**Web Development Internship**

**Day-6 Assignment**

**SE-IT-B-06**

**Abhishek Pandita**

**Q1.**

**Ans.**

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100"

style="border:1px solid #d3d3d3;">

Your browser does not support the canvas element.

</canvas>

<script>

var canvas = document.getElementById("myCanvas");

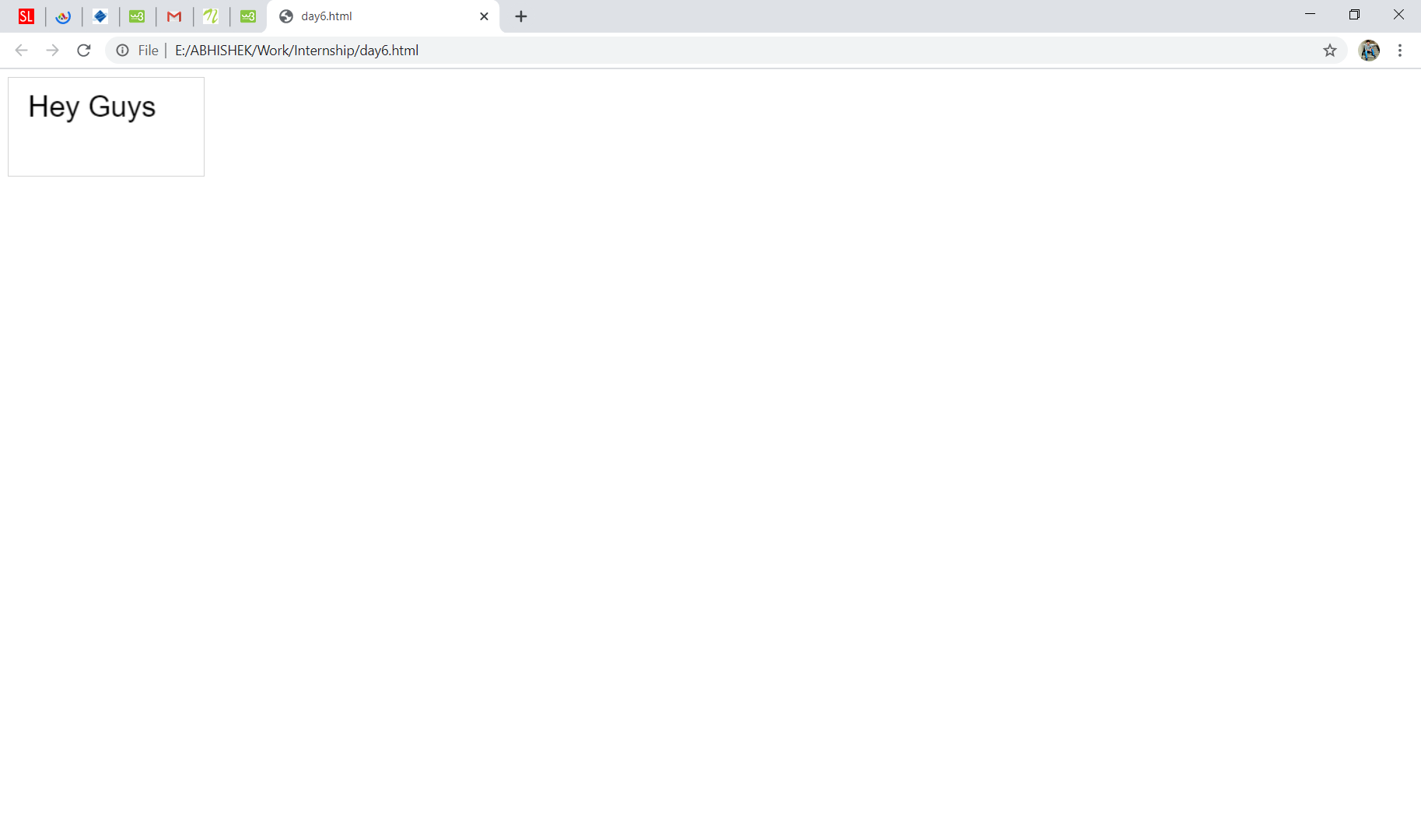
var ctx = canvas.getContext("2d");

ctx.font = "30px Arial";

ctx.fillText("Hey Guys",20,40);

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100"

style="border:1px solid #d3d3d3;">

Your browser does not support the canvas element.

</canvas>

<script>

var canvas = document.getElementById("myCanvas");

var ctx = canvas.getContext("2d");

ctx.beginPath();

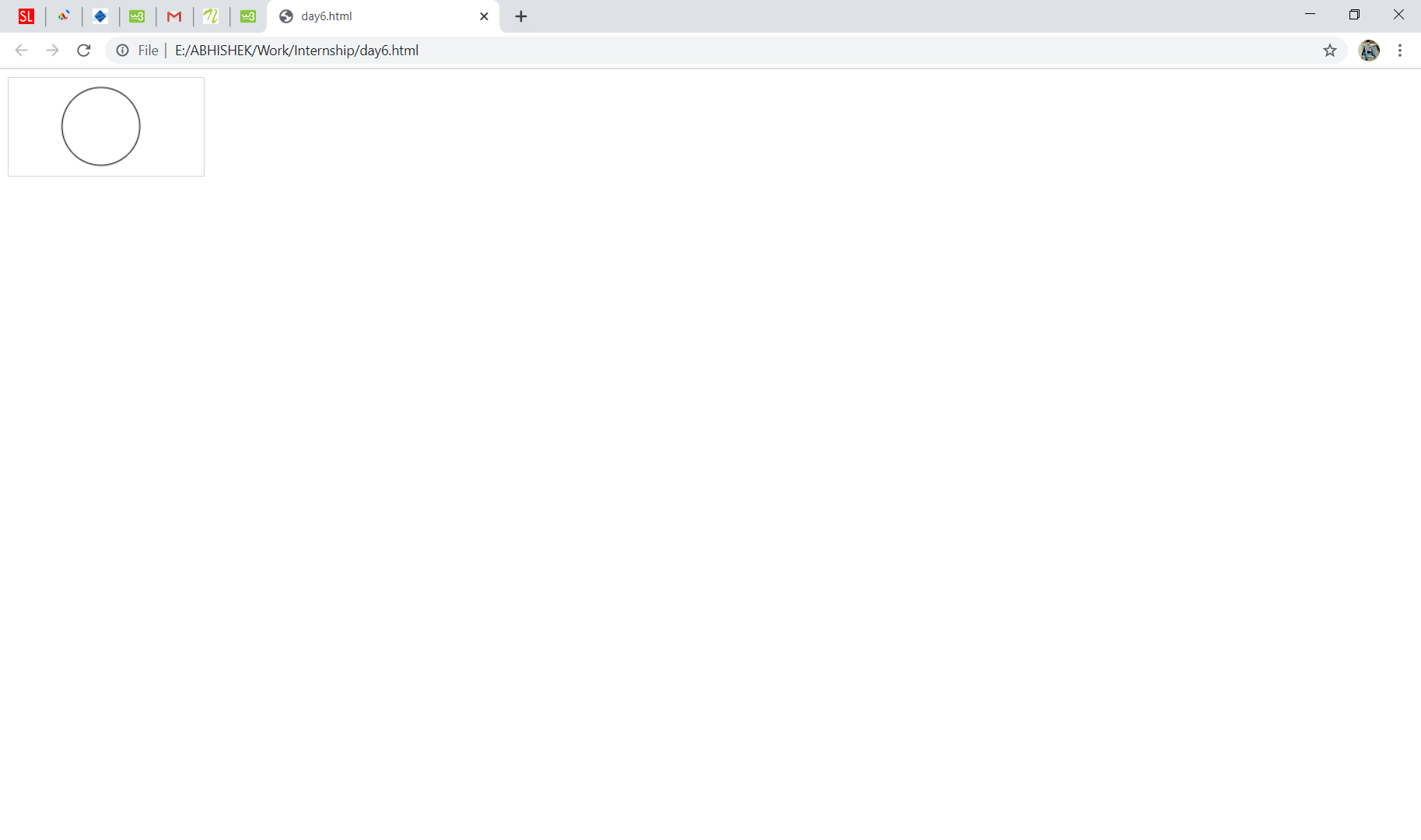
ctx.arc(95,50,40,0,2\*Math.PI);

ctx.stroke();

</script>

</body>

</html>



**Q2.**

**Ans.**

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100"

style="border:1px solid #d3d3d3;">

Your browser does not support the HTML5 canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

var grd = ctx.createLinearGradient(0,0,200,0);

grd.addColorStop(0,"blue");

grd.addColorStop(1,"white");

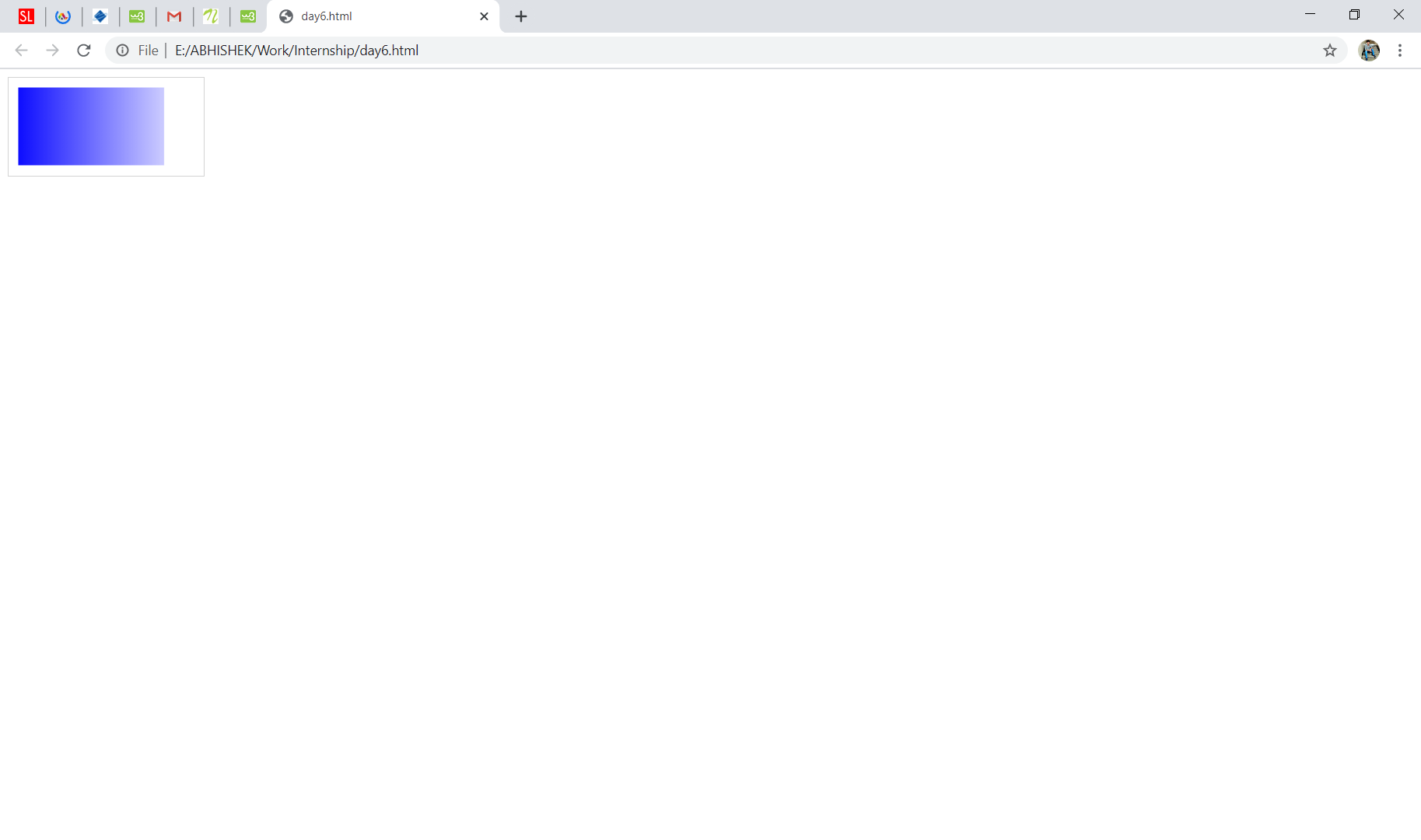
ctx.fillStyle = grd;

ctx.fillRect(10,10,150,80);

</script>

</body>

</html>



**Q3.**

**Ans.**

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100"

style="border:1px solid #d3d3d3;">

Your browser does not support the canvas element.

</canvas>

<script>

var canvas = document.getElementById("myCanvas");

var ctx = canvas.getContext("2d");

ctx.moveTo(0,50);

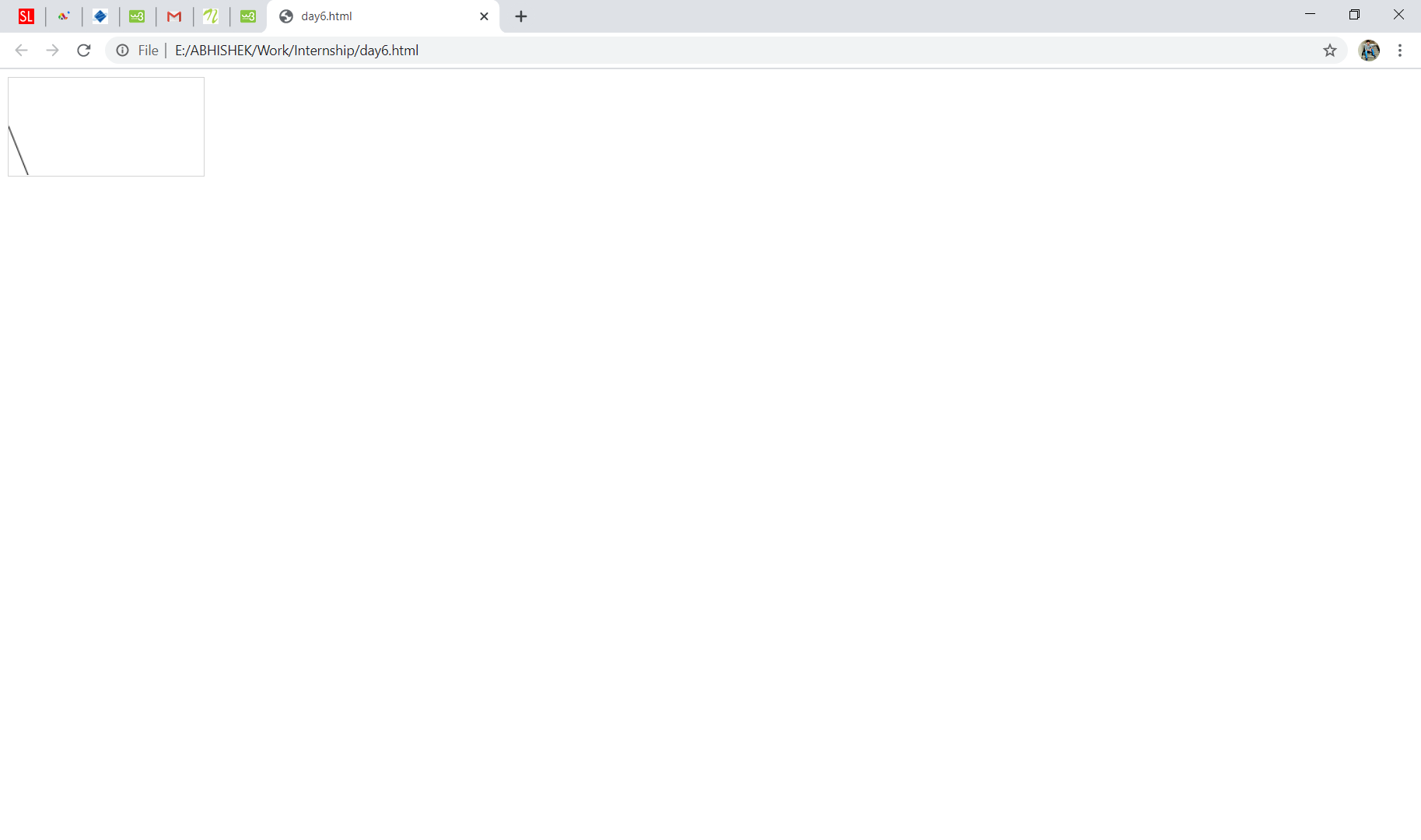
ctx.lineTo(100,300);

ctx.stroke();

</script>

</body>

</html>



**Q4.**

**Ans.**

<!DOCTYPE html>

<html>

<body>

<svg height="100" width="100">

<circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<svg width="400" height="110">

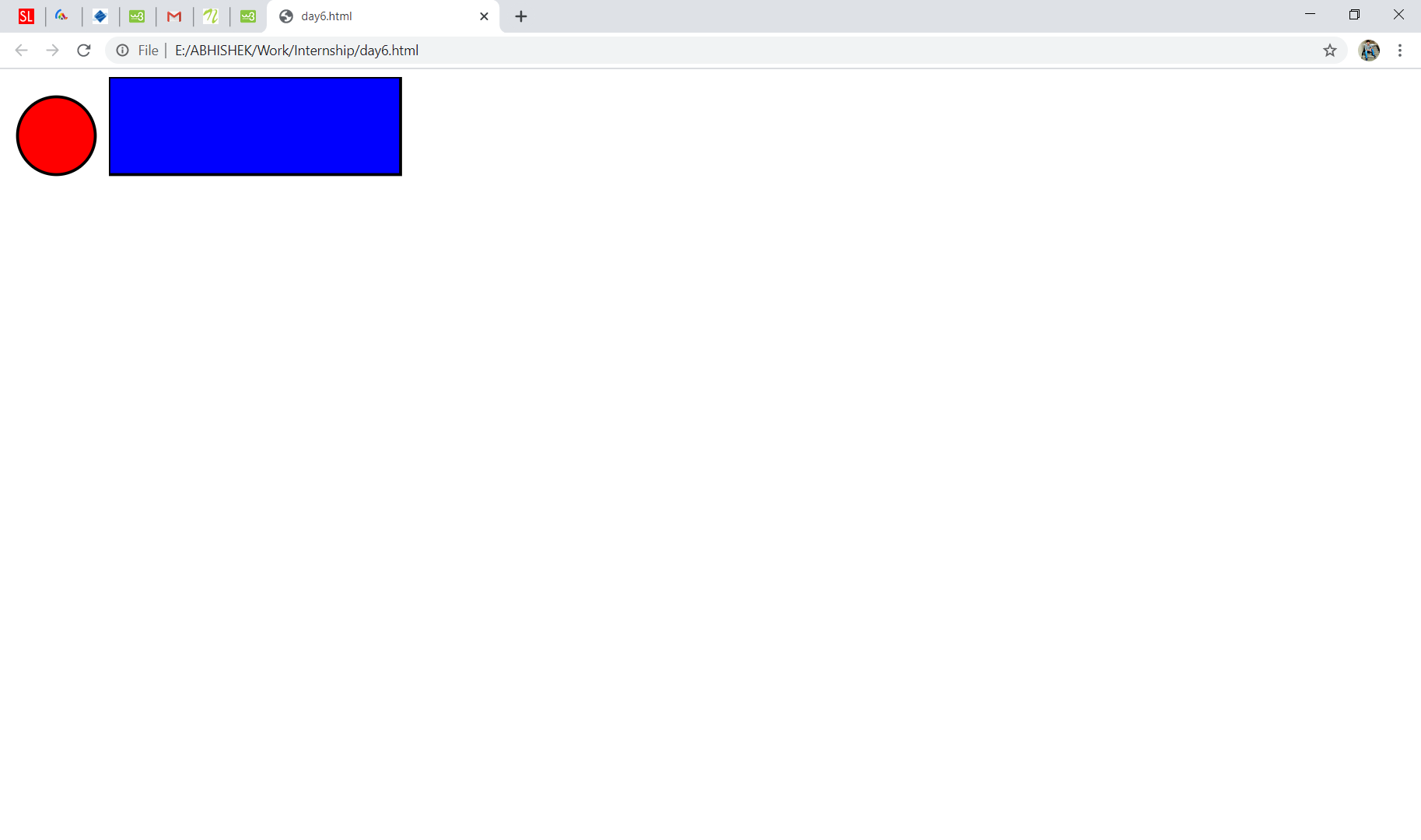
<rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>



**Q5.**

**Ans.**

The HTML Geolocation API is used to get the geographical position of a user.

Geolocation is most accurate for devices with GPS, like smartphone.

The example below returns the latitude and longitude of the user's position:

* Check if Geolocation is supported
* If supported, run the getCurrentPosition() method. If not, display a message to the user
* If the getCurrentPosition() method is successful, it returns a coordinates object to the function specified in the parameter (showPosition)
* The showPosition() function outputs the Latitude and Longitude

The second parameter of the getCurrentPosition() method is used to handle errors. It specifies a function to run if it fails to get the user's location:

<!DOCTYPE html>

<html>

<body>

<p>Click to get your coordinates.</p>

<button onclick="getLocation()">Get Your Location</button>

<script>

var x = document.getElementById("demo");

function getLocation() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showPosition, showError);

} else {

x.innerHTML = "Geolocation is not supported by this browser.";

}

}

function showPosition(position) {

x.innerHTML = "Latitude: " + position.coords.latitude +

"<br>Longitude: " + position.coords.longitude;

}

function showError(error) {

switch(error.code) {

case error.PERMISSION\_DENIED:

x.innerHTML = "User denied the request for Geolocation."

break;

case error.POSITION\_UNAVAILABLE:

x.innerHTML = "Location information is unavailable."

break;

case error.TIMEOUT:

x.innerHTML = "The request to get user location timed out."

break;

case error.UNKNOWN\_ERROR:

x.innerHTML = "An unknown error occurred."

break;

}

}

</script>

</body>

</html>

