

Name: Konika Khurana	UID: 23BCS12925
Branch: CSE	Section: 23 BCS_FS 622-A
Semester: 5	Date of Performance: 26/08/2025
Subject: FULL STACK-LAB	Subject Code: 23CSP-339

Practice 1:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initialscale=1.0">
  <title>Live Character Counter for Textarea</title>
  <style>  body {    font-
family: Arial, sans-serif;
margin: 20px;
    }
    .container {    border:
1px solid #000;
padding: 15px;    margin-
bottom: 20px;    width:
500px;
    }    textarea {
width: 100%;
height: 100px;
padding: 10px;
font-size: 16px;
resize: none;
    }
    .counter {    margin-
top: 10px;    font-
weight: bold;
    }
  </style>
</head>
<body>
  <div class="container">
    <h3>Live Character Counter</h3>
```

```
<textarea id="textInput" placeholder="Start typing..."></textarea>
<div class="counter">Characters: <span id="charCount">0</span></div>
</div>
```

Script.js

```
<!-- Link external JS file -->
<script src="script.js"></script>
</body>
</html>
```

```
const textarea = document.getElementById("textInput"); const
charCount = document.getElementById("charCount");
```

```
textarea.addEventListener("input", () => {
charCount.textContent = textarea.value.length;
});
```

OUTPUT:

Live Character Counter

Start typing...

Characters: 0

Live Character Counter

Hello from ByteXL

Characters: 17

PRACTICE 2:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dynamic Product Filter</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 20px;
    }
    .container {
      border: 1px solid #333;
      padding: 15px;
      margin-bottom: 20px;
      width: 500px;
    }
    select {
      padding: 6px;
      font-size: 16px;
      margin: 10px 0 20px 10px;
    }
    .product-list {
      display: flex;
      flex-direction: column;
      gap: 10px;
    }
    .product {
      border: 1px solid #ccc;
      padding: 10px;
      border-radius: 5px;
      font-size: 16px;
    }
  </style>
</head>
<body>
  <div class="container">
    <h2>Product List</h2>

    <!-- Dropdown Filter -->
    <label for="categoryFilter"><b>Filter by Category:</b></label>
    <select id="categoryFilter">
      <option value="all">All</option>
      <option value="clothing">Clothing</option>
      <option value="electronics">Electronics</option>
```

```
<option value="books">Books</option>
</select>

<!-- Product List -->
<div class="product-list" id="productList">
  <div class="product" data-category="clothing">T-Shirt</div>
  <div class="product" data-category="clothing">Jeans</div>
  <div class="product" data-category="electronics">Headphones</div>
  <div class="product" data-category="electronics">Smartphone</div>
  <div class="product" data-category="books">Novel</div>
  <div class="product" data-category="books">Cookbook</div>
</div>
</div>

<!-- External JS -->
<script src="script.js"></script>
</body>
</html>
```

script.js

```
const categoryFilter = document.getElementById("categoryFilter"); const
products = document.querySelectorAll(".product");

categoryFilter.addEventListener("change", () => {
  const selectedCategory = categoryFilter.value;

  products.forEach(product => {    if (selectedCategory === "all" ||
product.dataset.category === selectedCategory) {      product.style.display =
"block";
    } else {
      product.style.display = "none";
    }
  });
});
```

OUTPUT:



Product List

Filter by Category: All ▼

T-Shirt

Jeans

Headphones

Smartphone

Novel

Cookbook

Product List

Filter by Category: Clothing ▼

T-Shirt

Jeans

Product List

Filter by Category: Books ▼

Novel

Cookbook

PRACTICE 3:

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SVG Drawing Tool</title>
  <style> body { font-
family: Arial, sans-serif;
margin: 20px;
}
.container { border:
1px solid #333;
padding: 15px;
margin-bottom: 20px;
width: 600px;
} svg { border:
1px solid black;
background: #f9f9f9;
width: 100%; height:
400px;
cursor: crosshair;
}
</style>
</head>
<body>
  <div class="container">
    <h2>SVG Drawing Tool</h2>
    <svg id="drawingCanvas"></svg>
  </div>

  <!-- Link External JS -->
  <script src="script.js"></script>
</body>
</html>
```

script.js const svg =
document.getElementById("drawingCanvas");

let drawing = false;
let currentPath = null;

svg.addEventListener("mousedown", (e) => {
 drawing = true;

```
// Create a new path element
currentPath = document.createElementNS("http://www.w3.org/2000/svg", "path");
currentPath.setAttribute("stroke", "blue"); currentPath.setAttribute("stroke-width",
"2");
currentPath.setAttribute("fill", "none");

// Start path at mouse position  const x =
e.offsetX;  const y = e.offsetY;
currentPath.setAttribute("d", `M ${x} ${y}`);

svg.appendChild(currentPath);
});

svg.addEventListener("mousemove", (e) => {
  if (!drawing) return;

  const x = e.offsetX;
  const y = e.offsetY;

  // Append line segments as the mouse moves
  let d = currentPath.getAttribute("d");
  currentPath.setAttribute("d", `${d} L ${x} ${y}`);
});

svg.addEventListener("mouseup", () => {
  drawing = false;
});
```

OUTPUT:

SVG Drawing Tool

Byte^{XL}