder="Enter First Name">
        </div>
        <div class="form-group">
          <label for="lastName">Last Name:</label>
          <input type="text" class="form-control" id="lastName" placeholder="Enter Last Name">
        </div>
        <div class="form-group">
          <label for="email">Email:</label>
          <input type="email" class="form-control" id="email" placeholder="Enter email">
        </div>
      </form>
    </div>
  </div>
</div>

```



```

        </div>
        <div class="form-group">
            <label for="comment">Comment:</label>
            <textarea class="form-control" id="comment" rows="4" placeholder="Enter your
comment"></textarea>
        </div>
        <button type="submit" class="btn btn-submit">Submit</button>
    </form>
</div>
</div>
</div>
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.4.4/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

**** SLIP 17_Q1 : Design HTML 5 Page Using CSS which display the following Box (use Box Model Property in CSS)**

==> <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Box Model Design</title>

<style>

```

    body {
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        background-color: #f0f0f0;
    }

```

```

    .outer-box {
        width: 400px;
        height: 300px;
        background-color: orange; /* Outer box background color */
        border: 5px solid black; /* Outer border */
        display: flex;
        justify-content: center;
        align-items: center;
    }

```

```

    .inner-box {
        width: 300px;
        height: 200px;
        background-color: yellow; /* Inner box background color */
        border: 5px solid black; /* Inner border */
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
    }

```

```

        box-sizing: border-box; /* Ensures borders don't affect size */
    }

    .text-box {
        background-color: yellow;
        border: 5px solid black;
        padding: 10px;
        margin: 10px;
        text-align: center;
        font-weight: bold;
        font-family: Arial, sans-serif;
    }
</style>
</head>
<body>
    <div class="outer-box">
        <div class="inner-box">
            <div class="text-box">M.Sc. ( Computer Sci )</div>
            <div class="text-box">Academic Year 2023-24</div>
        </div>
    </div>
</body>
</html>

```

**** SLIP 18_Q1 :** Create a web page, place an image in center and apply 2d transformation on it. (rotation, scaling, translation)

==> <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Image Transformation</title>

<style>

body {

margin: 0;

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

background-color: #f4f4f4;

}

.image-container {

width: 300px;

height: 300px;

overflow: hidden;

position: relative;

}

img {

width: 100%;

height: 100%;

object-fit: cover;

transition: transform 1s ease; /* Smooth transition for transformations */

}

/* Applying 2D Transformations */

```

    img:hover {
        transform: rotate(45deg) scale(1.2) translate(50px, 50px); /* Rotation, Scaling, Translation */
    }
</style>
</head>
<body>

    <div class="image-container">
        
    </div>

</body>
</html>

```

**** SLIP 19_Q1 :** Create a web page in which show a button with a text “start download” , when click in start download a progress bar must be initialized with value 0 then increase by 10 in each second, change the color of progress bar after every three seconds..

```

==> <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Download Progress</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            padding: 20px;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            background-color: #f4f4f4;
        }
        .progress-container {
            width: 80%;
            max-width: 500px;
            margin-top: 20px;
        }
        #progress-bar {
            width: 100%;
            height: 30px;
            border: 1px solid #ccc;
            background-color: #e0e0e0;
            border-radius: 5px;
            position: relative;
        }
        #progress {
            height: 100%;
            width: 0;
            background-color: green;
            border-radius: 5px;
        }
        button {
            padding: 10px 20px;

```

```

    font-size: 16px;
    background-color: #4CAF50;
    color: white;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}
button:hover {
    opacity: 0.9;
}
</style>
</head>
<body>

<h1>Download Progress</h1>
<button id="start-btn">Start Download</button>

<div class="progress-container">
    <div id="progress-bar">
        <div id="progress"></div>
    </div>
</div>

<script>
    const startBtn = document.getElementById("start-btn");
    const progress = document.getElementById("progress");
    let progressValue = 0;
    let interval;
    let colorChangeInterval;

    const colors = ['green', 'blue', 'orange', 'red']; // Colors to change every 3 seconds
    let colorIndex = 0;

    // Function to start the download simulation
    startBtn.addEventListener("click", function() {
        // Disable the button after clicking
        startBtn.disabled = true;
        startBtn.innerText = "Downloading...";

        progressValue = 0;
        progress.style.width = `${progressValue}%`;

        // Increase the progress bar value every second
        interval = setInterval(function() {
            progressValue += 10;
            if (progressValue <= 100) {
                progress.style.width = `${progressValue}%`;
            }

            // Change color every 3 seconds
            if (progressValue % 30 === 0) {
                progress.style.backgroundColor = colors[colorIndex];
                colorIndex = (colorIndex + 1) % colors.length; // Cycle through the colors
            }

            // Stop the interval when progress reaches 100%

```

```

        if (progressValue >= 100) {
            clearInterval(interval);
            startBtn.innerText = "Download Complete";
            setTimeout(() => startBtn.disabled = false, 2000); // Re-enable button after 2 seconds
        }
    }, 1000); // Increase progress every second
});
</script>

</body>
</html>

```

**** SLIP 20_Q1 :** Create a web page , show a button with a text “start download” , when click in start download a progress bar must be initialized with value 0 then increase by 5 in each second then at the end of downloading process alert the message “Download completed”

```

==> <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Download Progress</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            padding: 20px;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            background-color: #f4f4f4;
        }
        .progress-container {
            width: 80%;
            max-width: 500px;
            margin-top: 20px;
        }
        #progress-bar {
            width: 100%;
            height: 30px;
            border: 1px solid #ccc;
            background-color: #e0e0e0;
            border-radius: 5px;
            position: relative;
        }
        #progress {
            height: 100%;
            width: 0;
            background-color: green;
            border-radius: 5px;
        }
        button {
            padding: 10px 20px;
            font-size: 16px;
            background-color: #4CAF50;

```

```

        color: white;
        border: none;
        cursor: pointer;
        border-radius: 5px;
    }
    button:hover {
        opacity: 0.9;
    }
</style>
</head>
<body>

<h1>Download Progress</h1>
<button id="start-btn">Start Download</button>

<div class="progress-container">
    <div id="progress-bar">
        <div id="progress"></div>
    </div>
</div>

<script>
    const startBtn = document.getElementById("start-btn");
    const progress = document.getElementById("progress");
    let progressValue = 0;
    let interval;

    // Function to start the download simulation
    startBtn.addEventListener("click", function() {
        // Disable the button after clicking
        startBtn.disabled = true;
        startBtn.innerText = "Downloading...";

        progressValue = 0;
        progress.style.width = `${progressValue}%`;

        // Increase the progress bar value every second
        interval = setInterval(function() {
            progressValue += 5;
            if (progressValue <= 100) {
                progress.style.width = `${progressValue}%`;
            }

            // Stop the interval when progress reaches 100%
            if (progressValue >= 100) {
                clearInterval(interval);
                alert("Download completed");
                startBtn.innerText = "Download Complete";
            }
        }, 1000); // Increase progress every second
    });
</script>

</body>
</html>

```

** SLIP 21_Q1 : Write a css3 script for the student registration form with appropriate message display also high light compulsory fields in a different color

==> <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Student Registration Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.form-container {

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

width: 300px;

}

h2 {

text-align: center;

margin-bottom: 20px;

}

.form-group {

margin-bottom: 15px;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"],

input[type="email"],

input[type="password"],

select {

width: 100%;

padding: 8px;

border: 1px solid #ccc;

border-radius: 4px;

transition: border-color 0.3s;

}

input[type="text"].required,

input[type="email"].required,

input[type="password"].required {

```

        border-color: #ff4d4d; /* Red for compulsory fields */
    }

    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        padding: 10px;
        border: none;
        border-radius: 4px;
        cursor: pointer;
        width: 100%;
        transition: background-color 0.3s;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }

    .message {
        margin-top: 15px;
        text-align: center;
        color: #ff4d4d; /* Red for error messages */
    }
</style>
</head>
<body>
    <div class="form-container">
        <h2>Student Registration</h2>
        <form id="registrationForm">
            <div class="form-group">
                <label for="name">Name <span class="required">*</span></label>
                <input type="text" id="name" class="required" required>
            </div>
            <div class="form-group">
                <label for="email">Email <span class="required">*</span></label>
                <input type="email" id="email" class="required" required>
            </div>
            <div class="form-group">
                <label for="password">Password <span class="required">*</span></label>
                <input type="password" id="password" class="required" required>
            </div>
            <div class="form-group">
                <label for="course">Select Course</label>
                <select id="course">
                    <option value="">--Select a course--</option>
                    <option value="math">Mathematics</option>
                    <option value="science">Science</option>
                    <option value="history">History</option>
                </select>
            </div>
            <input type="submit" value="Register">
            <div class="message" id="message"></div>
        </form>
    </div>

<script>

```



```

document.getElementById('registrationForm').addEventListener('submit', function(event) {
    event.preventDefault(); // Prevent form submission

    // Clear previous message
    const messageDiv = document.getElementById('message');
    messageDiv.textContent = "";

    // Check if required fields are filled
    const name = document.getElementById('name').value.trim();
    const email = document.getElementById('email').value.trim();
    const password = document.getElementById('password').value.trim();

    if (!name || !email || !password) {
        messageDiv.textContent = 'Please fill in all compulsory fields.';
        return;
    }

    // If validation passes
    messageDiv.textContent = 'Registration successful!';
    messageDiv.style.color = '#4CAF50'; // Green for success message
    // Here you can add code to submit the form data to a server
});
</script>
</body>
</html>

```

**** SLIP 21_Q2 :** Model the following Medical information as a graph model, and answer the following queries using Cypher. There are various brands of medicine like Dr. Reddy, Cipla, SunPharma etc. Their uses vary across different states in India. The uses of medicine is measured as %, with a high use defined as $\geq 90\%$, Medium Use between 50 to 90%, and Low Use $< 50\%$. Each medicine manufactures various types of medicine products like Tablet, Syrup, and Powder etc.

```

==> // Create Brands
CREATE (b1:Brand {name: "Dr. Reddy"}),
      (b2:Brand {name: "Cipla"}),
      (b3:Brand {name: "SunPharma"});

// Create Medicines
CREATE (m1:Medicine {name: "MedicineA", use_category: "High", state: "Rajasthan",
usage_percentage: 92}),
      (m2:Medicine {name: "MedicineB", use_category: "Medium", state: "Gujarat",
usage_percentage: 75}),
      (m3:Medicine {name: "MedicineC", use_category: "Low", state: "Rajasthan",
usage_percentage: 45}),
      (m4:Medicine {name: "MedicineD", use_category: "High", state: "Gujarat", usage_percentage:
91});

// Create Product Types
CREATE (pt1:ProductType {type: "Tablet"}),
      (pt2:ProductType {type: "Syrup"}),
      (pt3:ProductType {type: "Powder"});

// Create Relationships
MATCH (b:Brand), (m:Medicine)
WHERE (b.name = "Dr. Reddy" AND m.name = "MedicineA") OR
      (b.name = "Cipla" AND m.name = "MedicineB") OR

```

```
(b.name = "SunPharma" AND m.name = "MedicineC") OR
(b.name = "Dr. Reddy" AND m.name = "MedicineD")
CREATE (b)-[:MANUFACTURES]->(m);
```

```
MATCH (m:Medicine), (pt:ProductType)
WHERE (m.name = "MedicineA" AND pt.type = "Tablet") OR
(m.name = "MedicineB" AND pt.type = "Syrup") OR
(m.name = "MedicineC" AND pt.type = "Powder") OR
(m.name = "MedicineD" AND pt.type = "Tablet")
CREATE (m)-[:IS_AVAILABLE_AS]->(pt);
```

a) List the names of different medicines considered in your graph
=> MATCH (m:Medicine)
RETURN m.name AS MedicineName;

b) List the medicines that are highly used in Rajasthan:
=>MATCH (m:Medicine)
WHERE m.use_category = "High" AND m.state = "Rajasthan"
RETURN m.name AS HighlyUsedMedicinesInRajasthan;

c) List the highly used tablet in Gujarat.
=> MATCH (m:Medicine)-[:IS_AVAILABLE_AS]->(pt:ProductType)
WHERE m.use_category = "High" AND m.state = "Gujarat" AND pt.type = "Tablet"
RETURN m.name AS HighlyUsedTabletsInGujarat;

d) List the medicine names manufacturing "Powder"
=> MATCH (m:Medicine)-[:IS_AVAILABLE_AS]->(pt:ProductType)
WHERE pt.type = "Powder"
RETURN m.name AS MedicinesManufacturingPowder;

** SLIP 22_Q1 : Create a web page to create 3D text. Apply all text effects like text shadow, text overflow, wordwrap etc

```
==> <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple 3D Text with Effects</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      padding: 50px;
      background-color: #f4f4f4;
    }

    /* 3D text with shadow */
    .three-d-text {
      font-size: 60px;
      color: #3498db;
      font-weight: bold;
      text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.3);
      letter-spacing: 2px;
    }

    /* Text overflow effect */
```

```

.overflow-text {
    width: 300px;
    margin: 20px auto;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    border: 2px solid #ddd;
    padding: 10px;
    font-size: 18px;
}

/* Word wrapping effect */
.word-wrap {
    width: 300px;
    margin: 20px auto;
    font-size: 18px;
    word-wrap: break-word;
    border: 2px solid #ddd;
    padding: 10px;
}
</style>
</head>
<body>

<!-- 3D Text -->
<h1 class="three-d-text">3D Text Effect</h1>

<!-- Text overflow example -->
<div class="overflow-text">
    This is a long text that will overflow and show ellipsis if it exceeds the container width.
</div>

<!-- Word wrap example -->
<div class="word-wrap">
    This is a long text that will wrap to the next line when the container width is exceeded.
</div>

</body>
</html>

```

**** SLIP 22_Q2 :** Model the following Car Showroom information as a graph model, and answer the queries using Cypher. Consider a car showroom with different models of cars like sofas Honda city, Skoda, Creta, Swift, Ertiga etc. Showroom is divided into different sections, one section for each car model; each section is handled by a sales staff. A sales staff can handle one or more sections. Customer may enquire about car. An enquiry may result in a purchase by the customer.

==> // Create Car Models

```

CREATE (cm1:CarModel {name: "Honda City"}),
      (cm2:CarModel {name: "Skoda"}),
      (cm3:CarModel {name: "Creta"}),
      (cm4:CarModel {name: "Swift"}),
      (cm5:CarModel {name: "Ertiga"});

```

// Create Showroom Sections

```

CREATE (s1:ShowroomSection {name: "Honda Section"}),
      (s2:ShowroomSection {name: "Skoda Section"}),

```

```
(s3:ShowroomSection {name: "Creta Section"}),  
(s4:ShowroomSection {name: "Swift Section"}),  
(s5:ShowroomSection {name: "Ertiga Section"});
```

// Create Sales Staff

```
CREATE (ss1:SalesStaff {name: "John", experience_years: 5}),  
      (ss2:SalesStaff {name: "Doe", experience_years: 3}),  
      (ss3:SalesStaff {name: "Alice", experience_years: 7}),  
      (ss4:SalesStaff {name: "Bob", experience_years: 4});
```

// Assign Sales Staff to Sections

```
MATCH (s:ShowroomSection), (ss:SalesStaff)  
WHERE (s.name = "Honda Section" AND ss.name = "John") OR  
      (s.name = "Skoda Section" AND ss.name = "Doe") OR  
      (s.name = "Creta Section" AND ss.name = "Alice") OR  
      (s.name = "Swift Section" AND ss.name = "John") OR  
      (s.name = "Ertiga Section" AND ss.name = "Bob")  
CREATE (s)-[:ASSIGNED_TO]->(ss);
```

// Link Sections with Car Models

```
MATCH (s:ShowroomSection), (cm:CarModel)  
WHERE (s.name = "Honda Section" AND cm.name = "Honda City") OR  
      (s.name = "Skoda Section" AND cm.name = "Skoda") OR  
      (s.name = "Creta Section" AND cm.name = "Creta") OR  
      (s.name = "Swift Section" AND cm.name = "Swift") OR  
      (s.name = "Ertiga Section" AND cm.name = "Ertiga")  
CREATE (s)-[:HAS]->(cm);
```

// Create Customers

```
CREATE (c1:Customer {name: "Adam", contact: "1234567890"}),  
      (c2:Customer {name: "Eve", contact: "9876543210"});
```

// Create Enquiries

```
MATCH (c:Customer), (cm:CarModel)  
WHERE (c.name = "Adam" AND cm.name = "Honda City") OR  
      (c.name = "Eve" AND cm.name = "Skoda")  
CREATE (c)-[:ENQUIRES {date: "2024-11-18"}]->(cm);
```

// Create Purchases

```
MATCH (c:Customer), (cm:CarModel)  
WHERE (c.name = "Adam" AND cm.name = "Honda City")  
CREATE (c)-[:PURCHASED {date: "2024-11-19", amount: 1500000}]->(cm);
```

a) List the types of cars available in the showroom.

```
=> MATCH (cm:CarModel)  
RETURN cm.name AS CarModelName;
```

b) List the sections handled by Mr. Narayan.

```
=> MATCH (s:ShowroomSection)-[:ASSIGNED_TO]->(ss:SalesStaff)  
WHERE ss.name = "Narayan"  
RETURN s.name AS SectionName;
```

c) List the names of customers who have done only enquiry but not made any purchase.

```
=> MATCH (c:Customer)-[:ENQUIRES]->(cm:CarModel)  
WHERE NOT (c)-[:PURCHASED]->(cm)  
RETURN c.name AS CustomerName, c.contact AS ContactInfo;
```

d) List the highly sale car model.
=> MATCH (c:Customer)-[:PURCHASED]->(cm:CarModel)
RETURN cm.name AS CarModelName, COUNT(c) AS TotalSales
ORDER BY TotalSales DESC
LIMIT 1;

** SLIP 23_Q1 : Create a web page to display image on screen Apply the following

- a. Display image in tiles
- b. Rotate image in clockwise and anticlockwise direction

==> <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Image Tiles with Rotation</title>

<style>

body {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

}

.container {

display: flex;

flex-wrap: wrap;

width: 300px; /* Adjust width according to your image */

height: 300px; /* Adjust height according to your image */

overflow: hidden;

position: relative;

}

.image-tiles {

display: flex;

flex-wrap: wrap;

width: 100%;

height: 100%;

transition: transform 0.5s;

}

.tile {

width: 100px; /* Adjust tile size */

height: 100px; /* Adjust tile size */

overflow: hidden;

}

.tile img {

width: 100%;

height: auto;

}

.controls {

```

        margin-top: 20px;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="image-tiles" id="image-tiles"></div>
    </div>
    <div class="controls">
        <button id="rotate-left">Rotate Left</button>
        <button id="rotate-right">Rotate Right</button>
    </div>
    <script>
        const imageTiles = document.getElementById('image-tiles');
        const rotateLeftButton = document.getElementById('rotate-left');
        const rotateRightButton = document.getElementById('rotate-right');

        let rotationDegrees = 0;

        // Load the image and create tiles
        function createTiles(imageSrc) {
            const rows = 3; // Number of rows
            const cols = 3; // Number of columns

            for (let i = 0; i < rows; i++) {
                for (let j = 0; j < cols; j++) {
                    const tile = document.createElement('div');
                    tile.className = 'tile';
                    const img = document.createElement('img');
                    img.src = imageSrc;
                    img.style.transform = `translate(-${j * 100}px, -${i * 100}px)`;
                    img.style.width = `${rows * 100}px`; // Adjust according to tile size
                    img.style.height = `${cols * 100}px`; // Adjust according to tile size
                    tile.appendChild(img);
                    imageTiles.appendChild(tile);
                }
            }
        }

        // Rotate the image tiles
        function rotateTiles(direction) {
            if (direction === 'left') {
                rotationDegrees -= 90;
            } else if (direction === 'right') {
                rotationDegrees += 90;
            }
            imageTiles.style.transform = `rotate(${rotationDegrees}deg)`;
        }

        // Event listeners for rotation buttons
        rotateLeftButton.addEventListener('click', () => rotateTiles('left'));
        rotateRightButton.addEventListener('click', () => rotateTiles('right'));

        // Initialize the tiles with an image
        createTiles('https://via.placeholder.com/300'); // Replace with your image URL
    </script>

```

```
</body>
</html>
```

**** SLIP 23_Q2 :** Model the following Automobile information system as a graph model, and answer the following queries using Cypher. Consider an Automobile industry manufacturing different types of vehicles like Two- Wheeler, Four-Wheeler, etc. A customer can buy one or more types of vehicle. A person can recommend or rate a vehicle type.

==> // Create Vehicle Types

```
CREATE (v1:VehicleType {type: "Two-Wheeler", characteristics: "fuel-efficient"}),
      (v2:VehicleType {type: "Four-Wheeler", characteristics: "spacious and luxurious"}),
      (v3:VehicleType {type: "Two-Wheeler", characteristics: "compact and affordable"}),
      (v4:VehicleType {type: "Four-Wheeler", characteristics: "electric and eco-friendly"});
```

// Create Customers

```
CREATE (c1:Customer {name: "Alice", contact: "1234567890"}),
      (c2:Customer {name: "Bob", contact: "9876543210"}),
      (c3:Customer {name: "Charlie", contact: "1111111111"});
```

// Create Persons for Recommendations

```
CREATE (p1:Person {name: "David"}),
      (p2:Person {name: "Eve"});
```

// Create Purchase Relationships

```
MATCH (c:Customer), (v:VehicleType)
WHERE (c.name = "Alice" AND v.type = "Two-Wheeler") OR
      (c.name = "Alice" AND v.type = "Four-Wheeler") OR
      (c.name = "Bob" AND v.type = "Two-Wheeler") OR
      (c.name = "Charlie" AND v.type = "Four-Wheeler")
CREATE (c)-[:BOUGHT]->(v);
```

// Create Recommendation Relationships

```
MATCH (p:Person), (v:VehicleType)
WHERE (p.name = "David" AND v.type = "Two-Wheeler") OR
      (p.name = "Eve" AND v.type = "Four-Wheeler") OR
      (p.name = "David" AND v.type = "Four-Wheeler")
CREATE (p)-[:RECOMMENDS {rating: 5}]->(v);
```

a) List the characteristics of four wheeler types.

```
=> MATCH (v:VehicleType)
WHERE v.type = "Four-Wheeler"
RETURN v.characteristics AS Characteristics;
```

b) List the name of customers who bought a two wheeler vehicle.

```
=> MATCH (c:Customer)-[:BOUGHT]->(v:VehicleType)
WHERE v.type = "Two-Wheeler"
RETURN c.name AS CustomerName;
```

c) List the customers who bought more than one type of vehicle.

```
=> MATCH (c:Customer)-[:BOUGHT]->(v:VehicleType)
WITH c.name AS CustomerName, COUNT(DISTINCT v.type) AS VehicleCount
WHERE VehicleCount > 1
RETURN CustomerName;
```

d) List the most recommended vehicle type.

```
=> MATCH (p:Person)-[:RECOMMENDS]->(v:VehicleType)
RETURN v.type AS VehicleType, COUNT(r) AS RecommendationCount
```

```
ORDER BY RecommendationCount DESC
LIMIT 1;
```

** SLIP 24_Q1 : Create an html page named as “calendar.html” Use necessary input types and get following output

```
==> <!DOCTYPE html>
<html>
<head>
<title>Input Types Examples</title>
</head>
<body>
  <h1>Input Types Examples</h1>
  <form>
    <label for="date">Date:</label>
    <input type="date" id="date" name="date"><br><br>

    <label for="datetime">Datetime:</label>
    <input type="datetime-local" id="datetime" name="datetime"><br><br>

    <label for="datetime-local">Datetime-local:</label>
    <input type="datetime-local" id="datetime-local" name="datetime-local"><br><br>

    <label for="month">Month:</label>
    <input type="month" id="month" name="month"><br><br>

    <label for="time">Time:</label>
    <input type="time" id="time" name="time"><br><br>

    <label for="week">Week:</label>
    <input type="week" id="week" name="week"><br><br>

    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

** SLIP 24_Q2 : Model the following Library information system as a graph model, and answer the following queries using Cypher. Consider a library information system having different types of books like text, reference, bibliography etc. A student can buy one or more types of book. A student can recommend or rate a book according to its type.

==> // Create Books

```
CREATE (b1:Book {title: "Introduction to Mathematics", type: "text"}),
      (b2:Book {title: "Advanced Physics", type: "reference"}),
      (b3:Book {title: "History of Science", type: "bibliography"}),
      (b4:Book {title: "Fundamentals of Chemistry", type: "text"}),
      (b5:Book {title: "World Literature", type: "reference"});
```

// Create Students

```
CREATE (s1:Student {name: "Alice", contact: "1234567890"}),
      (s2:Student {name: "Bob", contact: "9876543210"}),
      (s3:Student {name: "Charlie", contact: "1111111111"});
```

// Create Purchase Relationships

```
MATCH (s:Student), (b:Book)
WHERE (s.name = "Alice" AND b.title = "Introduction to Mathematics") OR
      (s.name = "Alice" AND b.title = "Advanced Physics") OR
```



```

(s.name = "Bob" AND b.title = "Fundamentals of Chemistry") OR
(s.name = "Charlie" AND b.title = "World Literature") OR
(s.name = "Charlie" AND b.title = "Advanced Physics")
CREATE (s)-[:BOUGHT]->(b);

// Create Recommendation Relationships
MATCH (s:Student), (b:Book)
WHERE (s.name = "Alice" AND b.title = "Introduction to Mathematics") OR
      (s.name = "Bob" AND b.title = "Advanced Physics") OR
      (s.name = "Charlie" AND b.title = "World Literature") OR
      (s.name = "Alice" AND b.title = "Fundamentals of Chemistry")
CREATE (s)-[:RECOMMENDS {rating: 5}]->(b);

```

a) List the books of type “text”

```

=> MATCH (b:Book)
WHERE b.type = "text"
RETURN b.title AS BookTitle;

```

b) List the name of student who bought a text and reference types books.

```

=> MATCH (s:Student)-[:BOUGHT]->(b:Book)
WHERE b.type IN ["text", "reference"]
WITH s.name AS StudentName, COLLECT(b.type) AS Types
WHERE "text" IN Types AND "reference" IN Types
RETURN StudentName;

```

c) List the most recommended book type.

```

=> MATCH (s:Student)-[r:RECOMMENDS]->(b:Book)
RETURN b.type AS BookType, COUNT(r) AS RecommendationCount
ORDER BY RecommendationCount DESC
LIMIT 1;

```

d) List the student who buy the more than one type of book

```

=> MATCH (s:Student)-[:BOUGHT]->(b:Book)
WITH s.name AS StudentName, COUNT(DISTINCT b.type) AS BookTypeCount
WHERE BookTypeCount > 1
RETURN StudentName;

```

** SLIP 25_Q1 : Write the HTML5 code for generating the form as shown below. Apply the internal CSS to the following form to set the font size, font color, heading , background color etc.

```

==> <!DOCTYPE html>

```

```

<html>
<head>
  <title>Entry Form</title>
  <style>
    body {
      background-color: white;
      font-family: sans-serif;
    }

    .container {
      width: 300px; /* Reduced width */
      margin: 50px auto;
      padding: 15px; /* Reduced padding */
      border: 1px solid #ccc;
      background-color: lightblue;
    }

```

```
h1 {
  text-align: center;
  color: red;
  font-size: 20px; /* Reduced font size */
}

label {
  display: inline-block;
  width: 120px; /* Reduced width */
  margin-bottom: 8px;
  color: red;
  font-size: 14px; /* Reduced font size */
}

input[type="text"],
input[type="password"],
textarea,
select {
  width: calc(100% - 130px); /* Adjusted width to fit within smaller form */
  padding: 8px; /* Reduced padding */
  margin-bottom: 8px;
  font-size: 14px; /* Reduced font size */
  border: 1px solid #ccc;
}

input[type="radio"],
input[type="checkbox"] {
  margin-right: 5px;
}

.form-group {
  display: flex;
  align-items: center;
  margin-bottom: 8px; /* Reduced margin */
}

.checkbox-group {
  margin-left: 120px; /* Adjusted margin */
}

button {
  background-color: grey;
  color: white;
  padding: 8px 16px; /* Reduced padding */
  font-size: 14px; /* Reduced font size */
  border: none;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}

.button-group {
  display: flex;
```

```
        justify-content: space-between;
    }
</style>
</head>
<body>
<div class="container">
    <h1>ENTRY FORM</h1>
    <form>
        <div class="form-group">
            <label for="name">Enter your Name:</label>
            <input type="text" id="name" name="name" required>
        </div>

        <div class="form-group">
            <label for="age">Enter your Age:</label>
            <input type="text" id="age" name="age" required>
        </div>

        <div class="form-group">
            <label for="address">Enter your Address:</label>
            <textarea id="address" name="address" rows="2" required></textarea>
        </div>

        <div class="form-group">
            <label for="sex">Sex:</label>
            <input type="radio" id="male" name="sex" value="male">
            <label for="male">Male</label>
            <input type="radio" id="female" name="sex" value="female">
            <label for="female">Female</label>
        </div>

        <div class="form-group">
            <label for="country">Select your Country:</label>
            <select id="country" name="country">
                <option value="usa">USA</option>
                <option value="canada">Canada</option>
                <option value="uk">UK</option>
            </select>
        </div>

        <div class="form-group">
            <label for="languages">Languages Known:</label>
            <div class="checkbox-group">
                <input type="checkbox" id="c" name="languages" value="C">
                <label for="c">C</label>
                <input type="checkbox" id="cpp" name="languages" value="C++">
                <label for="cpp">C++</label>
                <input type="checkbox" id="vb" name="languages" value="VB">
                <label for="vb">VB</label>
                <input type="checkbox" id="java" name="languages" value="JAVA">
                <label for="java">JAVA</label>
                <input type="checkbox" id="asp" name="languages" value="ASP">
                <label for="asp">ASP</label>
                <input type="checkbox" id="others" name="languages" value="Others">
                <label for="others">Others</label>
            </div>
        </div>
    </form>
</div>
</body>
</html>
```

```

</div>

<div class="form-group">
  <label for="password">Enter your Password:</label>
  <input type="password" id="password" name="password" required>
</div>

<div class="button-group">
  <button type="submit">Submit</button>
  <button type="reset">Reset</button>
</div>
</form>
</div>
</body>
</html>

```

** SLIP 25_Q2 : Model the following University information system as a graph model, and answer the following queries using Cypher. University has various departments like Physics, Geography, Computer etc. Each department conducts various courses and a course may be conducted by multiple departments. Every course may have recommendations provided by people.

==> // Create Departments

```

CREATE (d1:Department {name: "Physics"}),
      (d2:Department {name: "Geography"}),
      (d3:Department {name: "Computer"}),
      (d4:Department {name: "Mathematics"});

```

// Create Courses

```

CREATE (c1:Course {name: "Quantum Mechanics", description: "Advanced physics concepts"}),
      (c2:Course {name: "Geographic Information Systems", description: "GIS and mapping"}),
      (c3:Course {name: "Data Structures", description: "Foundational programming course"}),
      (c4:Course {name: "Linear Algebra", description: "Matrix and vector theories"}),
      (c5:Course {name: "Environmental Science", description: "Study of ecosystems"}),
      (c6:Course {name: "Algorithms", description: "Algorithmic problem-solving"});

```

// Link Departments to Courses

```

MATCH (d:Department), (c:Course)
WHERE (d.name = "Physics" AND c.name = "Quantum Mechanics") OR
      (d.name = "Geography" AND c.name = "Geographic Information Systems") OR
      (d.name = "Geography" AND c.name = "Environmental Science") OR
      (d.name = "Computer" AND c.name = "Data Structures") OR
      (d.name = "Computer" AND c.name = "Algorithms") OR
      (d.name = "Mathematics" AND c.name = "Linear Algebra") OR
      (d.name = "Mathematics" AND c.name = "Algorithms")
CREATE (d)-[:OFFERS]->(c);

```

// Create People and Recommendations

```

CREATE (p1:Person {name: "Alice", role: "Student"}),
      (p2:Person {name: "Bob", role: "Professor"}),
      (p3:Person {name: "Eve", role: "Student"});

```

// Create Recommendations

```

MATCH (p:Person), (c:Course)
WHERE (p.name = "Alice" AND c.name = "Quantum Mechanics") OR
      (p.name = "Bob" AND c.name = "Geographic Information Systems") OR
      (p.name = "Eve" AND c.name = "Algorithms")
CREATE (p)-[:RECOMMENDS {rating: 5, feedback: "Excellent course"}]->(c);

```

a) List the details of all the departments in the university.

=> MATCH (d:Department)

RETURN d.name AS DepartmentName;

b) List the names of the courses provided by Physics department.

=> MATCH (d:Department)-[:OFFERS]->(c:Course)

WHERE d.name = "Physics"

RETURN c.name AS CourseName;

c) List the most recommended course in Geography department.

=> MATCH (d:Department)-[:OFFERS]->(c:Course)<-[:RECOMMENDS]-(p:Person)

WHERE d.name = "Geography"

RETURN c.name AS CourseName, COUNT(r) AS RecommendationCount

ORDER BY RecommendationCount DESC

LIMIT 1;

d) List the names of common courses across Mathematics and computer department.

=> MATCH (d1:Department)-[:OFFERS]->(c:Course)<-[:OFFERS]-(d2:Department)

WHERE d1.name = "Mathematics" AND d2.name = "Computer"

RETURN c.name AS CommonCourseName;