**NAME: Sharvayu Zade**

**PRN: 23070521135**

**SEC: B(B2)**

**Practical 3 Part 3**

**Ways to Take Input in JavaScript**

JavaScript can collect input from users using various methods:

**1. HTML Form Elements (via DOM)**

This is the **most common method** for user input in web applications. **Example:**

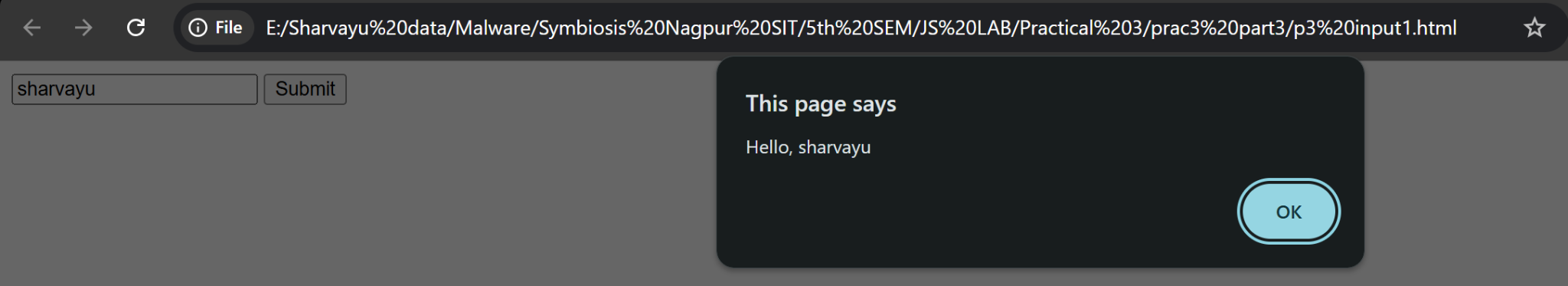
<input type="text" id="name" />

<button onclick="getName()">Submit</button> <script>

function getName() {

let name = document.getElementById("name").value; alert("Hello, " + name);

}

</script> 

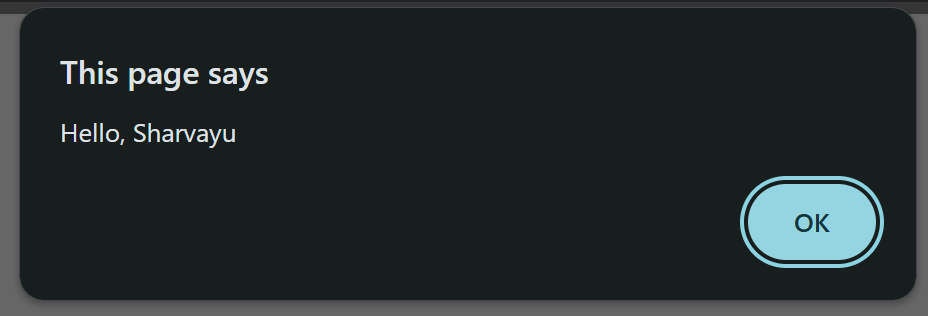
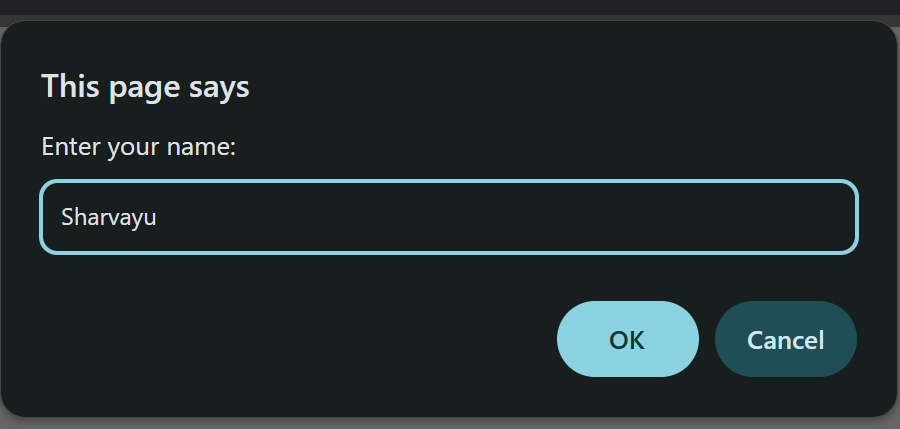
**Input Types:**

| **Type** | **Description** |
| --- | --- |
| text | Single line input |
| number | Numeric input only |
| password | Hidden characters |
| email | Valid email format |

| radio | Single choice from group |
| --- | --- |
| checkbox | Multiple selections |
| date | Calendar input |
| range | Slider input |
| file | Upload files |
| textarea | Multi-line input |
| select | Drop-down menu |

**2. prompt() (Browser Dialog Box)**

Useful for quick input during testing or simple applications.

let name = prompt("Enter your name:"); alert("Hello, " + name); 

**3. Event Listeners (Real-time Input Capture)**

You can capture input as the user types using events like keyup, change, etc.

**Syntax**

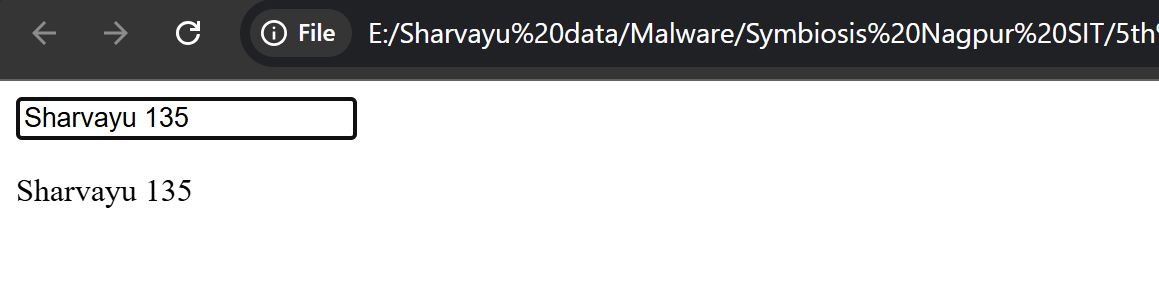
document.addEventListener(*event*, *function*, *Capture*)

<input type="text" id="liveInput" placeholder="Type something"> <p id="display"></p>

<script>

document.getElementById("liveInput").addEventListener("keyup", function() {

document.getElementById("display").innerText = this.value; });

</script> 

**4. Form Submission with submit Event**

Ideal for processing form data in a structured way.

<form id="myForm">

<input type="number" name="marks" />

<button type="submit">Submit</button>

</form>

<script>

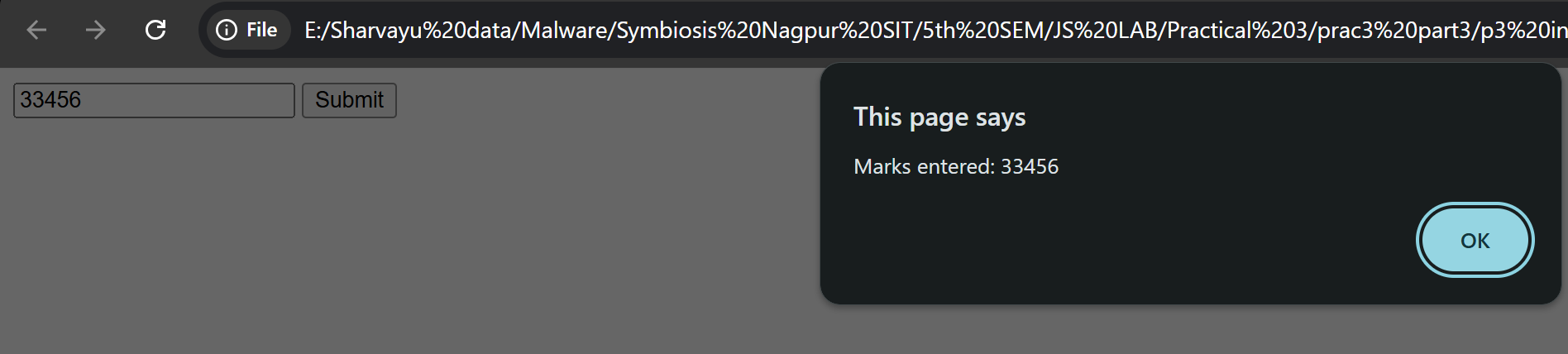
document.getElementById("myForm").addEventListener( "submit", function(e) {

e.preventDefault(); // Prevent reload

let marks = this.marks.value;

alert("Marks entered: " + marks);

});

</script> 

**5. File Input (Read File Data)**

Used when the user uploads a file.

<input type="file" id="fileInput">

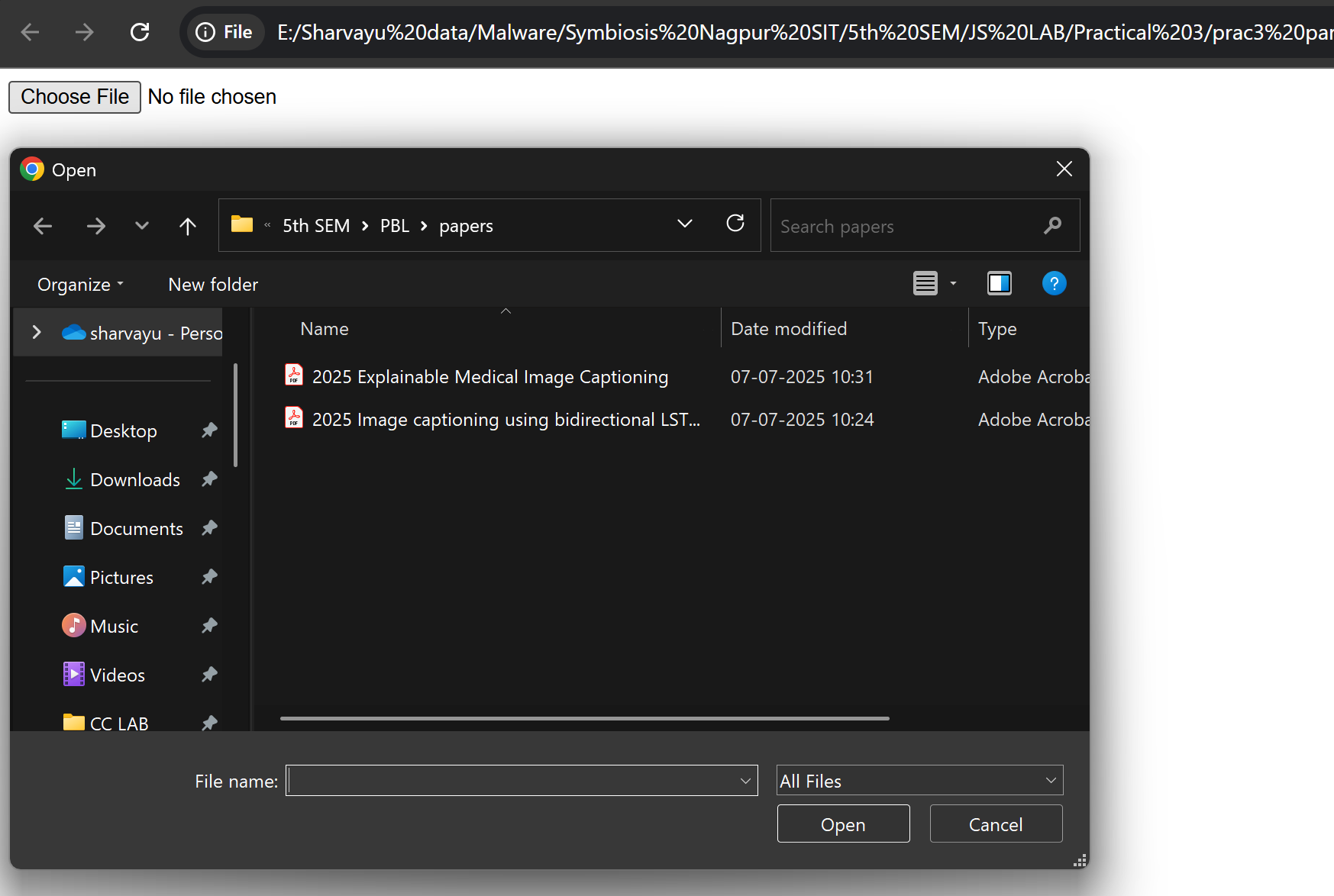
<script>

document.getElementById("fileInput").addEventListener( "change", function() {

const file = this.files[0];

alert("Selected file: " + file.name);

});

</script> 

**6. URLSearchParams (Query Parameters)**

If values are passed through the URL.

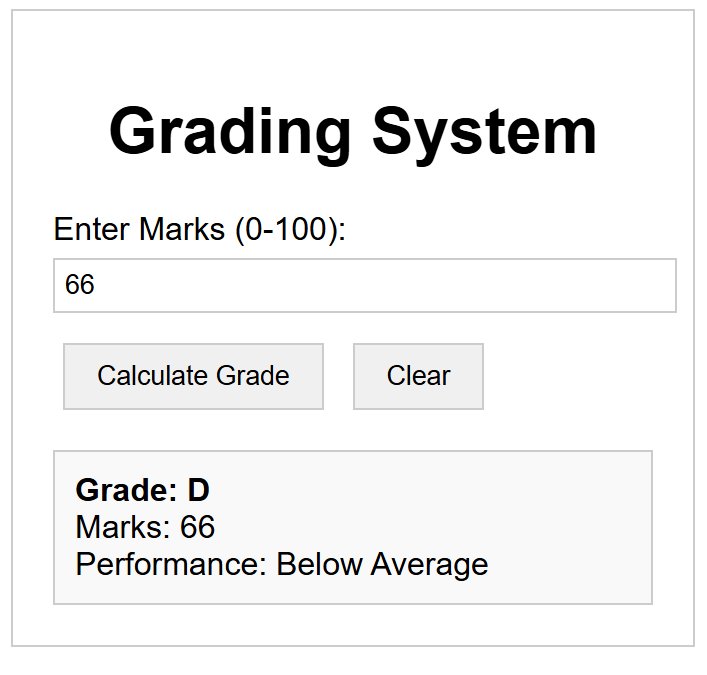
const params = new

URLSearchParams(window.location.search); const name = params.get('name'); // ?name=John

**Input controls Details**

| **Method** | **Use Case Example** | **Real**  **Time?** | **Common**  **In** |
| --- | --- | --- | --- |
| prompt() | Simple text input (testing) | No | Quick apps |
| input elements | Text, number, password, etc. | Yes | Forms |
| textarea | Multi-line input | Yes | Feedback |
| select / option | Dropdown selections | Yes | Menus |
| radio / checkbox | Multiple/Single options | Yes | Surveys |
| file input | Upload documents/images | Yes | Uploads |
| URLSearchParams | Get data from URL | No | Routing |
| Event Listeners  (keyup, change) | Capture typing in real-time | Yes | Live  preview |

**Task:**

**Create a grading system that classifies grades based on marks entered by the user.**