

WEB PROGRAMMING

jQuery – Home Assignment

Aditya Agarwal

220905106

Roll No.14

Q1 - 1. Design a webpage. The page contains:

- Dropdown list with HP, Nokia, Samsung, Motorola, Apple as items.
- Checkbox with Mobile and Laptop as items. Textbox where you enter quantity.
- There is a button with text as 'Produce Bill'.

On Clicking Produce Bill button, alert should be displayed with total amount.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Dynamic Product Billing</title>
<style>
body {
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
margin: 0;
padding: 0;
background: linear-gradient(to right, #f5f7fa, #c3cfe2);
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
}

.container {
background: #fff;
padding: 20px;
border-radius: 12px;
box-shadow: 0 4px 10px rgba(0, 0, 0, 0.15);
width: 400px;
}

h2 {
text-align: center;
color: #4b79a1;
}

label {
display: block;
margin: 15px 0 5px;
font-weight: bold;
}
```

```
select, input[type="text"], input[type="checkbox"] {
width: 100%;
padding: 10px;
margin-bottom: 15px;
border: 1px solid #ddd;
border-radius: 8px;
}
```

```
button {
width: 100%;
padding: 15px;
font-size: 16px;
background: linear-gradient(to right, #4b79a1, #283e51);
color: #fff;
border: none;
border-radius: 8px;
cursor: pointer;
transition: transform 0.2s ease, background 0.3s ease;
}
```

```
button:hover {
transform: scale(1.05);
background: linear-gradient(to right, #283e51, #4b79a1);
}
```

```
.error {
color: red;
font-size: 14px;
text-align: center;
}
```

```
</style>
</head>
<body>
<div class="container">
<h2>Product Billing</h2>
<label for="brand">Select Brand:</label>
<select id="brand">
<option value="HP">HP</option>
<option value="Nokia">Nokia</option>
<option value="Samsung">Samsung</option>
<option value="Motorola">Motorola</option>
<option value="Apple">Apple</option>
</select>
```

```
<label for="category">Select Category:</label>
<input type="checkbox" id="mobile" value="Mobile"> Mobile<br>
<input type="checkbox" id="laptop" value="Laptop"> Laptop<br>
```

```
<label for="quantity">Enter Quantity:</label>
<input type="text" id="quantity" placeholder="Enter quantity">
```

```
<div class="error" id="error-message"></div>
```

```
<button onclick="produceBill()">Produce Bill</button>
</div>
```

```
<script>
function produceBill() {
  const brand = document.getElementById('brand').value;
  const mobileChecked = document.getElementById('mobile').checked;
  const laptopChecked = document.getElementById('laptop').checked;
  const quantity = parseInt(document.getElementById('quantity').value);
  const errorMessage = document.getElementById('error-message');
```

```
  errorMessage.textContent = "";
```

```
  if (isNaN(quantity) || quantity <= 0) {
    errorMessage.textContent = 'Please enter a valid quantity.';
    return;
  }
```

```
  let totalAmount = 0;
```

```
  const prices = {
    HP: { mobile: 15000, laptop: 60000 },
    Nokia: { mobile: 12000, laptop: 50000 },
    Samsung: { mobile: 20000, laptop: 65000 },
    Motorola: { mobile: 18000, laptop: 55000 },
    Apple: { mobile: 80000, laptop: 150000 }
  };
```

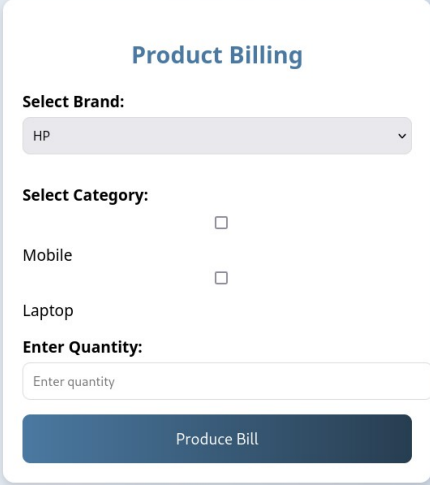
```
  if (mobileChecked) totalAmount += prices[brand].mobile * quantity;
  if (laptopChecked) totalAmount += prices[brand].laptop * quantity;
```

```
  if (!mobileChecked && !laptopChecked) {
    errorMessage.textContent = 'Please select at least one category (Mobile or Laptop).';
    return;
  }
```

```
  alert(`Brand: ${brand}\nQuantity: ${quantity}\nTotal Amount: ₹${totalAmount}`);
}
```

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

OUTPUT:

A screenshot of a web form titled "Product Billing". The form is white with a light blue border and is centered on a light blue background. It contains the following elements: a title "Product Billing" in blue; a "Select Brand:" label above a dropdown menu showing "HP"; a "Select Category:" label above two radio buttons, with "Mobile" and "Laptop" labels to the left; an "Enter Quantity:" label above a text input field containing "Enter quantity"; and a dark blue button labeled "Produce Bill".

Product Billing

Select Brand:
HP

Select Category:
☐ Mobile
☐ Laptop

Enter Quantity:
Enter quantity

Produce Bill

Q2 - Implement the bouncing ball using animate() function.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Bouncing Ball Animation</title>
<style>
/* Set up a gradient background for a more dynamic effect */
body {
margin: 0;
height: 100vh;
display: flex;
justify-content: center;
align-items: center;
background: linear-gradient(to right, #ff7e5f, #feb47b); /* Soft orange gradient */
overflow: hidden; /* Prevents scrolling */
flex-direction: column; /* Align ball and buttons vertically */
}

/* Style the ball with shadow and smooth movement */
#ball {
width: 80px;
height: 80px;
background-color: #ff4747; /* Bright red */
border-radius: 50%;
position: absolute;
top: 50%; /* Position ball in the middle */
```

```
left: 50%;  
margin-left: -40px; /* To center the ball horizontally */  
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2); /* Subtle shadow for 3D effect */  
transform-origin: center; /* Ensure bounce looks centered */  
}
```

```
/* Style the buttons */  
button {  
margin-top: 20px;  
padding: 15px 30px;  
font-size: 18px;  
background-color: #ff4747;  
color: white;  
border: none;  
border-radius: 10px;  
cursor: pointer;  
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.3);  
transition: all 0.3s ease; /* Smooth hover effect */  
}
```

```
/* Add hover effect on button */  
button:hover {  
background-color: #ff2a2a; /* Darker red */  
transform: scale(1.05); /* Slight scale effect */  
}
```

```
/* Add a bouncing shadow effect to make the button more interactive */  
button:active {  
transform: scale(0.95); /* Slight shrink effect on click */  
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.4);  
}
```

```
/* Add some space between buttons */  
.buttons-container {  
display: flex;  
gap: 20px;  
}
```

```
</style>
```

```
<!-- Include jQuery and jQuery UI for additional easing effects -->
```

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

```
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script> <!-- Include jQuery UI -->
```

```
<script>
```

```
$(document).ready(function() {
```

```
var animationRunning = false; // Flag to track animation status
```

```
// Function to make the ball bounce continuously
```

```
function bounceBall() {
```

```
if (animationRunning) return; // Prevent multiple animations from running at the same time
```

```
animationRunning = true; // Set flag to true
```

```

function animateBounce() {
$("#ball").animate(
{ top: '80%' }, // Move ball down
1200, // Duration for downward motion
"easeOutBounce", // Use easeOutBounce for a more natural bounce
function() {
// When the ball reaches the bottom, animate it back up
$(this).animate(
{ top: '50%' }, // Move ball back up
1200, // Duration for upward motion
"easeOutBounce", // Use easeOutBounce for a natural bounce effect
animateBounce // Call animateBounce again for continuous bouncing
);
}
);
}

```

```

animateBounce(); // Start the bouncing animation loop
}

```

```

// Start bouncing when the "Start Bouncing" button is clicked
$("#startButton").click(function() {
bounceBall();
});

```

```

// Stop the bouncing when the "Stop Bouncing" button is clicked
$("#stopButton").click(function() {
$("#ball").stop(true, false); // Stop the current animation
animationRunning = false; // Reset the flag
});
});

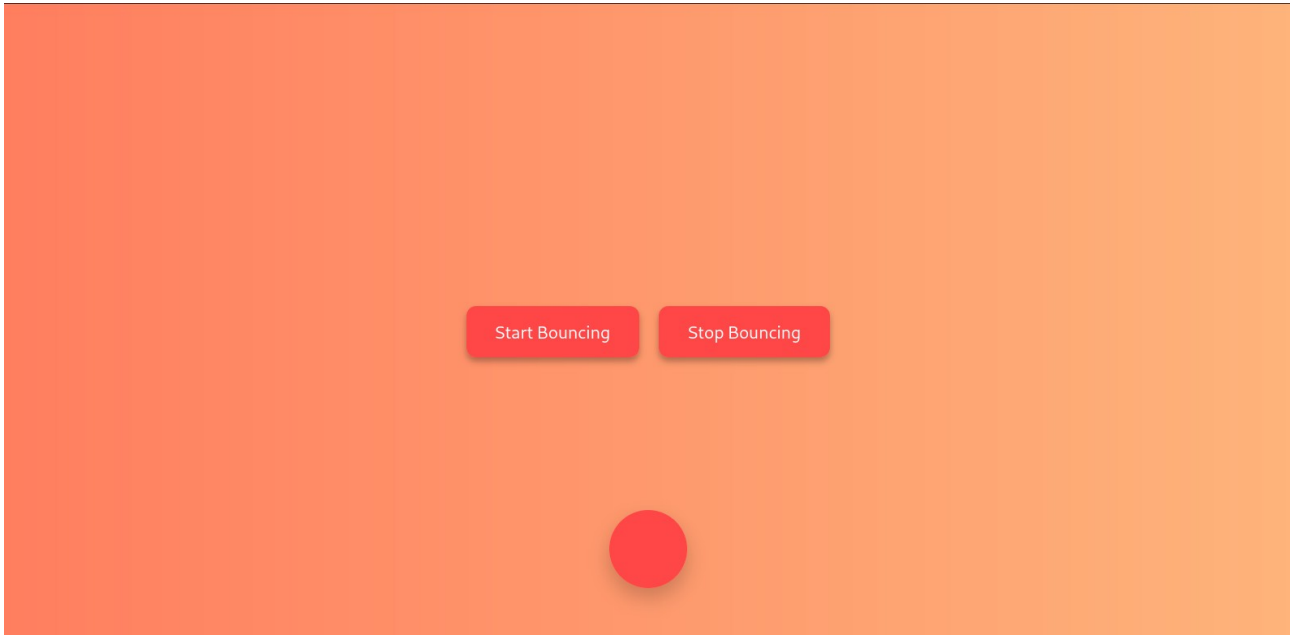
```

```

</script>
</head>
<body>
<!-- Ball Element -->
<div id="ball"></div>
<!-- Container for Buttons -->
<div class="buttons-container">
<!-- Buttons to control the animation -->
<button id="startButton">Start Bouncing</button>
<button id="stopButton">Stop Bouncing</button>
</div>
</body>
</html>

```

OUTPUT:



Q3 - Write a web page which displays image and show the sliding text on the image.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Sliding Text on Image</title>
<style>
/* Style the body to center the content */
body {
margin: 0;
font-family: Arial, sans-serif;
}

/* Set the image as background */
.image-container {
position: relative;
width: 100%;
height: 100vh;
background-image: url('gradient.jpg'); /* Replace with your image URL */
background-size: cover;
background-position: center;
overflow: hidden;
}

/* Sliding text container */
.sliding-text {
position: absolute;
bottom: 20px; /* Position it near the bottom of the image */
```

```

left: 0;
font-size: 2rem;
color: white;
font-weight: bold;
padding: 10px;
background-color: rgba(0, 0, 0, 0.5); /* Semi-transparent background */
width: 100%;
text-align: center;
transform: translateX(100%); /* Initially position text outside the image */
animation: slideIn 5s infinite; /* Sliding animation */
}

/* Keyframes for sliding text animation */
@keyframes slideIn {
0% {
transform: translateX(100%); /* Start from the right */
}
50% {
transform: translateX(0); /* Slide to its normal position */
}
100% {
transform: translateX(-100%); /* Slide out to the left */
}
}

</style>
</head>
<body>
<!-- Image container -->
<div class="image-container">
<!-- Sliding text -->
<div class="sliding-text">
This is sliding text on the image!
</div>
</div>
</body>
</html>

```

OUTPUT:

The image features a dark gradient background that transitions from a deep purple on the left to a bright red on the right. A horizontal band of slightly darker, more uniform color runs across the lower portion of the image, serving as a backdrop for the text.

This is sliding text on the image!