DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CU CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Report File FULL STACK

Student Name: S S Sneha

UID: 23BAI70259

Branch: BE-AIT-CSE

Section/Group: 23AIT-KRG-G2

Semester: 5th

Subject Code: 23CSP-339

Subject Name: Full Stack

Aim

To create an interactive SVG-based drawing tool using DOM and mouse event handling in JavaScript.

Objectives

Create an SVG drawing area in HTML

Handle mouse events to draw shapes (e.g., circles)

Allow dynamic drawing using createElementNS()

Implement optional color selection and undo functionality

Hardware/Software Requirements

Category Requirements

Hardware i3+ CPU, 4GB RAM, 1920x1080 display

Software VS Code, Chrome/Firefox, Live Server

About the Experiment

Concepts Covered:

SVG elements in HTML

Event handling (mousedown, mousemove, mouseup)

Creating SVG elements dynamically with createElementNS

Coordinate tracking and shape rendering

Undo stack implementation

Real-world Applications:

Drawing tools in design apps like Adobe Illustrator or Figma

Vector-based user input on web platforms

Code Implementation

🔷 ***HTML***

```
<svg id="canvas" width="500" height="400" style="border:1px solid black;"></svg>
<button onclick="undo()">Undo</button>
```

🔷 ***JavaScript***

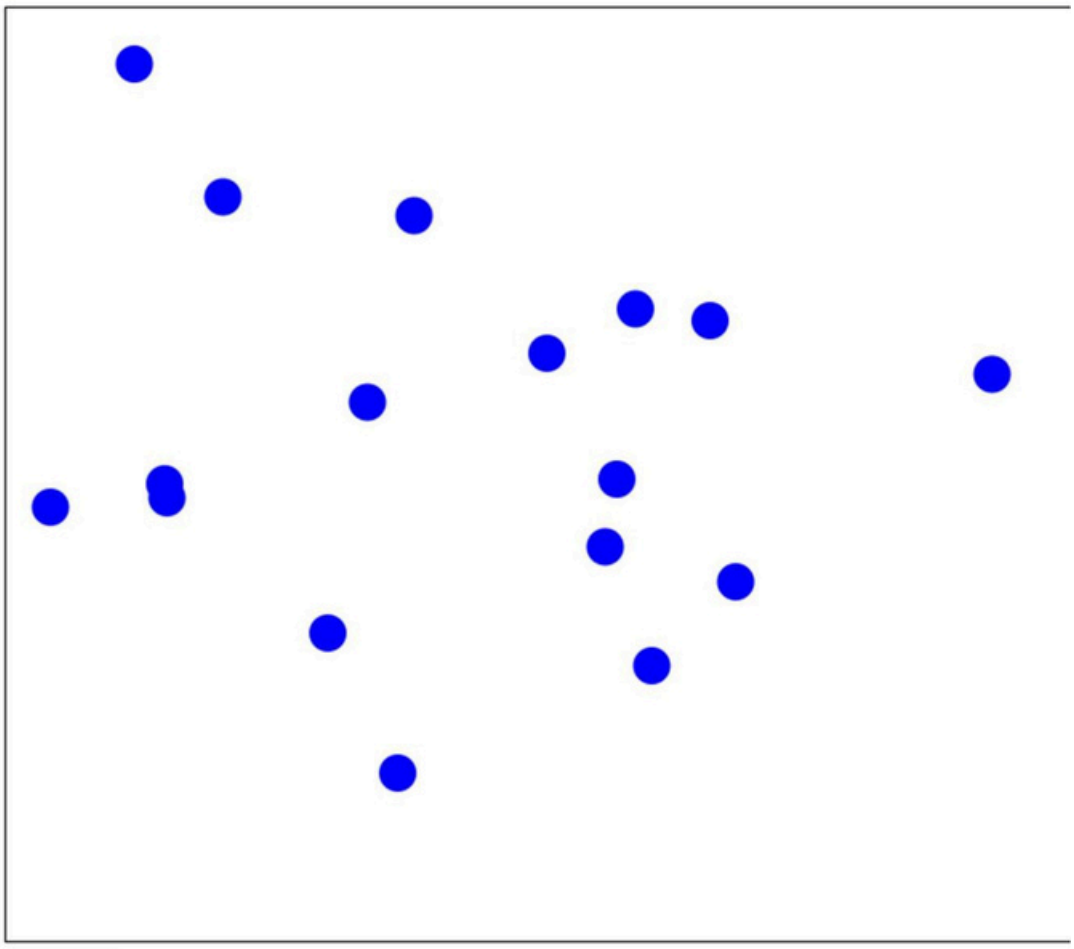
```
const svg = document.getElementById('canvas');
let drawings = [];

svg.addEventListener('mousedown', (e) => {
  const circle = document.createElementNS('http://www.w3.org/2000/svg', 'circle');
  circle.setAttribute('cx', e.offsetX);
  circle.setAttribute('cy', e.offsetY);
  circle.setAttribute('r', '8');
  circle.setAttribute('fill', 'blue');
  svg.appendChild(circle);
  drawings.push(circle);
});

function undo() {
  const last = drawings.pop();
  if (last) {
  svg.removeChild(last);
  }
}
```

Expected Output

A blank SVG canvas
User clicks to draw circles at mouse positions
Circles appear dynamically
Undo button removes the last drawn circle



Undo