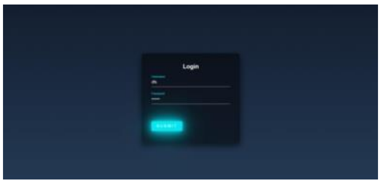
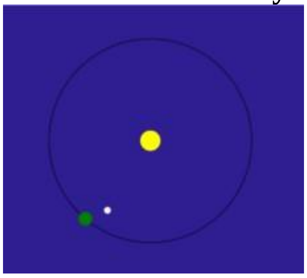
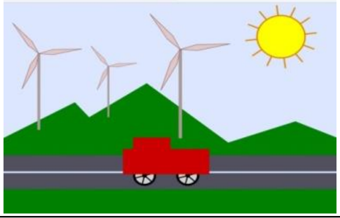
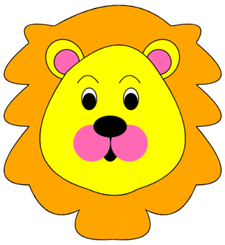
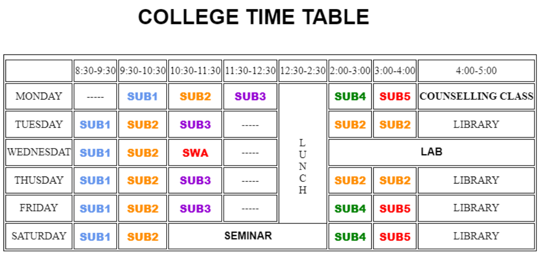
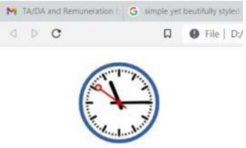
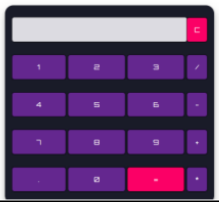
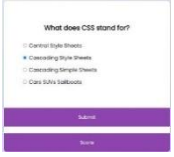
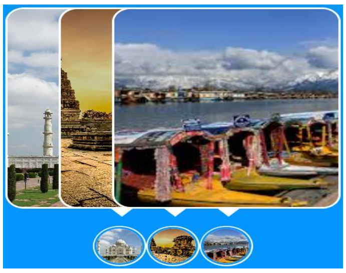
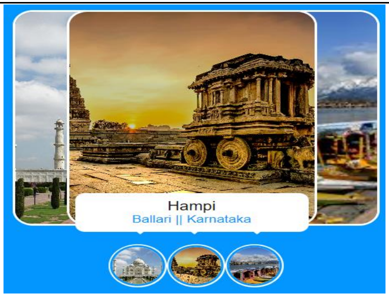
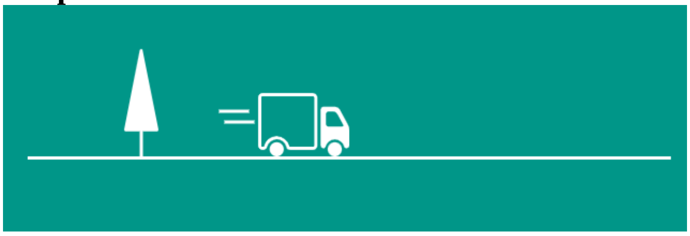
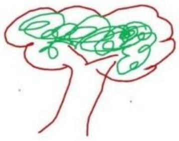
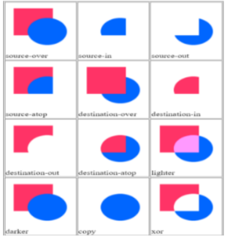
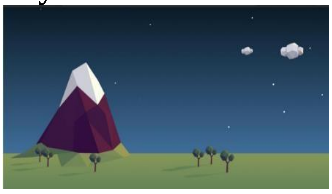
INDEX

	And while mouse is hovered on SUBMIT button it should look like  	**19-22**
5.	**Create a web page to model solar system using canvas element animation, where it contains sun, earth and moon (all must be created using canvas shapes not images). Earth should revolve around sun and moon should revolve around earth simultaneously. Sample screen shot below:** 	**23-25**
6.	**Create the following drawing in html page using only SVG** 	**26-28**
7.	**Create the following drawing using SVG** 	**29-30**
8.	**Create a web page using HTML and CSS to create a timetable as follows:** 	**31-37**

COLLEGE TIME TABLE

	8:30-9:30	9:30-10:30	10:30-11:30	11:30-12:30	12:30-2:30	2:00-3:00	3:00-4:00	4:00-5:00
MONDAY	-----	SUB1	SUB2	SUB3	L U N C H	SUB4	SUB5	COUNSELLING CLASS
TUESDAY	SUB1	SUB2	SUB3	-----		SUB2	SUB2	LIBRARY
WEDNESDAT	SUB1	SUB2	SWA	-----		LAB		
THUSDAY	SUB1	SUB2	SUB3	-----		SUB2	SUB2	LIBRARY
FRIDAY	SUB1	SUB2	SUB3	-----		SUB4	SUB5	LIBRARY
SATURDAY	SUB1	SUB2	SEMINAR			SUB4	SUB5	LIBRARY

SL NO.	PART-B	PAGE NO.
1.	**Create a web page using HTML5 canvas element to show a clock which changes time for every second, minute and hours (as that of an analog clock). Clock should have second, minute and hour needles and minute marking must be there (as shown in screen shot).**  TA/DA and Remuneration simple yet beutifully styled File | D:/	**39-42**
2.	**Create a web page containing simple calculator which should have basic arithmetic (+,-,*,/) operation on two floating point numbers and show result. Validations to be followed:** **• . (Decimal point) should be taken only once for an operand.** **• Operand can be negative.** **• Division by zero must be shown proper error message in result. Sample screen shot:**  	**43-47**
3.	**Create a HTML page make a quiz game where user should answer one question at a time, answers must be shown in radio buttons. Without submitting the answer, quiz should not move to next question (Minimum five questions must be there). When user wishes to get score (using score button) score should be displayed in alert 22 message. All the question must be loaded in same page (no page navigation is allowed) Sample screen shot:**  What does CSS stand for? ○ Central Style Sheets ● Cascading Style Sheets ○ Cascading Simple Sheets ○ Cars SUVs Sailboats 	**48-55**
4.	**Create a web page using HTML/CSS which contains cards (shown as a stack of cards) with image of a tourist place and below that is a thumbnail (shown in circle with image), when mouse hovers over thumbnail corresponding card comes in front and also small description about the tourist place will be displayed. All these must happen using css animation and transition.** **Initial interface:**  **Interface should look like below screenshot when mouse hovered on thumbnail:**	**56-61**

	 **(Java script should not be used to animate.)**	
5.	**Create a web page using HTML5/CSS3 to animate a truck movement. While truck moves on mountains and trees should move in the back ground.** **Output screen shot:**  **Background hills must be created using CSS only and for tree, truck and wheels download the images from the following URLs. https://s3-us-west-2.amazonaws.com/s.cdpn.io/130015/tree.svg https://s3-us-west-2.amazonaws.com/s.cdpn.io/130015/truck.svg https://s3-us-west-2.amazonaws.com/s.cdpn.io/130015/wheels.** **svg Animation must be implemented using ONLY CSS and Java script should not be used.**	**62-64**
6.	**Create a simple paint app which draws lines based on the selected colour (chosen using color input) with selected thickness (chosen using number input) and there must be CLEAR button to clear the canvas. Sample screen shot:** 	**65-69**
7.	**Create web page using HTML5 canvas element to illustrate all canvas composition. Output must exactly look like the following screenshot:**  source-over source-in source-out source-atop destination-over destination-in destination-out destination-atop lighter darker copy xor	**70-72**
8.	**Create a web page which must be as shown in below image using HTML5, SVG and CSS3. Here Mountain must be drawn using SVG, Stars in sky changes their position randomly for every time page is loaded (java script can be used). Sky and stars must be created using element.**  **Note: No Online (live access) CSS files must be used.**	**73-75**

PART A

/* ** */

PROGRAM:1 **DATE:**

NAME: **REG.NO:U05DP22S0**

/* ** */

Aim: Create a home page for a college website containg all latest HTML5 tags like <article>, <aside>, <nav>, <header>, <footer>, <section>, <figure>. And in <nav>. Create hyper links for courses, facilities and contact details. On clicking

• Course hyperlink, display the page with course names offered in the college using ordered list,

• Facilities hyperlink, display the page describing the facilities using unordered list

• Contact hyperlink, display the page to show phone number, email and address in separate columns with respective headings.

/**/

Homepage.html

```
<html>

<head><title>GFGC carstreet</title></head>

<body bgcolor="skyblue">

<center><header><h6 style="color blue">Aflitated to Mangalore university</h6></header></center>

<figure><figcaption align="center"><img src="F:\Drive
Info\OneDrive\Pictures\Seal_of_Karnataka.svg.png"width="100" height="100"></figure>

<center><h2 style+"color red"><strong>DR.P.DAYANANDA PAI-P.SATHISHA PAI

</h2></strong><br>GOVERNMENT FIRST GRADE COLLEGE, MANGALORE

<h3>Carstreet Mangaluru-575001</h3></center>

<article>

<center>

<nav>

<table>

<tr>

<td><a href="F:\CMA programs\Program1\Course.html">Course</a>|</td>

<td><a href="F:\CMA programs\Program1\Facilities.html">Facility</a>|</td>

<td><a href="F:\CMA programs\Program1\Contact.html">contact us</a>|</td>

<tr>

</table>

</nav>

</center>
```

```
</article>
<aside>
<section>
<center><img src="F:\Drive Info\OneDrive\Pictures\GFGCc.jpeg" width="400" height="200">
</section>
</aside>
<footer align="center"><p><a href="http://dce.kar.nic.in ">Department of Collegeiate education |
dce.kar.nic.in</a></p>
</footer>
</body>
</html>
```

Contact.html

```
<html>
<head><title>contact</title></head>
<body bgcolor="grey">
<table style="width:100%">
<colgroup>
<col span="3" style="background-color:pink">
</colgroup>
<tr>
<th><font color="blue">ADDRESS</font></th>
<th><font color="blue">PHONE NUMBER</font></th>
<th><font color="blue">EMAIL</font></th>
</tr>
<tr>
<td width="40%">DR.P.DAYANANDA PAI-P.SATHISHA PAI<br>GOVERNMENT FIRST GRADE COLLEGE<br>
CARSTREET,MANGALORE.<br>
PIN-575001<br>
District-DAKSHINA KANNADA<br>
STATE-KARNATAKA<br>
Google map:<a href="https://maps.app.goo.gl/fDW5YqHKGZKMmceu9">Location</a>
</td>
```

```
<td>Phone:0824-2494109 / 2491073<br>
Mobile No:+91 9874561220
</td>
<td>Email:gfgcmangalorecarstreet@rocketmail.com</td>
</tr>
</table>
<br>
<a href="D:\shamith_chethan_2bca'a'\homepage.html">HOMEPAGE</a>
</body>
</html>
```

Facility.html

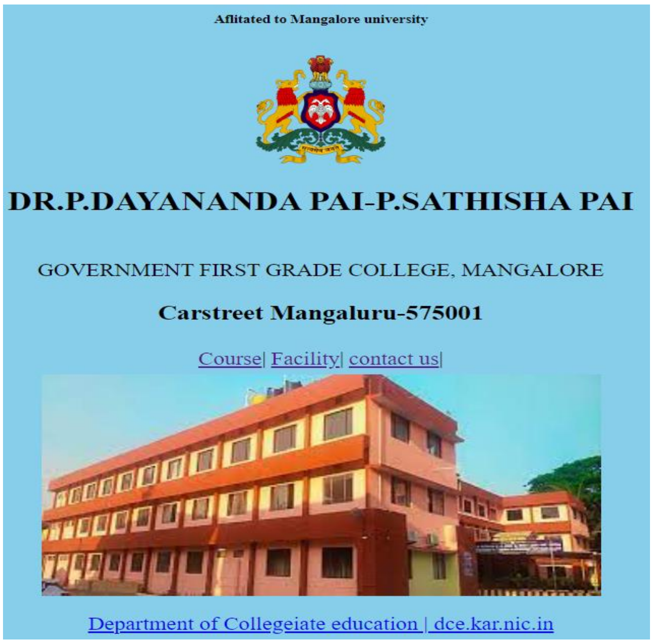
```
<html>
<head><title>Facilities</title></head>
<body bgcolor="lightgreen">
<h2><font color="Blue">Facilities</font></h2>
<hr>
<p>The college has 1.67 acres of prime land in the heart of the city, on which the following facilities are available.</p>
<ul>
<li>Well furnished Office with the required facilities</li>
<li>CCTV Surveillance</li>
<li>Gym facility</li>
<li>Student Help Desk</li>
<li>Computer lab with 60 computers with high speed internet connection.</li>
<li>Parking area.</li>
</ul>
<br>
<a href="D:\shamith_chethan_2bca'a'\homepage.html">HOMEPAGE</a>
</body>
</html>
```

Course.html

```
<html>
<head><title>course</title></head>
```

```
<body bgcolor="yellow">

<h1>Courses</h1>

<h3>PG and UG Course</h3>

<ol>

<li>MA</li>

<li>MCOM</li>

<li>MSW</li>

<li>BCA</li>

<li>BCOM</li>

</ol>

<br>

<a href="D:\shamith_chethan_2bca'a'\homepage.html">HOMEPAGE</a>

</body>

</html>
```

OUTPUT:



Aflitated to Mangalore university

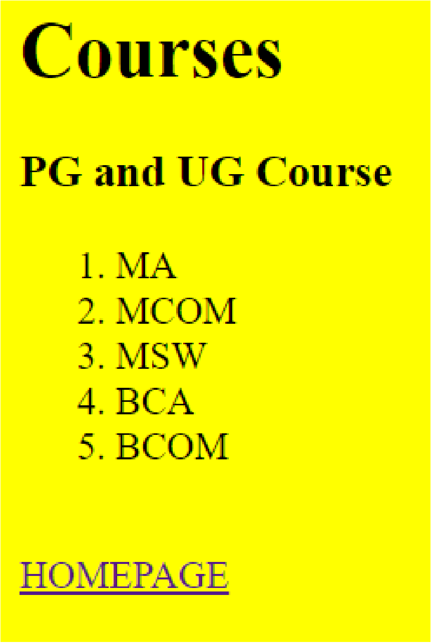
DR.P.DAYANANDA PAI-P.SATHISHA PAI

GOVERNMENT FIRST GRADE COLLEGE, MANGALORE

Carstreet Mangaluru-575001

Course| Facility| contact us|

Department of Collegeiate education | dce.kar.nic.in

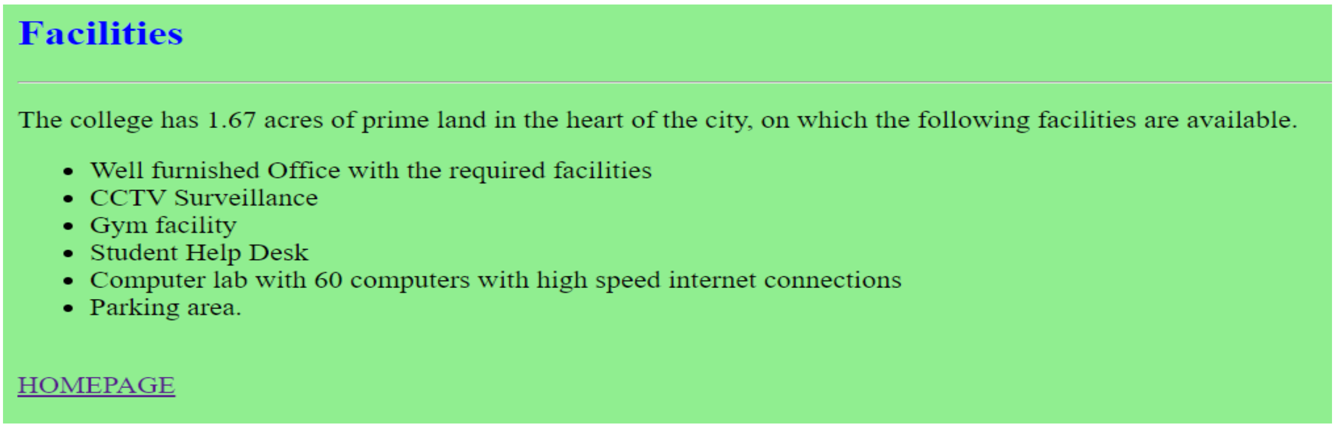


Courses

PG and UG Course

1. MA
2. MCOM
3. MSW
4. BCA
5. BCOM

HOMEPAGE

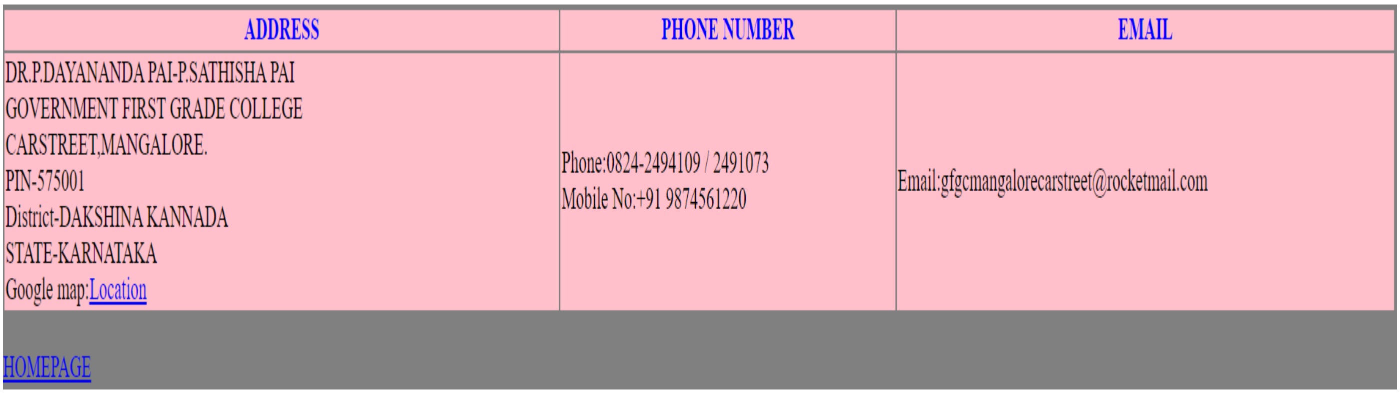


Facilities

The college has 1.67 acres of prime land in the heart of the city, on which the following facilities are available.

- Well furnished Office with the required facilities
- CCTV Surveillance
- Gym facility
- Student Help Desk
- Computer lab with 60 computers with high speed internet connections
- Parking area.

HOMEPAGE



ADDRESS	PHONE NUMBER	EMAIL
DR.P.DAYANANDA PAI-P.SATHISHA PAI GOVERNMENT FIRST GRADE COLLEGE CARSTREET,MANGALORE. PIN-575001 District-DAKSHINA KANNADA STATE-KARNATAKA Google map:Location	Phone:0824-2494109 / 2491073 Mobile No:+91 9874561220	Email:gfgcmangalorecarstreet@rocketmail.com

HOMEPAGE

/* ** */

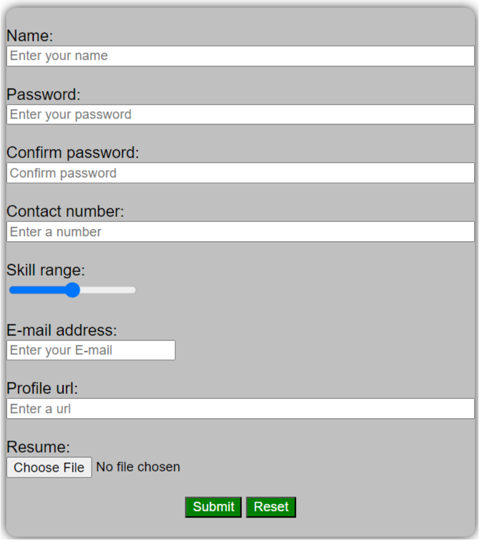
PROGRAM:2 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Design a HTML5 web page containing form with text, password, number, range, email, url, file, submit and reset elements which must be styled using CSS3 according to following screen shot.

REGISTRATION FORM



Name:
Enter your name

Password:
Enter your password

Confirm password:
Confirm password

Contact number:
Enter a number

Skill range:

E-mail address:
Enter your E-mail

Profile url:
Enter a url

Resume:
Choose File No file chosen

Submit Reset

Note that:

- **Apply the style exactly same as shown in the above screen shot (with border radius, box shadow and colours).**
- **Submit and Reset buttons must change their colour on mouse hovering.**
- **Name and passwords should not be empty. If empty, provide error message when submit is clicked .**
- **When clicked on submit button email, Profile URLs must be validated for proper input.**
- **Contact number must contain only 10 digits not lesser and not more.**
- **Clicking on Reset button must clear all fields' entry.**

/***/

PartA2.html

```
<!DOCTYPE html>

<html>

    <style>

        form

        {

            margin:0px auto;

            width:fit-content;

            font-family:sans-serif;
            border:1px grey;
            border-radius:10px;
            box-shadow:rgba(0,0,0,0.8)0 0 10px;
            background-color:silver;
        }
        .button:hover
        {
            background-color:green;
            color:black;
            cursor: pointer;
        }
        </style>
        <body>
            <h1 style="text-align:center;color:green">REGISTRATION FORM</h1>
            <form action= "Outputpage.html" method="post">
                <br>
                Name:<br>
                <input type="text" size="65" name="Name" required placeholder="Enter your name"/><br><br>
                Password:<br>
                <input type="password" size="65"  id="password"name="Password" required placeholder="Enter your password"/><br><br>
                Confirm password:<br>
                <input type="password" size="65"  id="confirm_password" name="Confirm_Password" required placeholder="Confirm password"/><br><br>
                Contact number:<br>
                <input type="tel" size="65" name="Contact number" pattern="[0-9]{10}" required placeholder="Enter a number"/><br><br>
                Skill range:<br>
                <input type="range" name="Skil range" width="100px" required/><br><br>
                E-mail address:<br>
                <input type="email" name="E-mail" required placeholder="Enter your E-mail"/><br><br>
                Profile url:<br>
                <input type="url" size="65"  name="Profile" required placeholder="Enter a url"/><br><br>
                Resume:<br>
```

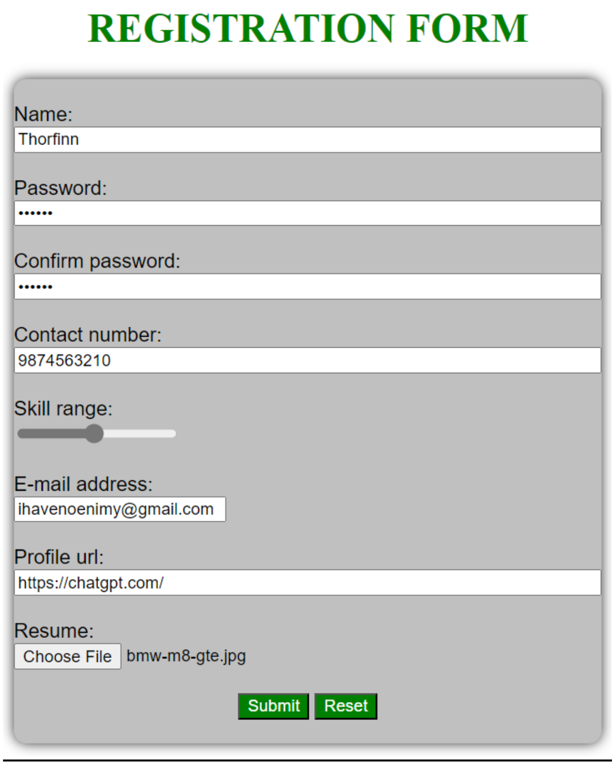
```
            <input type="file" id="Resume"name="resume" required/><br>
            <center>
               <br>
               <input type="submit" class="button" value="Submit" name="Submit" style="backgroundcolor:
green;color:white" />
               <input type="reset" class="button" value="Reset" name="Reset" style="background-color:
green;color:white" />
               <br>
            </center><br>
               </form>
               <script>
                  varpassword=document.getElementById("password"),confirm_password=document.getElem
entById("confirm_password");
                  function validatePassword()
                  {
                     if(password.value!=confirm_password.value)
                     {
                        confirm_password.setCustomValidity("Password dont match");
                     }
                     else
                     {
                        confirm_password.setCustomValidity("");
                     }
                  }
                  password.onchange=validatePassword;
                  confirm_password.onkeyup=validatePassword;
               </script>
      </body>
</html>
```

OUTPUTPAGE.html

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

```
        <h1>Seccesfully registered</h1>

        <h2>Thank you</h2>

    </body>

</html>
```

OUTPUT:



REGISTRATION FORM

Name:
Thorfinn

Password:
••••••

Confirm password:
••••••

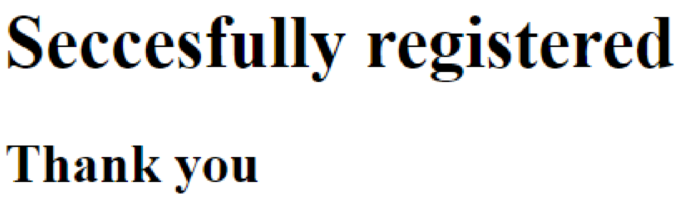
Contact number:
9874563210

Skill range:

E-mail address:
ihavenoenimy@gmail.com

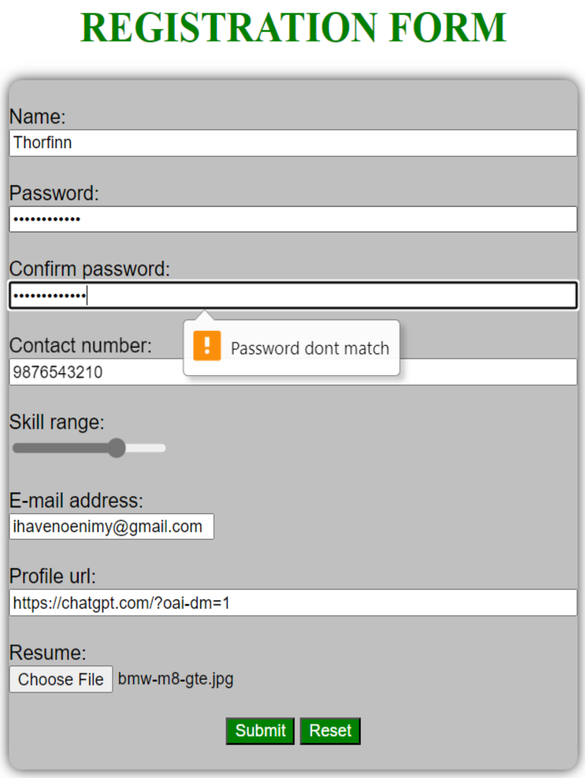
Profile url:
https://chatgpt.com/

Resume:
Choose File bmw-m8-gte.jpg



Seccesfully registered

Thank you



REGISTRATION FORM

Name:
Thorfinn

Password:
••••••••••••

Confirm password:
•••••••••••••

Password dont match

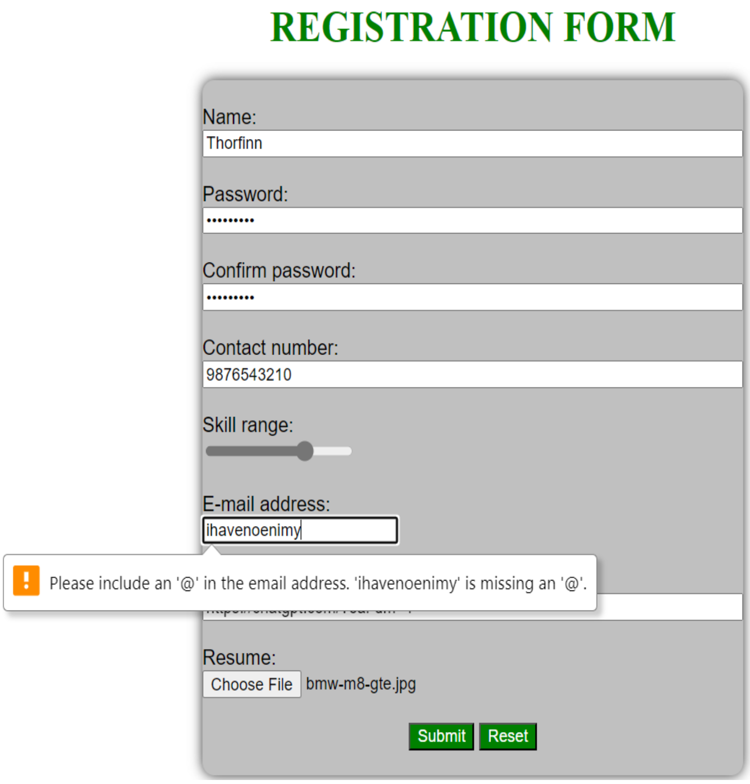
Contact number:
9876543210

Skill range:

E-mail address:
ihavenoenimy@gmail.com

Profile url:
https://chatgpt.com/?oai-dm=1

Resume:
Choose File bmw-m8-gte.jpg



REGISTRATION FORM

Name:
Thorfinn

Password:
•••••••••

Confirm password:
•••••••••

Contact number:
9876543210

Skill range:

E-mail address:
ihavenoenimy

Please include an '@' in the email address. 'ihavenoenimy' is missing an '@'.

Resume:
Choose File bmw-m8-gte.jpg

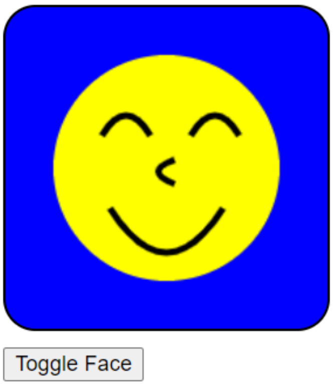
/***/

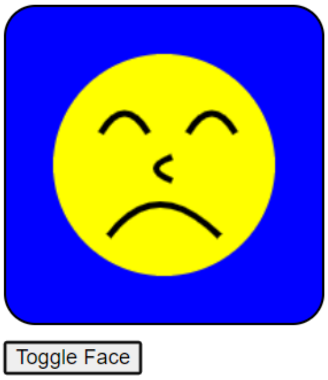
PROGRAM:3 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Create an HTML5 web page which shows a smiling face initially. On every click of 'Toggle Face' button display should toggle between smiling face and sad face. Note: Use only one button. And faces should be drawn using canvas element. Faces must be exactly like the following screen shots.





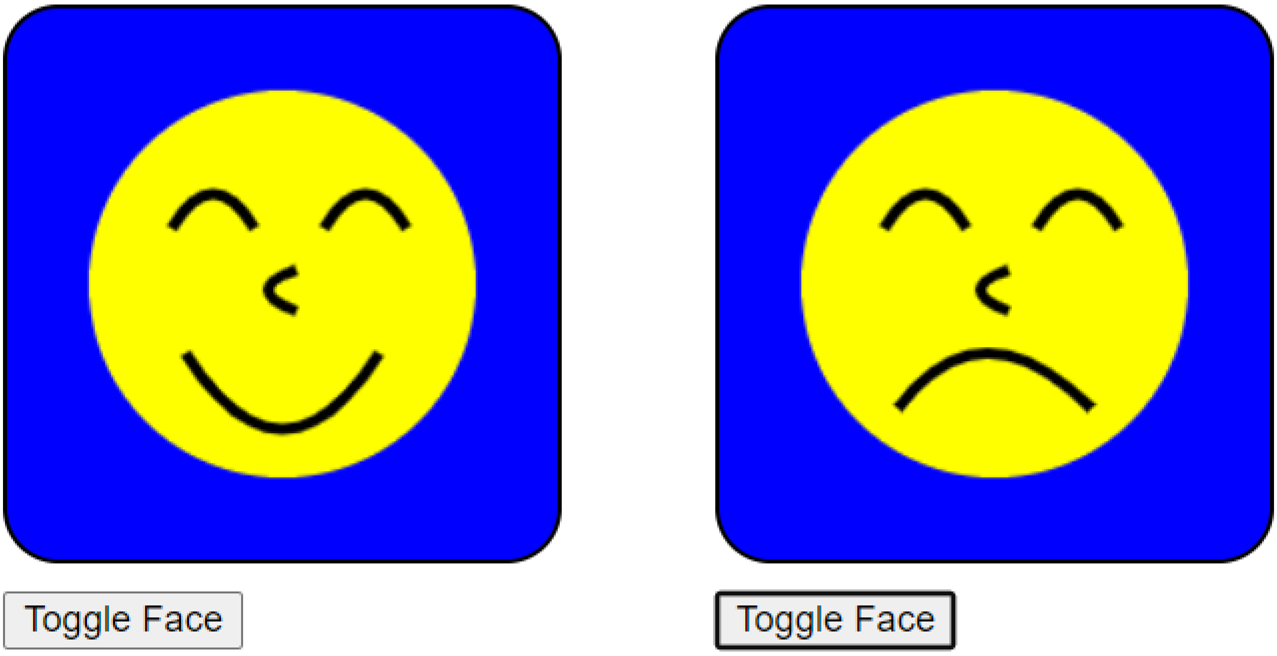
/***/

PartA3.html

```
<!DOCTYPE html>
<html>
  <head><title>Toggle Face</title>></head>
  <style>
    #canvas-container
      {
        border: 2px solid black;
        width:200px;
        height: 200px;
        display:flex;
        justify-content: center;
        align-items: center;
        border-radius: 20px;
        background-color: blue;
        margin-bottom:10px;
      }
      </style>
      <body>
        <div id="canvas-container">
        <canvas id="faceCanvas" width="150" height="150"></canvas>
      </div>
      <button onclick="toggleFace()">Toggle Face</button>
      <script>
        var canvas=document.getElementById("faceCanvas");
        var ctx=canvas.getContext("2d");
        var isSmiling=true;
        //Function to draw a smiling face
        function drawSmile()
        {
          //draw lips
          ctx.beginPath();
          ctx.moveTo(40,100);
          ctx.quadraticCurveTo(75,155,110,100);
          ctx.strokeStyle="#000";
          ctx.lineWidth=4;
          ctx.stroke();
        }
        function face()
        {
          //draw circle
          ctx.clearRect(0,0,canvas.width,canvas.height);
          ctx.beginPath();
          ctx.arc(75,75,70,0,Math.PI*2);
          ctx.fillStyle="yellow";
          ctx.fill();
          //draw eyes
          ctx.beginPath();
          ctx.moveTo(35,55);
          ctx.quadraticCurveTo(50,30,65,55);
          ctx.strokeStyle="#000";
          ctx.lineWidth=4;
          ctx.stroke();
          ctx.beginPath();
          ctx.moveTo(90,55);
          ctx.quadraticCurveTo(105,30,120,55);
          ctx.strokeStyle="#000";
          ctx.lineWidth=4;
          ctx.stroke();
          //draw nose
          ctx.beginPath();
          ctx.moveTo(80,70);
          ctx.quadraticCurveTo(60,77,80,85);
          ctx.strokeStyle="#000";
          ctx.lineWidth=4;
          ctx.stroke();
        }
        //Function to draw a sad face
        function drawSad()
        {
          //draw lips
          ctx.beginPath();
          ctx.moveTo(40,120);
          ctx.quadraticCurveTo(70,80,110,120);
          ctx.strokeStyle="#000";
          ctx.lineWidth=4;
          ctx.stroke();
        }
        //function to toggle between the smiling and sad face
        function toggleFace()
        {
          if(isSmiling)
          {
              face();
              drawSad();
```

```
                    isSmiling=false;
                }
                else
                {
                    face();
                    drawSmile();
                    isSmiling=true;
                }
            }
            //initial drawing of the smiling face
            face();
            drawSmile();
        </script>
    </body>
</html>
```

OUTPUT:



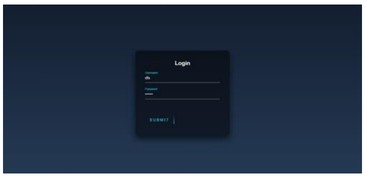
/* ** */

PROGRAM:4 **DATE:**

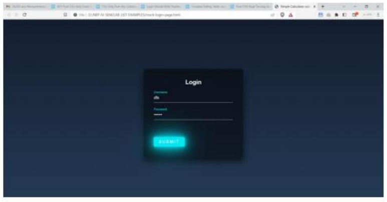
NAME: **REG.NO:U05DP22S0**

/*/**

Aim:4. Design a mock login page and style it using CSS3. Initially login page should look like the following screen shot



And while mouse is hovered on SUBMIT button it should look like



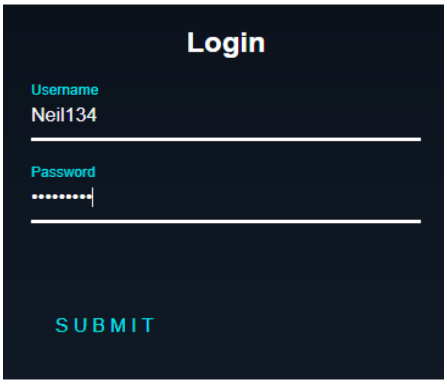
/***/

PartA4.html

```
<html>
<head>
    <title>Login Page</title>
    <style>
        body{
            font-family: sans-serif;
            background: linear-gradient(#141e30,#243b55);
            }
            .login-box
            {
                position: absolute;
                top: 50%;
                left:50%;
                width: 400px;
                padding: 40px;
                transform: translate(-50%,-50%);
                background: rgba(0,0,0,.5);
                box-sizing: border-box;
                box-shadow: 0 15px 25px rgba(0,0,0,.6);
                border-radius: 10px;
            }
            .login-box h2
            {
                margin: 0 0 30px;
                padding: 0;
                color: #fff;
                text-align: center;
            }
            .login-box .user-box
            {
                position: relative;
            }
            .login-box .user-box input
            {
                width: 100%;
                padding: 10px 0;
                font-size: 16px;
                color: #fff;
                margin-bottom: 30px;
                outline: none;
                background: transparent;
                border:none;
                padding-bottom: 1px solid #fff;
                border-bottom: 3px solid white;

            }
            .login-box .user-box label
            {
                position: absolute;
                top: 0;
                left: 0;
                padding: 10px 0;
                font-size: 16px;
                color: #fff;
            }
            .login-box .user-box input:focus~label,
            .login-box .user-box input:valid~label
            {
                top: -20px;
                left: 0;
                color: #03e9f4;
                font-size: 12px;
            }
            .login-box form a
            {
                position: relative;
                display:inline-block ;
                padding: 10px 20px;
                color: #03e9f4;
                font-size: 16px;
                text-decoration: none;
                margin-top: 40px;
                letter-spacing: 4px;
            }
            .login-box a:hover
            {
                background: #03e9f4;
                color: #fff;
                border-radius: 5px;
                box-shadow: 0 0 5px #03e9f4,
                            0 0 25px #03e9f4,
                            0 0 50px #03e9f4,
                            0 0 100px #03e9f4;
            }
```

```
    </style>
</head>
<body>
    <div class="login-box">
        <h2>Login</h2>
        <form>
            <div class="user-box">
                <input type="text" name="" required="">
                <label>Username</label>
            </div>
            <div class="user-box">
                <input type="password" name="" required="">
                <label>Password</label>
            </div>
            <a href="#">
                SUBMIT
            </a>
        </form>
    </div>
</body>
</html>
```

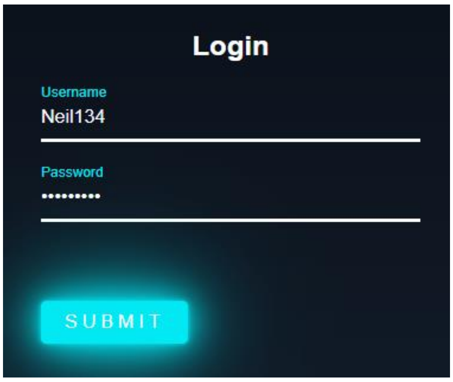
OUTPUT:



Login

Username
Neil134

Password
•••••••••



Login

Username
Neil134

Password
•••••••••

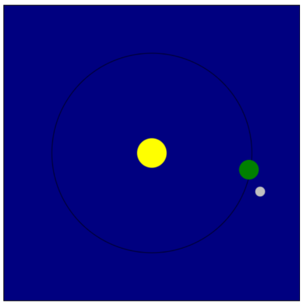
/* ** */

PROGRAM:5 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Create a web page to model solar system using canvas element animation, where it contains sun, earth and moon (all must be created using canvas shapes not images). Earth should revolve around sun and moon should revolve around earth simultaneously.



/***/

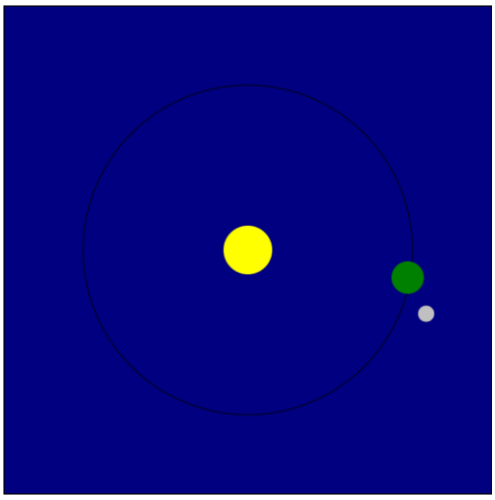
PartA5.html

```
<!DOCTYPE html>

<html>
    <head>
        <title>Solar System</title>
        <style>
            canvas
            {
                border:2px solid black;
                background-color:navy;
           }
        </style>
    </head>
    <body>
        <canvas id="canvas" width="600" height="600">
        </canvas>
        <script>
            const canvas=document.getElementById('canvas');
            const ctx=canvas.getContext('2d');
            //Define the sun,earth and moon
```

```
const sun={
    x:canvas.width/2,
    y:canvas.height/2,
    radius:50,
    color:'yellow'
};
const earth={
    x:sun.x+200,
    y:sun.y,
    radius:20,
    color:'green',
    angle:0,
    speed:0.0007
};
 const moon={
    x:earth.x+50,
    y:earth.y,
    radius:10,
    color:'silver',
    angle:0,
    speed:0.0045
};
function draw()
{
    ctx.clearRect(0,0,canvas.width,canvas.height);
    ctx.beginPath();
    ctx.arc(sun.x,sun.y,203,0,Math.PI*2,false);
    ctx.stroke();
    //Draw the sun
    ctx.beginPath();
    ctx.arc(sun.x,sun.y,30,0,Math.PI*2);
    ctx.fillStyle=sun.color;
    ctx.fill();
    //Draw the earth
```

```
            ctx.beginPath();
            ctx.arc(earth.x,earth.y,earth.radius,0,Math.PI*2);
            ctx.fillStyle=earth.color;
            ctx.fill();
             //Draw the moon
             ctx.beginPath();
            ctx.arc(moon.x,moon.y,moon.radius,0,Math.PI*2);
            ctx.fillStyle=moon.color;
            ctx.fill();
            //Update the earth's position
            earth.x=sun.x+200*Math.cos(earth.angle);
            earth.y=sun.y+200*Math.sin(earth.angle);
            earth.angle+=earth.speed;
             //Update the moon's position
             moon.x=earth.x+50*Math.cos(moon.angle);
            moon.y=earth.y+50*Math.sin(moon.angle);
            moon.angle+=moon.speed;
            //Request another animation frame
            requestAnimationFrame(draw);
        }
        //Start the animation loop
        draw();
      </script>
   </body>
</html>
```

OUTPUT:



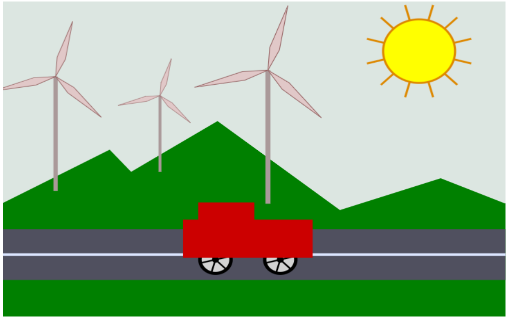
/* *** */

PROGRAM:6 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Create the following drawing in html page using only SVG.



/***/

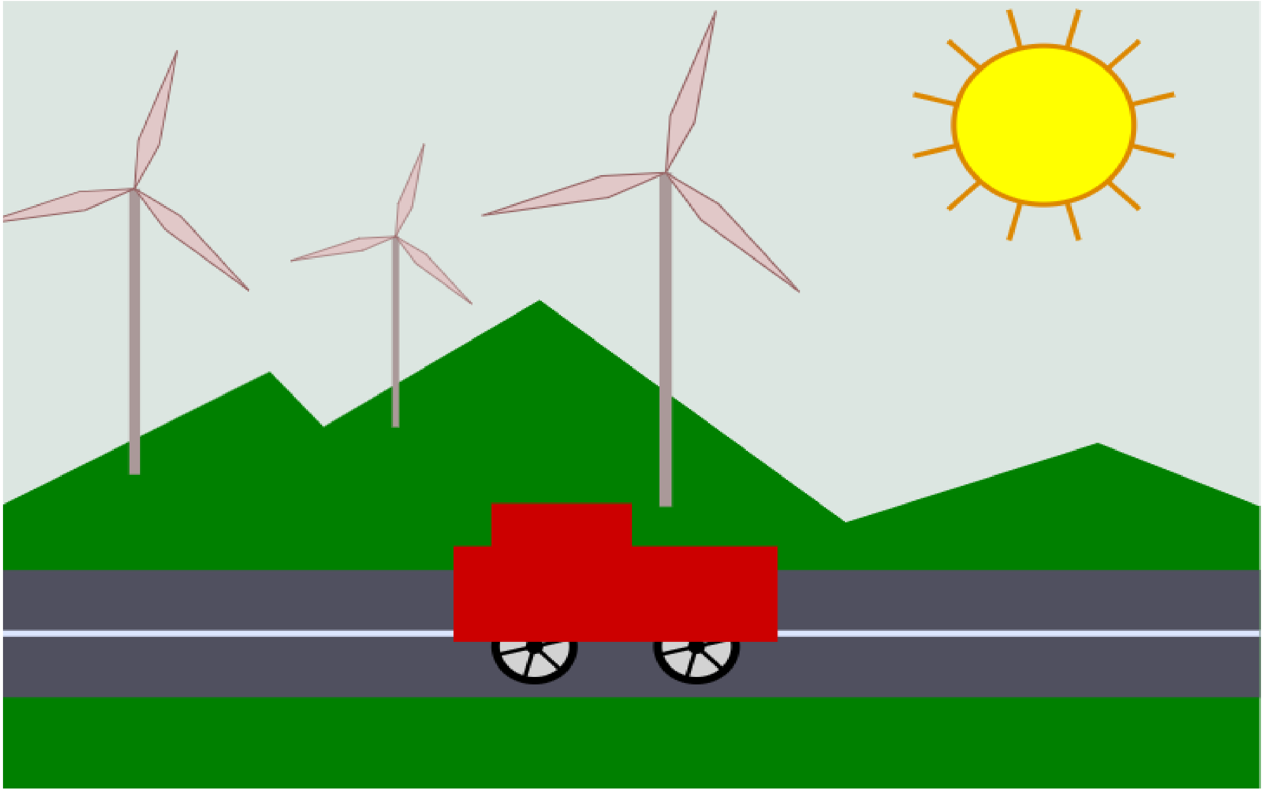
PartA6.html

```
<svg version="1.1"
width="14cm" height="10cm"
viewBox="0 0 7 5">
<defs>
   <g id="wheel" transform="rotate(0)">                    <!--A rotating wheel for the cart-->
  <animateTransform attributeName="transform" attributeType="XML" type="rotate" from="360" to="0"
dur="0.5s" fill="remove" repeatCount="indefinite"/>
   <circle cx="0" cy="0" r="1" fill="black"/>
   <circle cx="0" cy="0" r="0.8" fill="lightGray"/>
   <rect x="-0.9" y="-0.05" width="1.8" height=".1" fill="black"/>
   <rect x="-0.9" y="-0.05" width="1.8" height=".1" fill="black" transform="rotate(120)"/>
   <rect x="-0.9" y="-0.05" width="1.8" height=".1" fill="black" transform="rotate(240)"/>
   <circle cx="0" cy="0" r="0.2" fill="black"/>
   </g>
   <g id="cart" transform="translate(2.5,0)">          <!--A cart made from two wheels and two rects-->
   <animateTransform attributeName="transform" attributeType="XML" type="translate" from="-3,-0.05"
to="11,-0.05" dur="1s" fill="remove" repeatCount="indefinite"/>
   <g transform="scale(0.3,0.3)">
   <use xlink:href="#wheel" transform="translate(-1.5,-0.1) scale(0.8,0.8)"/>
   <use xlink:href="#wheel" transform="translate(1.5,-0.1) scale(0.8,0.8)"/>
   <rect x="-3" y="0" width="6" height="2" fill="#CC0000"/>
   <rect x="-2.3" y="1.9" width="2.6" height="1" fill="#CC0000"/>
   </g>
```

```
    </g>
    <g id="sun" fill="yellow" stroke="#DD8800" stroke-width="0.03"   <!--rotating sun-->
    <animateTransform attributeName="transform" attributeType="XML" type="rotate" from="0" to="360"
dur="0.5s" fill="remove" repeatCount="indefinite"/>
    <line x1="-0.75" y1="0" x2="0.75" y2="0"/>
    <line x1="-0.75" y1="0" x2="0.75" y2="0" transform="rotate(30)"/>
    <line x1="-0.75" y1="0" x2="0.75" y2="0" transform="rotate(60)"/>
    <line x1="-0.75" y1="0" x2="0.75" y2="0" transform="rotate(90)"/>
    <line x1="-0.75" y1="0" x2="0.75" y2="0" transform="rotate(120)"/>
    <line x1="-0.75" y1="0" x2="0.75" y2="0" transform="rotate(150)"/>
    <circle cx="0" cy="0" r="0.5"/>
    </g>
    <Polygon id="vane" points="0,0 0.5,0.1 1.5,0 0.5,-0.1" fill="rgb(225,200,200)"
stroke="rgb(150,100,100)" stroke-width="0.015"></Polygon>
    <g id="windmill">                                   <!--A windmill with a rotating set of three
vanes-->
    <rect x="-0.05" y="0" width="0.1" height="3" fill="#AA9999"/>
    <g transform="translate(0,3)">
    <g transform="rotate(0)">                                  <!--the set of three rotating vanes-->
    <animateTransform attributeName="transform" attributeType="XML" type="rotate" from="360" to="0"
dur="0.2s" fill="remove" repeatCount="indefinite"/>
    <use xlink:href="#vane"/>
    <use xlink:href="#vane" transform="rotate(120)"/>
    <use xlink:href="#vane" transform="rotate(240)"/>
    </g>
    </g>
    </g>
    </defs>
    <g transform="translate(0,4) scale(1,-1)">
    <rect x="0" y="-1" width="7" height="5" fill="rgb(220,230,225)"/>          <!--sky as background-->
    <Polygon points="0,-1 0,0.8 1.5,1.65 1.8,1.3 3,2.1 4.7,0.7,6.1,1.2,7,0.8,7,-1" fill="green"/>
<-Mountains-->
    <rect x="0" y="-0.4" width="7" height="0.8" fill="rgb(80,80,95)"/>          <!--Road-->
    <rect x="0" y="-0.02" width="7" height="0.04" fill="rgb(220,230,255)"/>          <!--Stripe in road-->
    <use xlink:href="#sun" transform="translate(5.8,3.2)"/>          <!--Sun-->
    <use xlink:href="#windmill" transform="translate(0.75,1) scale(0.6,0.6)"/>          <!--Three Windmill-->
```

```
<use xlink:href="#windmill" transform="translate(2.2,1.3) scale(0.4,0.4)"/>

<use xlink:href="#windmill" transform="translate(3.7,0.8) scale(0.7,0.7)"/>

<use xlink:href="#cart"/>                                    <!--Cart-->

</g>

</svg>
```

OUTPUT:



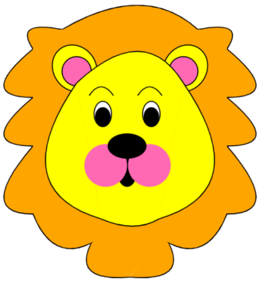
/* ** */

PROGRAM:7 **DATE:**

NAME: **REG.NO:U05DP22S0**

/**/

Aim: Create the following drawing using SVG.

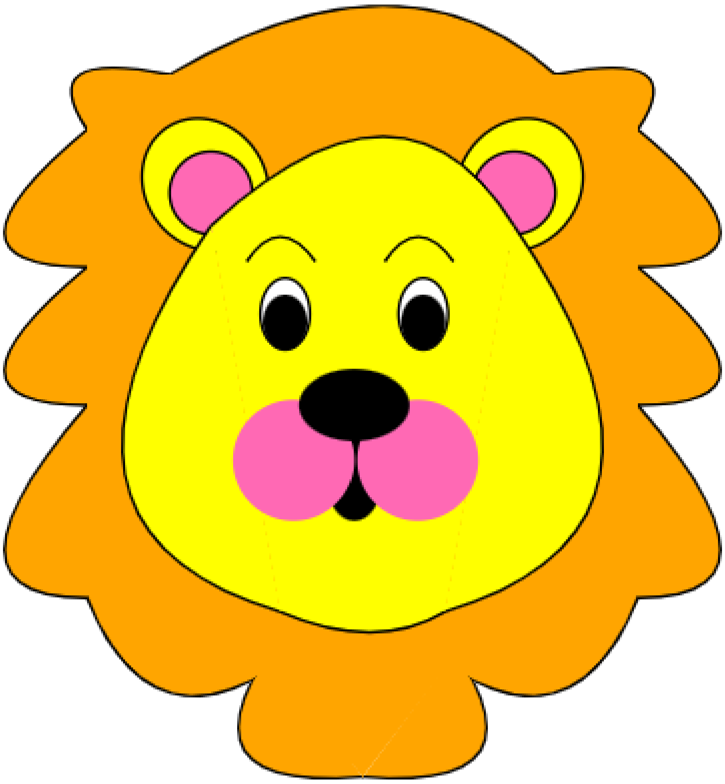


/**/

PartA7.html

```
<!DOCTYPE html>
<html>
<head>
  <title>SVG Lion</title>
</head>
<body>
  <svg height="800" width="960" xmlns="http://www.w3.org/2000/svg">
     <path d="M 150 80 Q 235 30 290 80" stroke="black" fill="orange" />
    <g id="fur">
   <path d="M 120 100 Q 100 70 150 80" stroke="black" fill="orange"/>
      <path d="M 120 100 Q 60 150 120 150" stroke="black" fill="orange" />
      <path d="M 120 150 Q 60 200 120 200" stroke="black" fill="orange" />
      <path d="M 120 200
               Q 60 270 120 270
               Q 140 320 180 300
               Q 160 340 220 335
               " stroke="black" fill="orange" />
    </g>
    <use xlink:href="#fur" transform="translate(440,0) scale(-1,1)" />
    <rect x="120" y="80" width="200" height="150" fill="orange"></rect>
    <polygon points="130,230 220,335 300,230" fill="orange" />
    <g id="ear">
      <ellipse cx="163" cy="120" rx="22" ry="25" stroke="black" fill="yellow" transform="rotate(-
35,163,120)" />
      <circle cx="165" cy="123" r="15" stroke="black" fill="hotpink" />
    </g>

    <use xlink:href="#ear" transform="translate(440,0) scale(-1,1)" />
    <g id="face">
     <path d="M 165 135 Q 90 245 190 275" stroke="black" fill="yellow" />
    </g>
    <use xlink:href="#face" transform="translate(440,0) scale(-1,1)" />
    <polygon points="165,135 275, 135 250,275 190,275" fill="yellow" />
    <path d="M 165 135 Q 235 70 275 135" stroke="black" fill="yellow" />
    <path d="M 190 275 Q 225 290 250 275" stroke="black" fill="yellow" />

    <g id="eye">
      <path d="M 178 148 Q 190 130 203 148" stroke="black" fill="transparent" />
      <ellipse cx="192" cy="167" rx="9" ry="13" stroke="black" fill="white" />
      <ellipse cx="192" cy="170" rx="8" ry="10" />
    </g>

    <use xlink:href="#eye" transform="translate(50,0)" />
    <ellipse cx="217" cy="227" rx="10" ry="15" fill="black" />
    <circle cx="195" cy="220" r="22" fill="hotpink" />
    <circle cx="240" cy="220" r="22" fill="hotpink" />
    <ellipse cx="217" cy="200" rx="20" ry="13" fill="black" />
</svg>
</body>
</html>
```

OUTPUT:



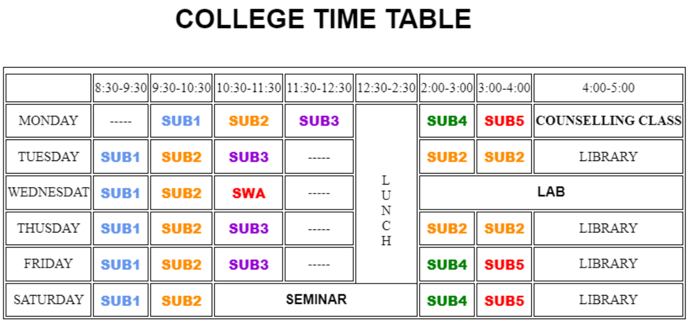
/* ** */

PROGRAM:8 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Create a web page using HTML and CSS to create a timetable as follows:



COLLEGE TIME TABLE

	8:30-9:30	9:30-10:30	10:30-11:30	11:30-12:30	12:30-2:30	2:00-3:00	3:00-4:00	4:00-5:00
MONDAY	-----	SUB1	SUB2	SUB3	LUNCH	SUB4	SUB5	COUNSELLING CLASS
TUESDAY	SUB1	SUB2	SUB3	-----		SUB2	SUB2	LIBRARY
WEDNESDAT	SUB1	SUB2	SWA	-----		LAB		
THUSDAY	SUB1	SUB2	SUB3	-----		SUB2	SUB2	LIBRARY
FRIDAY	SUB1	SUB2	SUB3	-----		SUB4	SUB5	LIBRARY
SATURDAY	SUB1	SUB2	SEMINAR			SUB4	SUB5	LIBRARY

/***/

PartA8.html

```
<html>

<head><title>TIME TABLE</title>

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

</head>

<body>

  <header>

    <h1> COLLEGE TIME TABLE</h1>

  </header>

  <table>

    <th>

      <tr>

        <td id="SUB1"></td>

        <td id="SUB">8:30-9:30</td>

        <td id="SUB">9:30-10:30</td>

        <td id="SUB">10:30-11:30</td>

        <td id="SUB">11:30-12:30</td>

        <td id="SUB">12:30-2:30</td>

        <td id="SUB">2:00-3:00</td>

        <td id="SUB">3:00-4:00</td>

        <td id="SUB">4:00-5:00</td>
      </tr>
    </th>
    <tr>
      <td id="SUB">MONDAY</td>
      <td id="blank">-----</td>
      <td id="box1">SUB1</td>
      <td id="box2">SUB2</td>
      <td id="box3">SUB3</td>
      <td rowspan="6">L<br>U<br>N<br>C<br>H</td>
      <td id="box4">SUB4</td>
      <td id="box5">SUB5</td>
      <td id="S"><b>COUNSELLING CLASS</b></td>
    </tr>
    <tr>
      <td id="SUB">TUESDAY</td>
      <td id="box1">SUB1</td>
      <td id="box2">SUB2</td>
      <td id="box3">SUB3</td>
      <td id="blank">-----</td>
      <td id="box2">SUB2</td>
      <td id="box2">SUB2</td>
      <td id="SUB2">LIBRARY</td>
    </tr>
    <tr>
      <td id="SUB">WEDNESDAT</td>
      <td id="box1">SUB1</td>
      <td id="box2">SUB2</td>
      <td id="box5">SWA</td>
      <td id="blank">-----</td>
      <td colspan="3" id="SUB3"><b>LAB</b></td>
    </tr>
    <tr>
      <td id="SUB">THUSDAY</td>
```

```
            <td id="box1">SUB1</td>

            <td id="box2">SUB2</td>

            <td id="box3">SUB3</td>

            <td id="blank">-----</td>

            <td id="box2">SUB2</td>

            <td id="box2">SUB2</td>

            <td id="SUB2">LIBRARY</td>

        </tr>

        <tr>

            <td id="SUB">FRIDAY</td>

            <td id="box1">SUB1</td>

            <td id="box2">SUB2</td>

            <td id="box3">SUB3</td>

            <td id="blank">-----</td>

            <td id="box4">SUB4</td>

            <td id="box5">SUB5</td>

            <td id="SUB2">LIBRARY</td>

        </tr>

            <td id="SUB">SATURDAY</td>

            <td id="box1">SUB1</td>

            <td id="box2">SUB2</td>

            <td colspan="3" id="SEMINAR"><b>SEMINAR</b></td>

            <td id="box4">SUB4</td>

            <td id="box5">SUB5</td>

            <td id="SUB2">LIBRARY</td>

    </table>

</body>

</html>
```

stylesheet.css

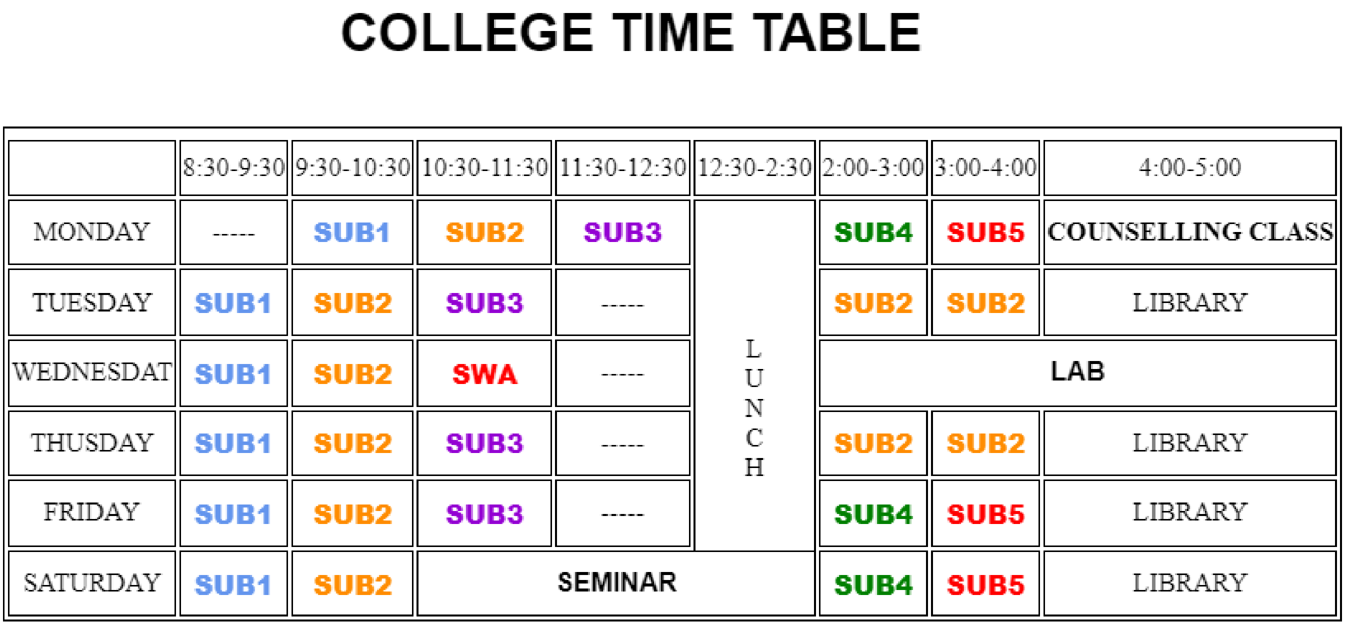
```
body

{

    background-color:#ffffff;

    mix-blend-mode:darken;
```

```
}
header h1
{
   margin-top:10%;
   text-align:center;
   color:black;
   font-weight:bold;
   font-family:sans-serif;
}
table
{
   margin-top:2%;
   width:max-content;
   border:1px solid black;
   height:300px;
   margin-left:30%;
}
td
{
   text-align:center;
   border:1px solid black;
}
#blank
{
   color:#000;
   background-color:#fff
}
#S
{
   font:weight 800;
   text-transform:uppercase;
   font-family:san-serif;
   background-color:#ffffff;
```

```
}
#SEMINAR
{
   font:weight 800;
   text-transform:uppercase;
   font-family:sans-serif;
   background-color: #ffffff;
}
#SUB
{
   color:black;
   font-weight:uppercase;
   font-family:san-serif;
   background-color: #ffffff;
}
#SUB1
{
   background-color:#fff;
}
#SUB2
{
   color:black;
   text-transform: uppercase;
   background-color: #ffffff;
}
#SUB3
{
   font-weight:800;
   font-family: sans-serif;
   background-color: #ffffff;
}
#box1
{
    font-weight:800;

    font-family: sans-serif;

    color:cornflowerblue;

}

#box2

{

     font-weight:800;

    font-family: sans-serif;

    color:darkorange;

}

#box3

{

    font-weight:800;

    font-family: sans-serif;

    color:darkviolet;

}

#box4

{

    font-weight:800;

    font-family: sans-serif;

    color:green;

}

#box5

{

    font-weight:800;

    font-family: sans-serif;

    color:red;

}

#box6

{

    font-weight:800;

    font-family: sans-serif;

    color:hotpink;
```

```
}
#box7
{
   font-weight:800;
   font-family: sans-serif;
   color:blue;
}
```

OUTPUT:



COLLEGE TIME TABLE

	8:30-9:30	9:30-10:30	10:30-11:30	11:30-12:30	12:30-2:30	2:00-3:00	3:00-4:00	4:00-5:00
MONDAY	-----	SUB1	SUB2	SUB3	L U N C H	SUB4	SUB5	COUNSELLING CLASS
TUESDAY	SUB1	SUB2	SUB3	-----		SUB2	SUB2	LIBRARY
WEDNESDAT	SUB1	SUB2	SWA	-----		LAB		
THUSDAY	SUB1	SUB2	SUB3	-----		SUB2	SUB2	LIBRARY
FRIDAY	SUB1	SUB2	SUB3	-----		SUB4	SUB5	LIBRARY
SATURDAY	SUB1	SUB2	SEMINAR			SUB4	SUB5	LIBRARY

PART B

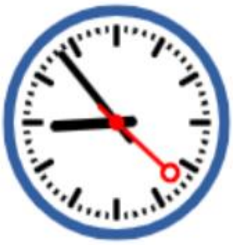
/* ** */

PROGRAM:1 **DATE:**

NAME: **REG.NO:U05DP22S0**

/**/

Aim: Create a web page using HTML5 canvas element to show a clock which changes time for every second, minute and hours (as that of an analog clock). Clock should have second, minute and hour needles and minute marking must be there (as shown in screen shot).



/**/

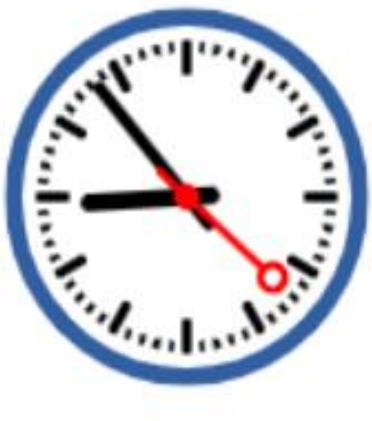
PartB1.html

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         canvas
         {
            background-color: white;
         }
      </style>
   </head>
         <body>
            <canvas id="canvas" width="150" height="150">The Current Time</canvas>
            <script>
               function clock()
               {
                  const now=new Date();
                  const canvas=document.getElementById("canvas");
                  const ctx=canvas.getContext("2d");
                  ctx.save();
                  ctx.clearRect(0,0,150,150);
                  ctx.translate(75,75);
                  ctx.scale(0.4,0.4);
                  ctx.rotate(-Math.PI/2);
                  ctx.strokeStyle="black"
                  ctx.fillStyle="white";
                  ctx.lineWidth=8;
                  ctx.lineCap="round";
                  //Hours marks
                  ctx.save();
                  for(let i=0;i<12;i++)
                  {
                     ctx.beginPath();
                     ctx.rotate(Math.PI/6);
                     ctx.moveTo(100,0);
                     ctx.lineTo(120,0);
                     ctx.stroke();
                  }
                  ctx.restore();
                   //Minute marks
                  ctx.save();
                  ctx.lineWidth=5;
                  for(let i=0;i<60;i++)
                  {
                     if(i%5!==0)
                     {
                        ctx.beginPath();
                        ctx.moveTo(117,0);
                        ctx.lineTo(120,0);
                        ctx.stroke();
                     }
                     ctx.rotate(Math.PI/30);
                  }
                  ctx.restore();
```

```
const sec=now.getSeconds();
const min=now.getMinutes();
const hr=now.getHours()%12;
ctx.fillStyle="black";
 //Write image description on
canvas.innerText='The Time is:${hr}:${min}';
//Write Hours
ctx.save();
ctx.rotate((Math.PI/6)*hr+(Math.PI/360)*min+(Math.PI/21600)*sec);
ctx.lineWidth=14;
ctx.beginPath()
ctx.moveTo(-20,0);
ctx.lineTo(80,0);
ctx.stroke();
ctx.restore();
//Write Minutes
ctx.save();
ctx.rotate((Math.PI/30)*min+(Math.PI/1800)*sec);
ctx.lineWidth=10;
ctx.beginPath()
ctx.moveTo(-28,0);
ctx.lineTo(112,0);
ctx.stroke();
ctx.restore();
 //Write seconds
 ctx.save();
ctx.rotate((sec*Math.PI)/30);
ctx.strokeStyle="red";
ctx.fillStyle="red";
ctx.lineWidth=6;
ctx.beginPath()
ctx.moveTo(-30,0);
ctx.lineTo(83,0);
```

```
                ctx.stroke();
                ctx.beginPath();
                ctx.arc(0,0,10,0,Math.PI*2,true);
                ctx.fill();
                ctx.beginPath();
                ctx.arc(95,0,10,0,Math.PI*2,true);
                ctx.stroke();
                ctx.fillStyle="rgba(0,0,0,0)";
                ctx.arc(0,0,3,0,Math.PI*2,true);
                ctx.fill();
                ctx.restore();
                ctx.beginPath();
                ctx.lineWidth=14;
                ctx.strokeStyle="#325FA2";
                ctx.arc(0,0,142,0,Math.PI*2,true);
                ctx.stroke();
                ctx.restore();
                window.requestAnimationFrame(clock);
            }
            window.requestAnimationFrame(clock);
        </script>
    </body>
</html>
```

OUTPUT:



/* *** */

PROGRAM:2 **DATE:**

NAME: **REG.NO:U05DP22S0**

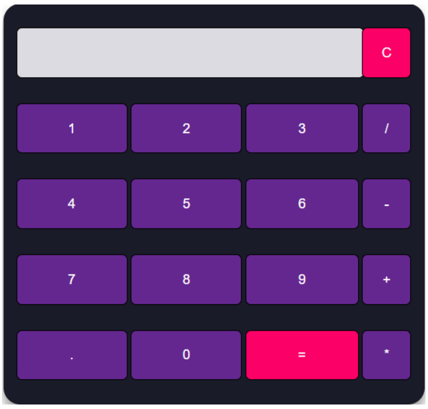
/***/

Aim: Create a web page containing simple calculator which should have basic arithmetic (+,-,*,/) operation on two floating point numbers and show result.

Validations to be followed:

- **. (Decimal point) should be taken only once for an operand.**
- **Operand can be negative.**
- **Division by zero must be shown proper error message in result.**

Sample screen shot:



/***/

PartB2.html

```
<!DOCTYPE html>
<html>
   <head>
      <title>Simple Calculator using HTML,CSS and JavaScript</title>
      <link rel="stylesheet" href="stylesheet.css">
   </head>
   <body>
      <table class="calculator">
         <tr>
            <td colspan="3"><input class="display-box" type="text" id="result" disabled/></td>
            <!--clearScreen() function clears all the values-->
            <td><input type="button" value="c" onclick="clearScreen()" id="btn" /> </td>
         </tr>
         <tr>
```

```
                <!--display() function display the value of clicked button-->
                <td><input type="button" value="1" onclick="display('1')" /> </td>
                <td><input type="button" value="2" onclick="display('2')" /> </td>
                <td><input type="button" value="3" onclick="display('3')" /> </td>
                <td><input type="button" value="/" onclick="display('/')" /> </td>
            </tr>
            <tr>
                <td><input type="button" value="4" onclick="display('4')" /> </td>
                <td><input type="button" value="5" onclick="display('5')" /> </td>
                <td><input type="button" value="6" onclick="display('6')" /> </td>
                <td><input type="button" value="-" onclick="display('-')" /> </td>
            </tr>
            <tr>
                <td><input type="button" value="7" onclick="display('7')" /> </td>
                <td><input type="button" value="8" onclick="display('8')" /> </td>
                <td><input type="button" value="9" onclick="display('9')" /> </td>
                <td><input type="button" value="+" onclick="display('+')" /> </td>
            </tr>
            <tr>
                <td><input type="button" value="." onclick="display('.')" /> </td>
                <td><input type="button" value="0" onclick="display('0')" /> </td>
                <!--calculate() function evaluates the mathematical expression-->
                <td><input type="button" value="=" onclick="calculate('=')" id="btn" /> </td>
                <td><input type="button" value="*" onclick="display('*')" /> </td>
            </tr>
        </table>
        <script type="text/JavaScript" src="script.js"></script>
    </body>
</html>
```

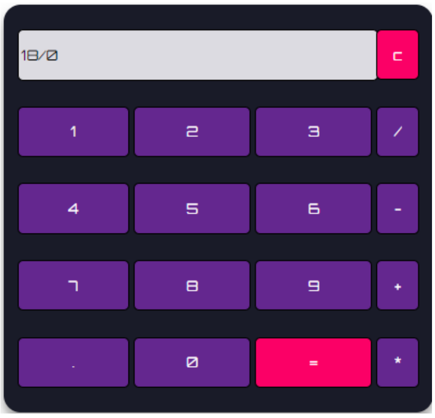
Script.js

```
//This function clear all the values
function clearScreen()
{
     document.getElementById("result").value="";
}
//This function display values
function display(value)
{
     document.getElementById("result").value+=value;
}
//This function evaluates the expression and returns result
function calculate()
{
    var p=document.getElementById("result").value;
    var q=eval(p);
    document.getElementById("result").value=q;
}
```

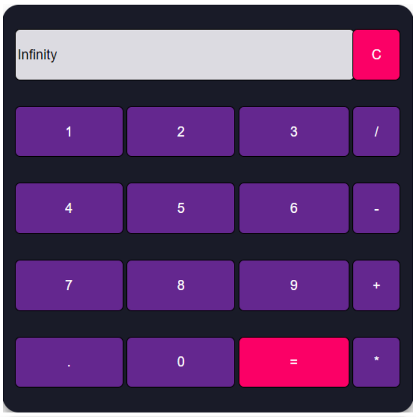
Style.css

```
.calculator
{
      padding: 10px;
      border-radius: 1em;
      height: 380px;
      width: 400px;
      margin: auto;
      background-color: #191b28;
      box-shadow: rgba(0,0,0,0.19) 0px 10px 20px,rgba(0,0,0,0.23) 0px 6px 6px ;
}
.display-box
{
    font-family: 'Orbitron', sans-serif;
    background-color: #dcdbe1;
    border: solid black 0.5px;
    color: black;
    border-radius: 5px;
    width: 100%;
    height: 65%;
}
#btn
{
    background-color: #fb0066;
}
input[type=button]
{
    font-family: 'Orbitron', sans-serif;
    background-color: #64278f;
    color: white;
    border: solid black 0.5px;
    width: 100%;
    border-radius: 5px;
    height: 70%;
    outline: none;
}
input:active[type=button]
{
    background: #e5e5e5;
    -webkit-box-shadow: inset 0px 0px 5px #c1c1c1;
    -moz-box-shadow: inset 0px 0px 5px #c1c1c1;
    box-shadow: inset 0px 0px 5px #c1c1c1;
}
```

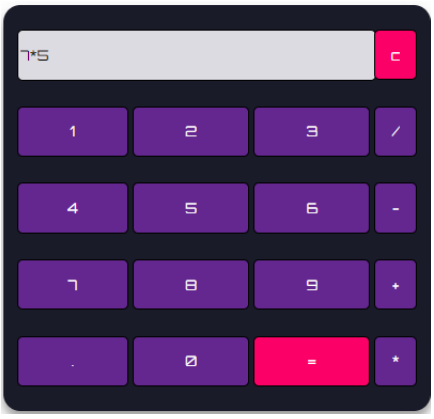
OUTPUT:



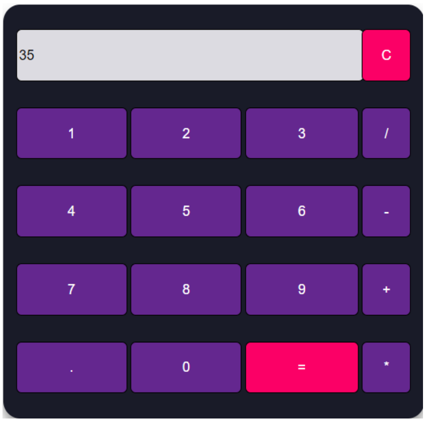
18/0



Infinity



7*5



35

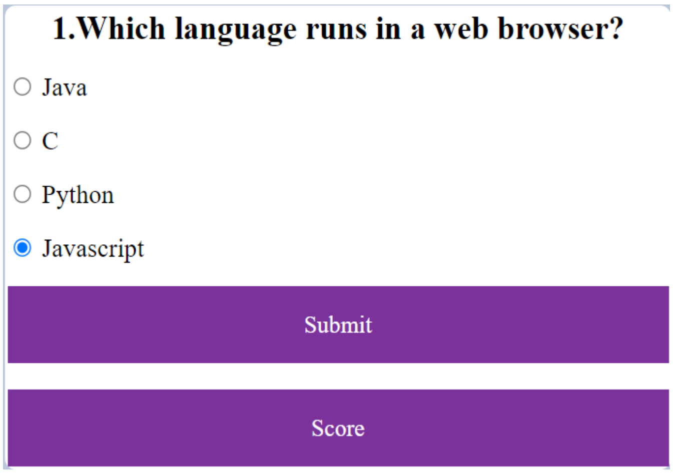
/* ** */

PROGRAM:3 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Create a HTML page make a quiz game where user should answer one question at a time, answers must be shown in radio buttons. Without submitting the answer, quiz should not move to next question (Minimum five questions must be there). When user wishes to get score (using score button) score should be displayed in alert 22 message. All the question must be loaded in same page (no page navigation is allowed) Sample screen shot:



1.Which language runs in a web browser?

○ Java

○ C

○ Python

● Javascript

/***/

PartB3.html

```
<!DOCTYPE html>

<html>

    <head>

        <title>QUIZ</title>

        <style>

            *{

                box-sizing: border-box;

            }

            .quiz-header

            {

                padding: 2px;

                border-radius: 8px;

                background-color: white;

            }

            h2

            {
```

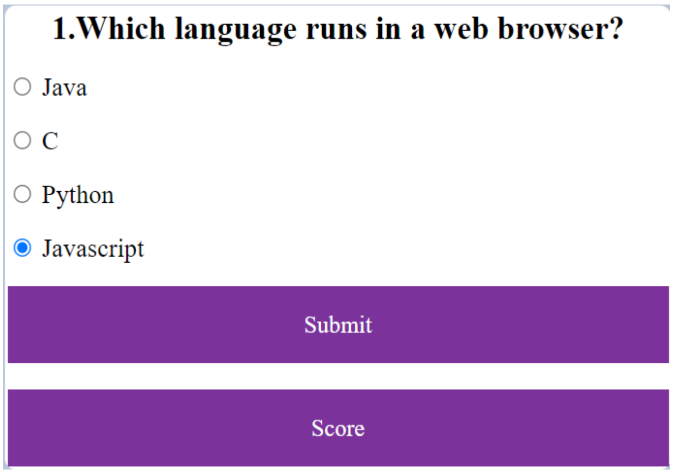
```
    padding: 1px;
    text-align: center;
    margin: 0;
}
body
{
    background-color: #b8c6db;
    display: flex;
    align-items: center;
    justify-content: center;
    height: 100vh;
    margin: 0;
}
.quiz-container
{
    background-color: #fff;
    border-radius: 100px;
    width: 500px;
    max-width: 95px;
}
ul
{
    list-style-type: none;
    padding: 0;
}
ul li
{
    font-size: 1.2rem;
    margin: 1rem 0;
}
ul li label
{
    cursor: pointer;
}
```

```
        button
        {
           background-color: #7b339b;
           color: #fff;
           border: none;
           display: block;
           width: 100%;
           cursor: pointer;
           font-size: 1.1rem;
           font-family: inherit;
           padding: 1.1rem;
        }
        button:hover
        {
           background-color: #732d91;
        }
        button:focus
        {
           outline: none;
           background-color: #5e3370;
        }
    </style>
</head>
<body>
    <section>
        <div class="quiz-container" id="quiz"></div>
        <div class="quiz-header">
            <h2 id="question">Question is loading...</h2>
            <ul>
                <li>
                    <input type="radio" name="answer" id="a" class="answer" />
                    <label for="a" id="a_text">Answer...</label>
                </li>
                <li>
                    <input type="radio" name="answer" id="b" class="answer" />
                    <label for="a" id="b_text">Answer...</label>
                </li>
                <li>
                    <input type="radio" name="answer" id="c" class="answer" />
                    <label for="a" id="c_text">Answer...</label>
                </li>
                <li>
                    <input type="radio" name="answer" id="d" class="answer" />
                    <label for="a" id="d_text">Answer...</label>
                </li>
           </ul>
           <button id="submit">Submit</button>
           <br>
           <button id="score" onclick="scorecheck()">Score</button>
        </div>
    </section>
    <script>
        const quizData=
        [
        {
           question: "1.Which language runs in a web browser?",
           a:"Java",
           b:"C",
           c:"Python",
           d:"Javascript",
           correct:"d",
        },
        {
           question: "2.What does CSS stand for?",
           a:"Central Style Sheets",
           b:"Cascading Style Sheets",
           c:"Cascading Simple Sheets",
           d:"Cars SUVs Sailboats",
           correct:"b",
        },
        {
           question: "3.What does HTML stand for?",
           a:"Hypertext Markup Language",
           b:"Hypertext Markdown Language",
           c:"Hyperloop Machine Language",
           d:"Helicopters Terminals Motorboats Lamborginis",
           correct:"a",
        },
        {
           question: "4.What yera was Javascript launched?",
           a:"1996",
           b:"1995",
           c:"1994",
           d:"none of the above",
           correct:"b",
        },
        {
           question: "5. <br> What type of tag is this?",
           a:"Break tag",
           b:"Broken one",
           c:"An opening tag",
           d:"A closing tag",
           correct:"a",
        }];
        const quiz=document.getElementById("quiz");
        const answerElements=document.querySelectorAll(".answer");
        const questionElement=document.getElementById("question");
        const a_text=document.getElementById("a_text");
        const b_text=document.getElementById("b_text");
        const c_text=document.getElementById("c_text");
        const d_text=document.getElementById("d_text");
        const submitButton=document.getElementById("submit");
```

```
let currentQuiz=0;
let score=0;
const deselectAnswers=()=>
{
   answerElements.forEach((answer)=> (answer.checked=false));
};
const getSelected=()=>
{
   let answer;
   answerElements.forEach((answerElements)=>
   {
      if (answerElements.checked)answer=answerElements.id;
   });
   return answer;
};
   const loadQuiz=()=>
   {
      deselectAnswers();
      const currentQuizData=quizData[currentQuiz];
      questionElement.innerText=currentQuizData.question;
      a_text.innerText=currentQuizData.a;
      b_text.innerText=currentQuizData.b;
      c_text.innerText=currentQuizData.c;
      d_text.innerText=currentQuizData.d;
   };
   function scorecheck()
   {
      alert(score);
   }
   loadQuiz();
   submitButton.addEventListener("click",()=>
   {
      const answer=getSelected();
      if (answer)
```

```
          {
            if (answer===quizData[currentQuiz].correct)score++;
              currentQuiz++;
            if (currentQuiz<quizData.length)
              loadQuiz();
            else
            {
              quiz.innerHTML=`<h2>You answered ${score}questions correctly</h2>
              <button onclick="history.go(0)">PlayAgain </button>`
            }
          }
        });
    </script>
  </body>
</html>
```

OUTPUT:



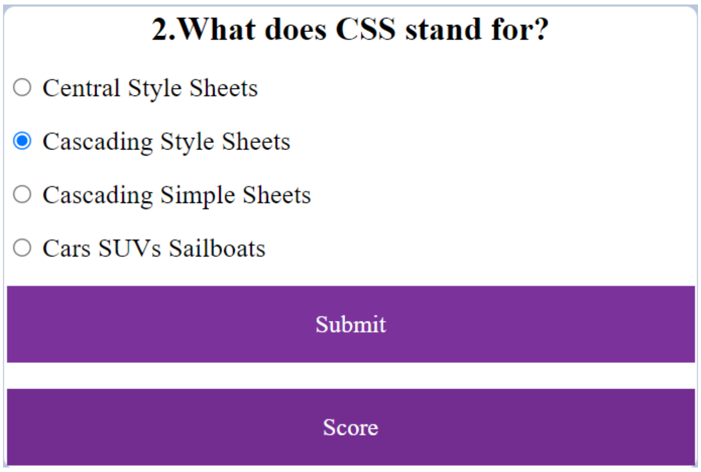
1.Which language runs in a web browser?

○ Java

○ C

○ Python

● Javascript



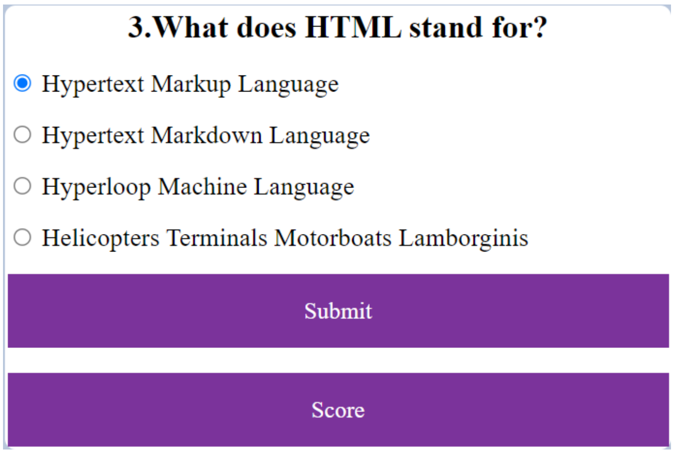
2.What does CSS stand for?

○ Central Style Sheets

● Cascading Style Sheets

○ Cascading Simple Sheets

○ Cars SUVs Sailboats



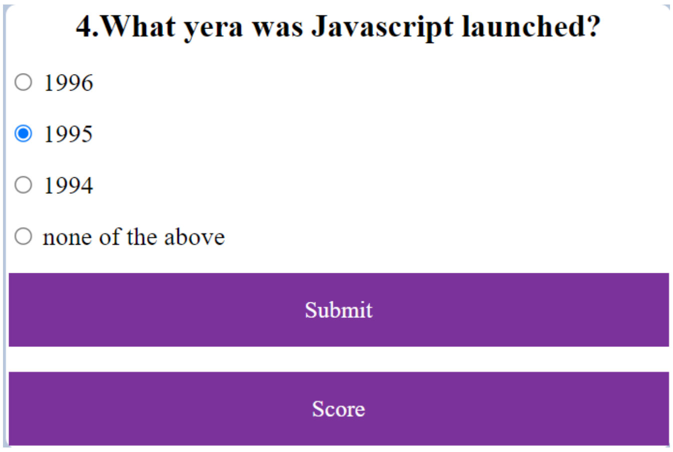
3.What does HTML stand for?

● Hypertext Markup Language

○ Hypertext Markdown Language

○ Hyperloop Machine Language

○ Helicopters Terminals Motorboats Lamborginis



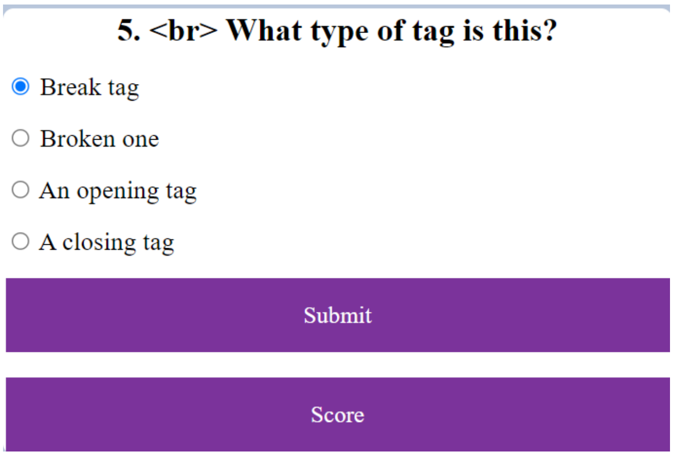
4.What yera was Javascript launched?

○ 1996

● 1995

○ 1994

○ none of the above



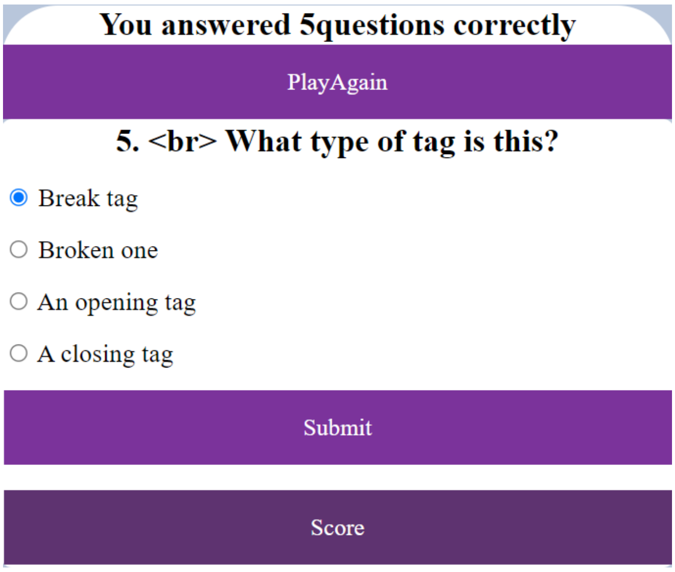
5.
 What type of tag is this?

● Break tag

○ Broken one

○ An opening tag

○ A closing tag



You answered 5questions correctly

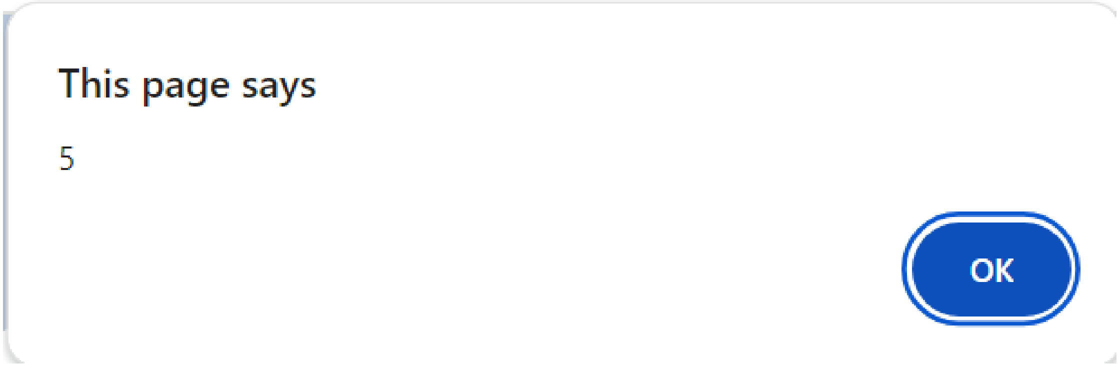
5.
 What type of tag is this?

● Break tag

○ Broken one

○ An opening tag

○ A closing tag



This page says

5

/* ** */

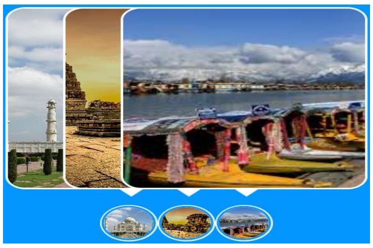
PROGRAM:4 **DATE:**

NAME: **REG.NO:U05DP22S0**

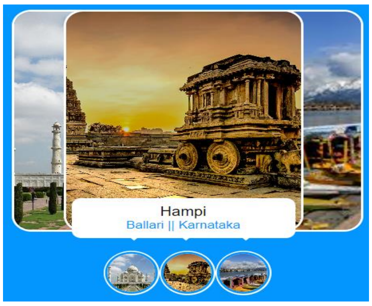
//**

Aim: Create a web page using HTML/CSS which contains cards (shown as a stack of cards) with image of a tourist place and below that is a thumbnail (shown in circle with image). When mouse hovers over thumbnail, corresponding card comes in front and also small description about the tourist place will be displayed. Use ONLY CSS animation and transition.

Initial interface:



Interface should look like below screenshot when mouse hovered on thumbnail:



(Java script should not be used to animate.)

//**

PartB4.html

```
<!DOCTYPE html>
  <head>
    <title>  Hover Animation </title>
    <link rel="stylesheet" href="style.css">
   </head>
<body>
  <div class="container">
 <div class="icon-image">
          <div class="icon">
            <img src="TajMahal.jpg" alt="">
          </div>
          <div class="hover-image">
```

```
        <div class="img">
            <img src="TajMahal.jpg" alt="">
            </div>
             <div class="content">
            <div class="details">
            <div class="name">Taj Mahal</div>
            <div class="job"> Agra || Uttar Pradesh</div>
            </div>
            </div>
      </div>
  </div>
      <div class="icon-image">
        <div class="icon">
          <img src="hampi.jpg" alt="">
        </div>
      <div class="hover-image">
        <div class="img">
          <img src="hampi.jpg" alt="">
        </div>
      <div class="content">
        <div class="details">
          <div class="name">Hampi</div>
          <div class="job">Ballari || Karnataka</div>
       </div>
      </div>
      </div>
  </div>
   <div class="icon-image">
        <div class="icon">
          <img src="dallake.jpg" alt="">
        </div>
      <div class="hover-image">
        <div class="img">
```

```
          <img src="dallake.jpg" alt="">
        </div>
      <div class="content">
        <div class="details">
          <div class="name">Daal Lake</div>
          <div class="job">Kashmir || Kashmir</div>
        </div>
      </div>
      </div>
    </div>
  </div>
</body>
</html>
```

Style.css

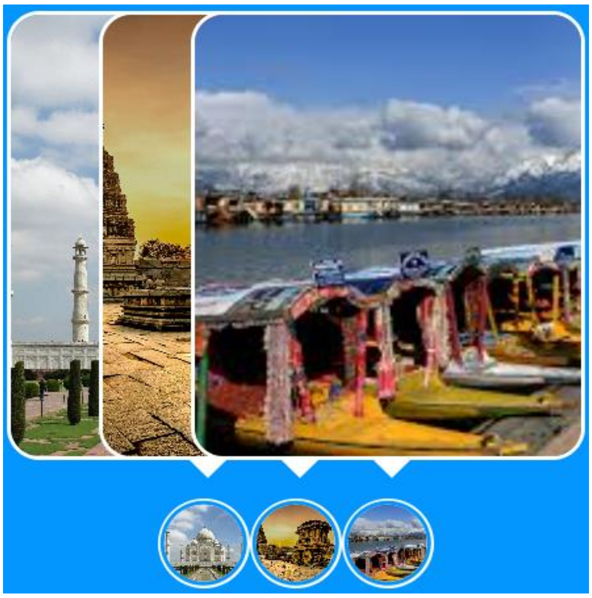
```
*{
    box-sizing: border-box;
    font-family: 'Poppins',sans-serif;
  }
  body{
    height: 100vh;
    width: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
    background: #0396FF;
  }
  .container{
    height: 500px;
    display: flex;
    min-width: 400px;
    align-items: flex-end;
   }
  .icon-image{
```

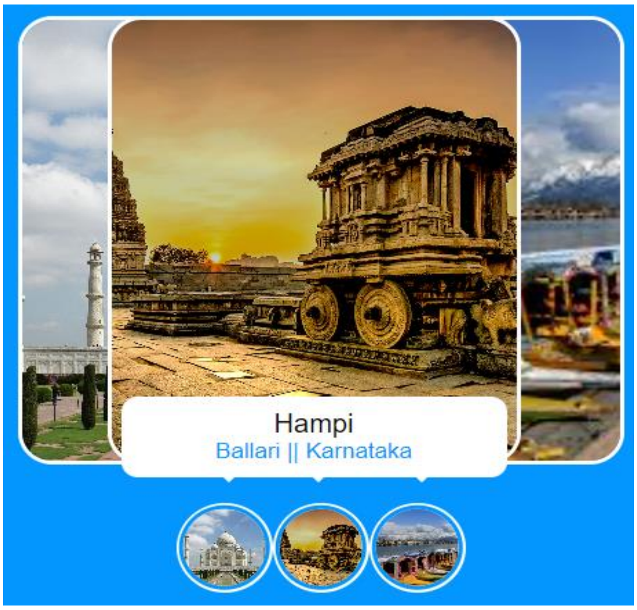
```
  position: relative;
  height: 70px;
  width: 70px;
  cursor: pointer;
  border-radius: 50%;
  background: #fff;
}
.icon-image .icon img{
  position: absolute;
  height: 95%;
  width: 95%;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
  object-fit: cover;
  border-radius: 50%;
  border: 3px solid #0396FF;
}
.icon-image .hover-image{
  position: absolute;
  height: 350px;
  width: 300px;
  bottom: 100px;
  left: 50%;
  transform: translateX(-50%);
 }
.icon-image:hover .hover-image{
  left: -200px;
  z-index: 12;
  transform: translateX(80px);
  transition: left 0.5s ease, z-index 0s, transform 0.5s ease;
  transition-delay: 0s, 0.5s, 0.5s;
}
```

```
.hover-image img{
  position: absolute;
  height: 100%;
  width: 100%;
  object-fit: cover;
  border: 3px solid #fff;
  border-radius: 25px;
}
.hover-image .content{
  position: absolute;
  width: 100%;
  bottom: -10px;
padding: 0 10px;
}
.content::before{
content: '';
position: absolute;
height: 20px;
width: 20px;
background: #fff;
left: 50%;
bottom: -7px;
transform: rotate(45deg) translateX(-50%);
z-index: -1;
}
.content .details{
  position: relative;
  background: #fff;
  padding: 10px;
  border-radius: 12px;
  text-align: center;
  opacity: 0;
  }
```

```
.icon-image:hover .details{

  transition: all 0.5s ease;

  transition-delay: 0.9s;

  opacity: 1;


}.content .details .name{

 font-size: 20px;

 font-weight: 500;

}

.content .details .job{

 font-size: 17px;

 color: #0396FF;

}
```

OUTPUT:





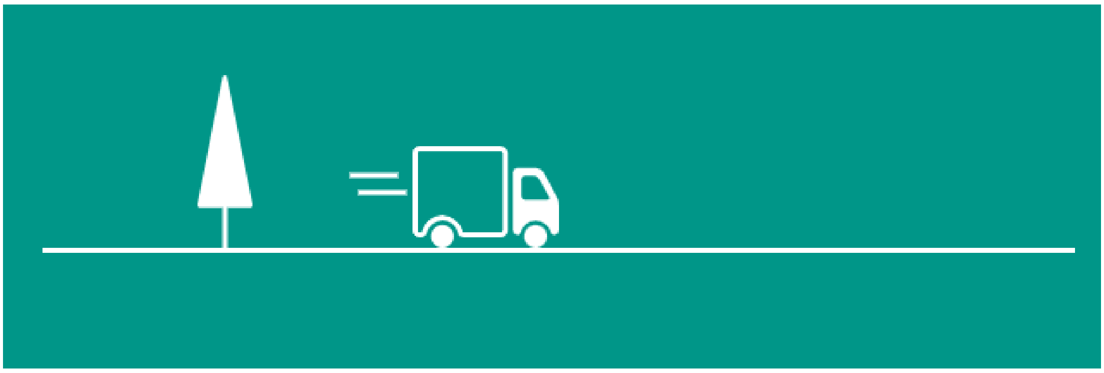
/* *** */

PROGRAM:5 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Create a web page using HTML5/CSS3 to animate a truck movement. While truck moves on mountains and trees should move in the back ground.



Background hills must be created using CSS only and for tree, truck and wheels download the images from the following URLs.

https://s3-us-west-2.amazonaws.com/s.cdpn.io/130015/tree.svg

https://s3-us-west-2.amazonaws.com/s.cdpn.io/130015/truck.svg

https://s3-us-west-2.amazonaws.com/s.cdpn.io/130015/wheels.svg

Animation must be implemented using ONLY CSS and Java script should not be used.

/***/

PartB5.html

```
<!DOCTYPE html>
<head>
<title>Speedy Truck</title>
<link rel="stylesheet" href="speedytruck.css">
</head>
<body>
<div class="loop-wrapper">
  <div class="tree"></div>
  <div class="truck"></div>
  <div class="wheels"></div>
</div>
</body>
</html>
```

speedytruck.css

```
body
 {
  background: #009688;
  }
.loop-wrapper
{
  margin: 0 auto;
  position: relative;
  width: 600px;
  height: 250px;
  overflow: hidden;
  border-bottom: 3px solid #fff;
   }

.tree
{
  position: absolute;
  height: 100px;
  width: 35px;
  bottom: 0;
  background: url(tree.svg) no-repeat;
}

.truck, .wheels
 {
  width: 85px;
  bottom: 0px;
  right: 50%;
  position: absolute;
   }
.truck
{
  background: url(truck.svg) no-repeat;
   height: 60px;
 }
.truck:before
{
  content: " ";
  position: absolute;
  width: 25px;
  box-shadow:
    -30px 28px 0 1.5px #fff,
     -35px 18px 0 1.5px #fff;
}
.wheels
{
  background: url(wheels.svg) no-repeat;
  height: 15px;
}
.tree
 {
    animation: tree 3s 0.000s linear infinite;
 }
@keyframes tree
{
  0%
{
     transform: translate(1350px);
}
50% {}
100%
{
    transform: translate(-50px);
 }
}
```

tree.svg

```
<svg xmlns="http://www.w3.org/2000/svg" width="32" height="100" viewBox="0 0 32 100">

<path fill="#FFF" d="M31.945 74.986L17.37 1.148A1.416 1.416 0 0 0 15.988 0c-.673 0-1.252.48-1.383
1.148L.027 74.986c-.083.42.025.854.292 1.186.268.332.669.523
1.091.523h13.167V100h2.821V76.695h13.165c.422 0 .821-.191 1.09-.523.27-.331.375-.766.292-1.186z"/>

</svg>
```

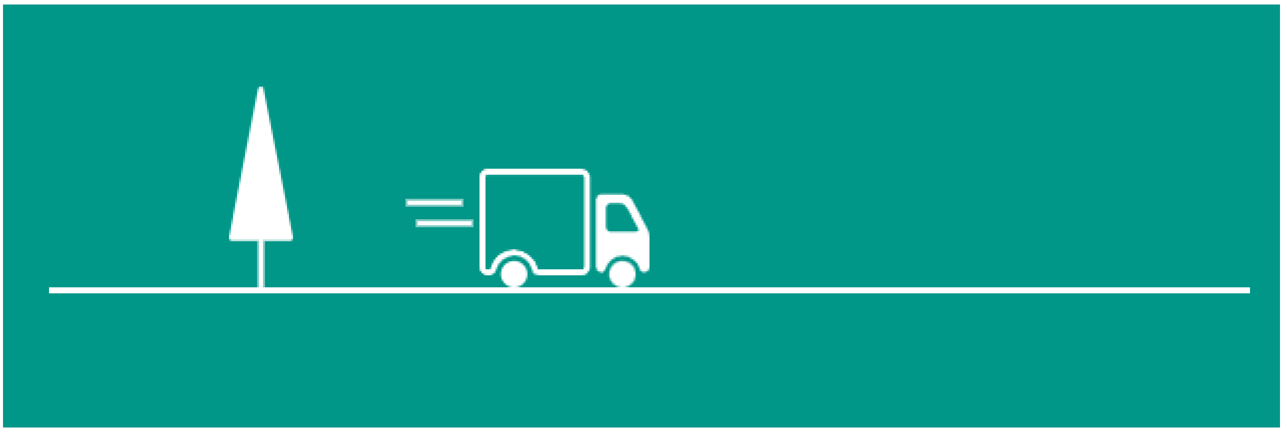
truck.svg

```
<svg xmlns="http://www.w3.org/2000/svg" width="85" height="60" viewBox="0 0 85 60">
  <path fill="#FFF" d="M5.138 54.039a3.77 3.77 0 0 1-.149-.005l-.859.002A4.134 4.134 0 0 1 0
49.906V5.195a4.135 4.135 0 0 1 4.13-4.133h46.973c2.272 0 4.123 1.95 4.123 4.229v44.612a4.133 4.133 0
0 1-4.127 4.131H29.962c-1.091-.019-3.239-.691-3.727-3.128-1.067-3.957-4.755-6.789-8.947-6.789-4.291
0-8.098 3.041-9.051 7.229-.225 1.817-1.723 2.693-3.099 2.693zM4.127 4.208c-.596 0-1.083.389-
1.083.987v44.709c0 .597.488 1.086 1.086 1.086h1.004c.005-.047.123-.271.133-.317 1.267-5.562 6.323-
9.597 12.022-9.597a12.356 12.356 0 0 1 11.912 9.137c.151.734.681.777.785.777h21.111c.596 0 1.083-.487
1.083-1.084V5.293c0-.598-.485-1.085-1.083-1.085H4.127zM84.929 35.828s-.093-2.093-.288-3.118c-.164-
.873-1.087-2.644-1.087-2.644l-7.299-13.59c-.951-1.52-3.115-2.707-4.558-2.707H60.785a2.608 2.608 0 0
0-2.606 2.609v33.526a2.609 2.609 0 0 0 2.606 2.609h.643s.949-.123 1.304-1.312c.986-3.933 4.514-6.862
8.749-6.862 4.145 0 7.612 2.808 8.682 6.613.442 1.612 2.231 1.561 2.231 1.561A2.608 2.608 0 0 0 85
49.904l-.071-14.076zm-6.603-3.511H65.312a1.996 1.996 0 0 1-1.989-1.99V20.143c0-1.098.894-1.99
1.989-1.99h4.365c2.483 0 4.117 2.009 4.63 2.978l5.065 9.434c.521.962.053 1.752-1.046 1.752z"/>
  <path fill="#009688" d="M4.127 4.208c-.596 0-1.083.389-1.083.987v44.709c0 .597.488 1.086 1.086
1.086h1.004c.005-.047.123-.271.133-.317 1.267-5.562 6.323-9.597 12.022-9.597a12.356 12.356 0 0 1
11.912 9.137c.151.734.681.777.785.777h21.111c.596 0 1.083-.487 1.083-1.084V5.293c0-.598-.485-1.085-
1.083-1.085H4.127z"/>
</svg>
```

wheels.svg

```
<svg xmlns="http://www.w3.org/2000/svg" width="85" height="15" viewBox="0 0 85 15">
  <circle fill="#FFF" cx="17.289" cy="8.413" r="6.587"/>
  <circle fill="#FFF" cx="71.48" cy="8.412" r="6.586"/>
</svg>
```

OUTPUT:



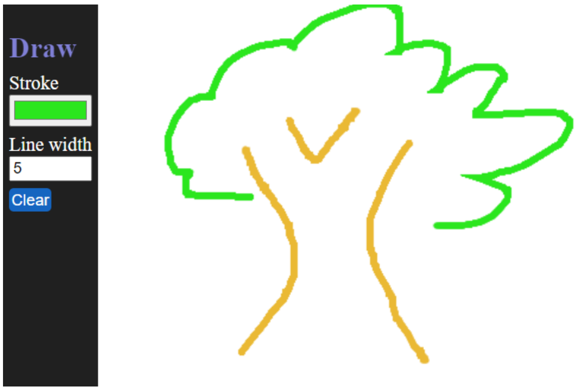
/* ** */

PROGRAM:6 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Create a simple paint app which draws lines based on the selected colour (chosen using color input) with selected thickness (chosen using number input) and there must be CLEAR button to clear the canvas. Sample screen shot:



/***/

PartB6.html

```
<html lang="en">
    <head>
        <link rel="stylesheet" href="Paint.css">
        <title>Drawing app</title>
    </head>
    <body>
        <section class="container">
            <div id="tollbar">
                <h1>Draw</h1>
                <label="stroke">Stroke</label>
                <input id="stroke" name='stroke'type="color">
                <label="lineWidth">Line width</label>
                <input id="lineWidth" name='lineWidth' type="number" value="5">
                <button id="clear">Clear</button>
            </div>
            <div class="drawing-board">
                <canvas id="drawing-board"></canvas>
            </div>
        </section>
        <script src="Paint.js"></script>

    </body>

</html>
```

Paint.CSS

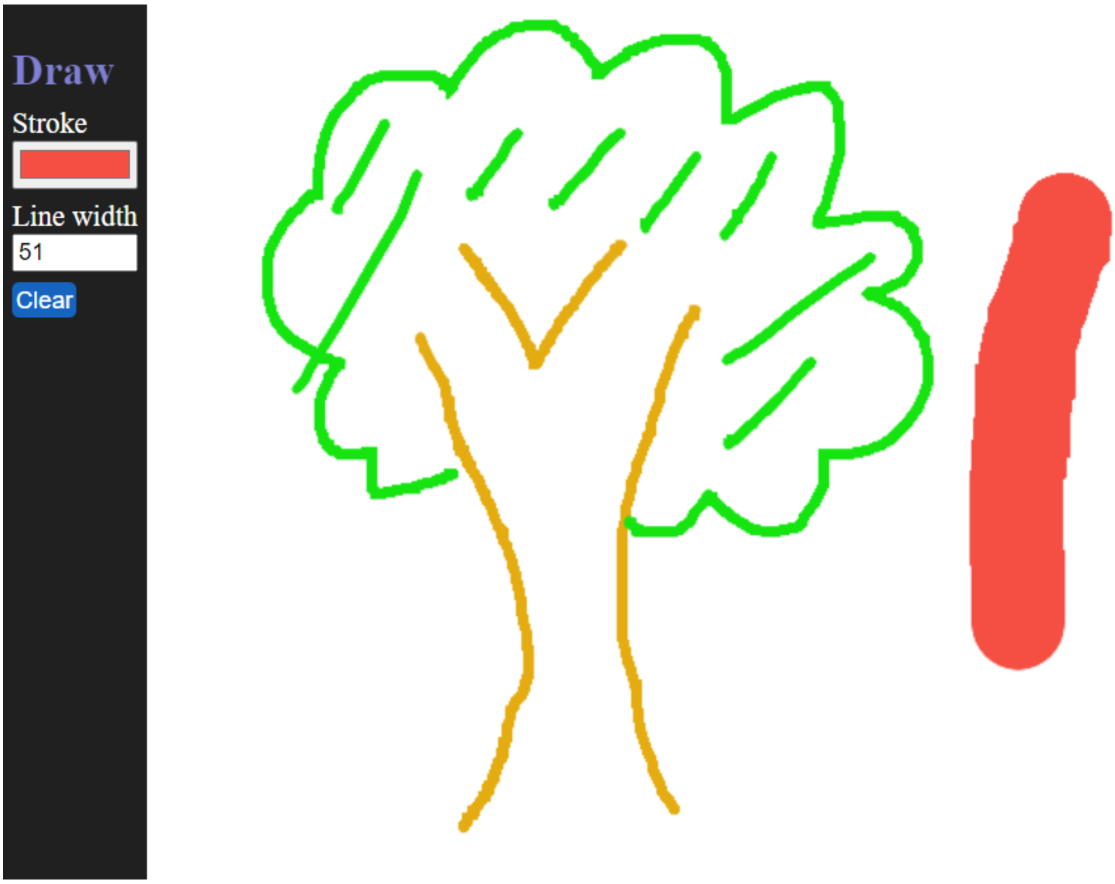
```
body

{

    margin: 0;

    padding: 0;

    height: 100%;

    overflow: hidden;

    color: white;

}

h1

{

    background: #7f7fd5;

    background: -webkit-linear-gradient(to right,#91EAE4,#86A8E7);

    background-clip: text;

    -webkit-background-clip: text;

    -webkit-text-fill-color: transparent;

}

.container

{

    height: 100%;

    display: flex;

}

#toolbar

{

    display: flex;

    flex-direction: column;

    padding: 5px;

    width: 70px;

    background-color: #202020;

}
```

```
#toolbar *
{
   margin-bottom: 6px;
}
#toolbar label
{
   font-size: 12px;
}
#toolbar input
{
   width: 100%;
}
#toolbar button
{
   background-color: #1565c0;
   border: none;
   border-radius: 4px;
   color: white;
   padding: 2px;
}
```

Paint.js

```
const canvas=document.getElementById('drawing-board');
const toolbar=document.getElementById('toolbar');
const ctx=canvas.getContext('2d');
const canvasoffsetX=canvas.offsetLeft;
const canvasoffsetY=canvas.offsetTop;
canvas.width=window.innerWidth-canvasoffsetX;
canvas.height=window.innerHeight-canvasoffsetY;
let isPainting=false;
let lineWidth=5;
let startX;
let startY;
toolbar.addEventListener('click',e=>{
    if(e.target.id==='clear')
    {
        ctx.clearRect(0,0,canvas.width,canvas.height);
    }
});
toolbar.addEventListener('change',e=>{
    if(e.target.id==='stroke')
    {
        ctx.strokeStyle=e.target.value;
    }
    if(e.target.id==='lineWidth')
    {
       lineWidth=e.target.value;
    }
});
const draw=(e)=>{
    if(!isPainting)
    {
        return;
    }
    ctx.lineWidth=lineWidth;
    ctx.lineCap='round';
    ctx.lineTo(e.clientX-canvasoffsetX,e.clientY);
    ctx.stroke();
}
canvas.addEventListener('mousedown',(e)=>{
    isPainting=true;
    startX=e.clientX;
    startY=e.clientY;
});
canvas.addEventListener('mouseup',e=>{
    isPainting=false;
    ctx.stroke();
    ctx.beginPath();

});

canvas.addEventListener('mousemove',draw);
```

OUTPUT:



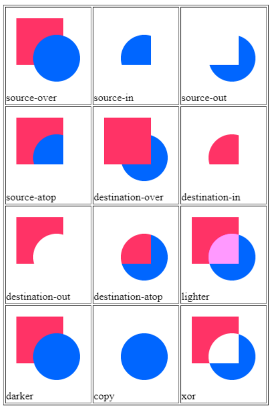
/* ** */

PROGRAM:7 **DATE:**

NAME: **REG.NO:U05DP22S0**

/**/

Aim: Create web page using HTML5 canvas element to illustrate all canvas composition. Output must exactly look like the following screenshot:



/**/

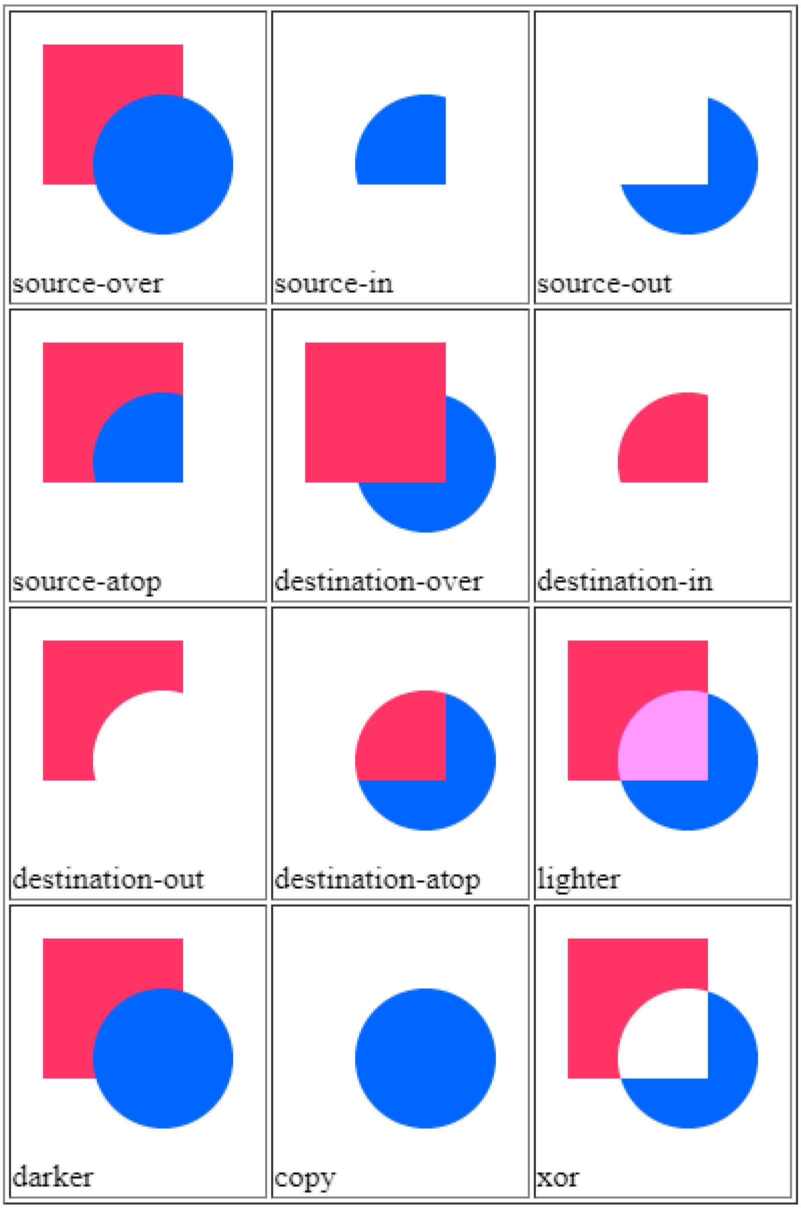
PartB7.html

```
<!DOCTYPE HTML>

<html>

  <head>
    <script type="text/javascript">
      var compositeTypes=
      [
        'source-over','source-in','source-out','source-atop',
        'destination-over','destination-in','destination-out','destination-atop',
        'lighter','darker','copy','xor'
      ];
      function drawShape()
      {
        for(i=0;i<compositeTypes.length;i++)
        {
          var label=document.createTextNode(compositeTypes[i]);
          document.getElementById('lab'+i).appendChild(label);
          var ctx=document.getElementById('tut'+i).getContext('2d');

          //draw rectangle
          ctx.fillStyle="#ff3366";
          ctx.fillRect(15,15,70,70);

          //set composite properly
          ctx.globalCompositeOperation=compositeTypes[i];

          //draw circle
          ctx.fillStyle="#0066ff";
          ctx.beginPath();
          ctx.arc(75,75,35,0,Math.PI*2,true);
```

```
                ctx.fill();
              }
           }
        </script>
    </head>
    <body onload="drawShape();">
        <table border="1" align="center">
           <tr><td>
                <canvas id="tut0" width="125" height="125"></canvas><br/>
                <label id="lab0"></label>
              </td>
              <td>
                <canvas id="tut1" width="125" height="125"></canvas><br/>
                <label id="lab1"></label>
              </td>
              <td>
                <canvas id="tut2" width="125" height="125"></canvas><br/>
                <label id="lab2"></label>
              </td>
           </tr>
 <tr><td>
                <canvas id="tut3" width="125" height="125"></canvas><br/>
                <label id="lab3"></label>
              </td>
              <td>
                <canvas id="tut4" width="125" height="125"></canvas><br/>
                <label id="lab4"></label>
              </td>
              <td>
                <canvas id="tut5" width="125" height="125"></canvas><br/>
                <label id="lab5"></label>
              </td>
           </tr>
           <tr> <td>
                <canvas id="tut6" width="125" height="125"></canvas><br/>
                <label id="lab6"></label>
              </td>
              <td>
                <canvas id="tut7" width="125" height="125"></canvas><br/>
                <label id="lab7"></label>
              </td>
              <td>
                <canvas id="tut8" width="125" height="125"></canvas><br/>
                <label id="lab8"></label>
              </td>
           </tr>
           <tr> <td>
                <canvas id="tut9" width="125" height="125"></canvas><br/>
                <label id="lab9"></label>
              </td>
              <td>
                <canvas id="tut10" width="125" height="125"></canvas><br/>
                <label id="lab10"></label>
              </td>
```

```
                <td>
                    <canvas id="tut11" width="125" height="125"></canvas><br/>
                    <label id="lab11"></label>
                </td>
            </tr>
        </table>
    </body>
</html>
```

OUTPUT:



source-over	source-in	source-out
source-atop	destination-over	destination-in
destination-out	destination-atop	lighter
darker	copy	xor

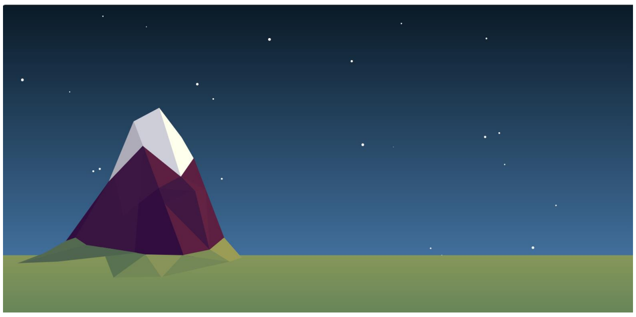
/* ** */

PROGRAM:8 **DATE:**

NAME: **REG.NO:U05DP22S0**

/***/

Aim: Create a web page which must be as shown in below image using HTML5, SVG and CSS3. Here Mountain, trees and clouds must be drawn using SVG, Clouds must have bounce animation (css animation), and stars in sky changes their position randomly for every time page is loaded (java script can be used). Sky and stars must be created using element.



/***/

PartB8.html

```
<!DOCTYPE html>

<head>

<title> SVG SCENE</title>
<link rel="stylesheet" type="text/css" href="sky.css"/>
<script>
window.onload = function()
{
var sky = document.getElementById("sky");

sky.width = document.body.clientWidth;
sky.height = document.body.clientHeight;

if (sky.getContext){
   var skyContext = sky.getContext("2d");

   var radius = 2;

   for(var star = 0; star < 50; star++){
      var min = ( Math.random() * 30 + 5 ) / 10;
      var max = sky.width - radius;

      var centerX = Math.floor(Math.random() * (max - min + 1)) + min;
      var centerY = Math.floor(Math.random() * (max - min + 1)) + min;

      skyContext.beginPath();
      skyContext.arc(centerX, centerY, min, 0, 2 * Math.PI);

      skyContext.fillStyle = "rgb(255, 255, 255)";
      skyContext.fill();
   }
}
}
</script>
</head>
<body>
<section class="stage">

    <svg height="390" width="550" id="mountains">
        <polygon points="300,390 190,90 320,130 400,340" fill="#5d2042" />
        <polygon points="0,340 108,190 194,100 201,120 301,390" fill="#320e40" />
        <polygon points="14,348 117,174 194,102 172,377" fill="#3b1642" fill-opacity="0.8" />
        <polygon points="120,174 194,102 233,206 144,275" fill="#3d1744" fill-opacity="0.9" />

        <polygon points="233,206 288,177 324,214" fill="#421943" />
        <polygon points="233,206 324,214 247,245" fill="#3e1743" />
        <polygon points="247,245 324,214 360,360" fill="#411842" />
        <polygon points="324,214 288,177 350,210" fill="#632242" />
        <polygon points="324,214 350,210 360,360" fill="#652343" />

        <g id="apex">
            <polygon points="108,190 170,40 194,100" fill="#aeacb9" />
            <polygon points="170,40 234,6 260,70 288,178 194,102" fill="#ceced8" />
            <polygon points="234,6 290,80 320,132 288,178" fill="#ffffed" />
        </g>
    </svg>
    <svg id="ground"></svg>

    <svg id="hills" width="700" height="170">
        <polygon points="480,70 530,100 560,90 516,40" fill="#9b9d57" />
        <polygon points="480,70 530,100 412,84" fill="#7d8f57" />
        <polygon points="530,100 412,84 360,138" fill="#748857" />
        <polygon points="360,138 240,140 320,82" fill="#748857" />

        <polygon points="412,84 360,140 320,82" fill="#88945a" />
        <polygon points="320,82 240,140 210,64" fill="#597252" />
        <polygon points="300,78 100,100 0,104 170,58" fill="#4f654f" />
        <polygon points="172,58 145,40 122,48 66,79 0,104" fill="#536a50" />
    </svg>

    <canvas id="sky"></canvas>
</section>
</body>
</html>
```

sky.css

```
html, body
{
             height: 100%;
            width: 100%
}

 body {
             min-height: 100%;
             background: linear-gradient(to bottom, #0a1a27, #4e83b7);
             position: relative;
}
body #mountains {
             z-index: 2;
             position: absolute;
             bottom: 20%;
             margin-bottom: -16px;
             left: 10%;
}
 body #ground {
             z-index: 1;
             width: 100%;
             height: 20%;
             position: absolute;
             bottom: 0;
             background: #879759;
             background: linear-gradient(to bottom, #879759, #648459);
}
 body #hills {
             //background: rgba(255, 0, 0, 0.3);
             position: absolute;
             left: 10%;
             margin-left: -120px;
             margin-bottom: -86px;
             bottom: 20%;
             z-index: 3;
}
 body #stars {
             width: 100%;
             height: 100%;
}
```

OUTPUT: