

Experiment 6

AIM: to design a prototype for a tTV control panel

procedure:

- 1.The HTML document creates a mobile keypad interface with buttons enclosed in a <div> container.
- 2.CSS styles are applied to define the layout and appearance of the keypad.
- 3.JavaScript code tracks the pressed keys using an array.
- 4.When a button is clicked, the handleKeyPress() function is invoked, which adds the clicked key to the pressedKeys array.
- 5.The feedback message is updated by joining the elements of the pressedKeys array.

Code:

```
<!DOCTYPE html>

<html>

<head>

  <title>Mobile Keypad</title>

  <style>

    body {

      display: flex;

      justify-content: center;

      align-items: center;

      height: 100vh;

      margin: 0;

    }

    .keypad {

      display: grid;

      grid-template-columns: repeat(3, 1fr);

      gap: 10px;

      width: 200px;
```

```
}  
  
.key {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 50px;  
  background-color: #ccc;  
  border-radius: 5px;  
  cursor: pointer;  
}  
  
#feedback {  
  margin-top: 10px;  
  font-weight: bold;  
  text-align: center;  
}  
  
button {  
  margin-top: 10px;  
}  
  
</style>  
</head>  
<body>  
  <div class="keypad">  
    <div class="key" onclick="handleKeyPress('1')">1</div>  
    <div class="key" onclick="handleKeyPress('2')">2</div>  
    <div class="key" onclick="handleKeyPress('3')">3</div>  
    <div class="key" onclick="handleKeyPress('4')">4</div>  
    <div class="key" onclick="handleKeyPress('5')">5</div>  
    <div class="key" onclick="handleKeyPress('6')">6</div>  
    <div class="key" onclick="handleKeyPress('7')">7</div>  
    <div class="key" onclick="handleKeyPress('8')">8</div>  
    <div class="key" onclick="handleKeyPress('9')">9</div>
```

```
<div class="key" onclick="handleKeyPress('*")">*</div>
<div class="key" onclick="handleKeyPress('0')">0</div>
<div class="key" onclick="handleKeyPress('#')">#</div>
</div>
```

```
<button onclick="deleteLastKeyPressed()">Delete</button>
<p id="feedback"></p>
```

```
<script>
```

```
  var pressedKeys = [];
```

```
  function handleKeyPress(key) {
    pressedKeys.push(key);
    updateFeedback();
  }
```

```
  function deleteLastKeyPressed() {
    pressedKeys.pop();
    updateFeedback();
  }
```

```
  function updateFeedback() {
    document.getElementById("feedback").innerText = "Pressed keys: " + pressedKeys.join("");
  }
```

```
</script>
```

```
</body>
```

```
</html>
```



result:

thus, the TV remote prototype is executed successfully