

VALIDATE DATE

Q. Validate Date : it should accept only today date and before days date but not tomorrow date when we enter tomorrow date it should display the invalid date

```
<script>
    // Get today's date in the proper format
    const today = new Date();
    const year = today.getFullYear();
    const month = String(today.getMonth() + 1).padStart(2, '0'); //
Month is zero-indexed
    const day = String(today.getDate()).padStart(2, '0');
    const maxDate = `${year}-${month}-${day}`; // Today's date in
YYYY-MM-DD format

    // Set the maximum date in the input field
    const slotDateInput = document.getElementById('slotDate');
    slotDateInput.max = maxDate;

    // Handle form submission
    const form = document.getElementById('dateForm');
    form.addEventListener('submit', function(event) {
        event.preventDefault(); // Prevent form submission for this demo

        const selectedDate = slotDateInput.value;
        if (selectedDate) {
            alert(`You selected: ${selectedDate}`);
        } else {
            alert('Please select a valid date.');
```

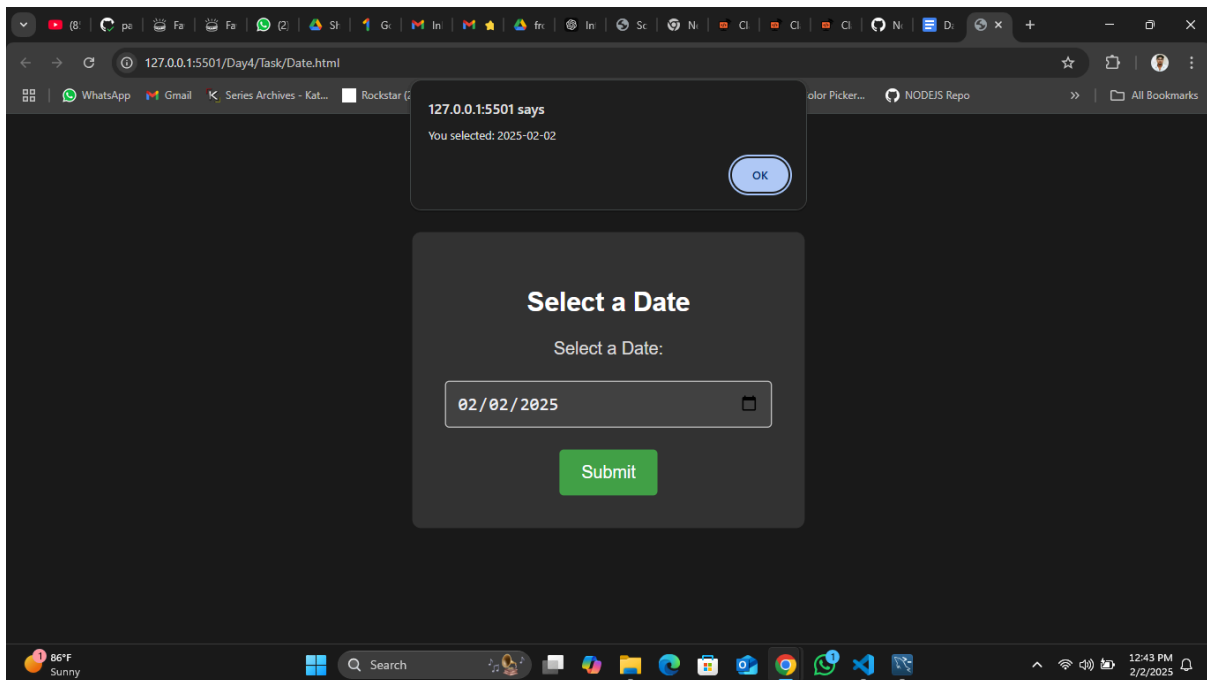
JavaScript calculates today's date in **YYYY-MM-DD** format.

This value is set as the maximum allowed date for the input field using the **max** attribute (**slotDateInput.max = maxDate**).

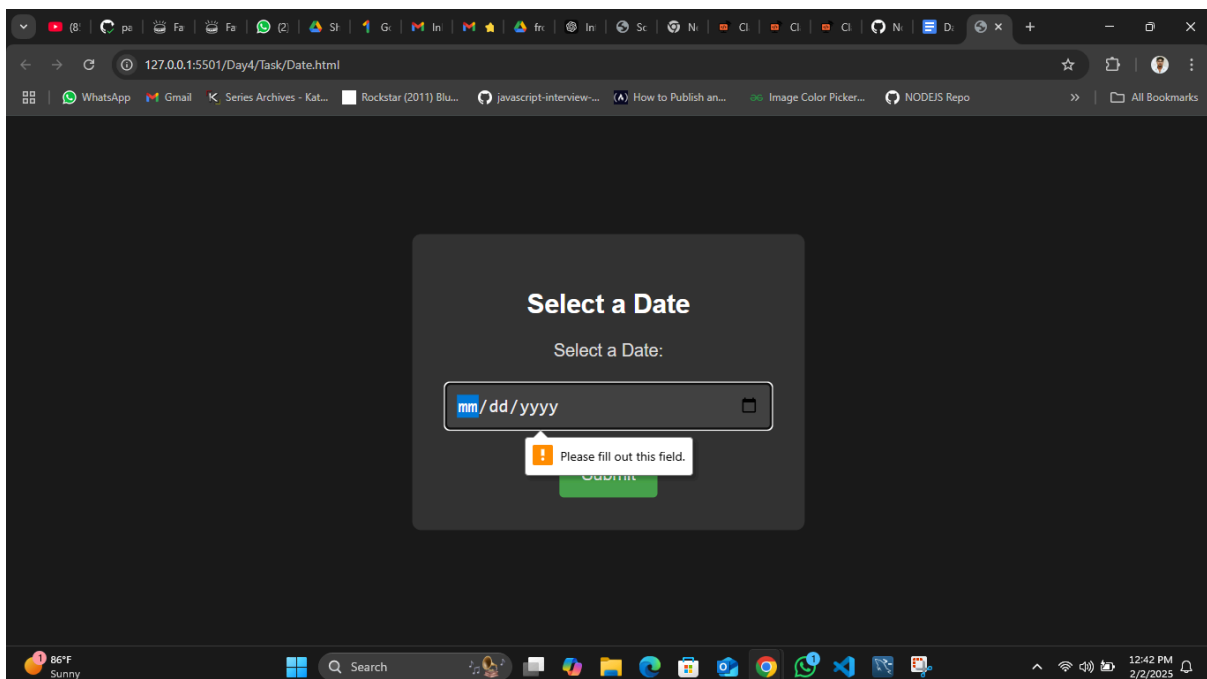
When the form is submitted, JavaScript prevents the default form submission using **event.preventDefault()**.

It checks if a valid date has been selected:

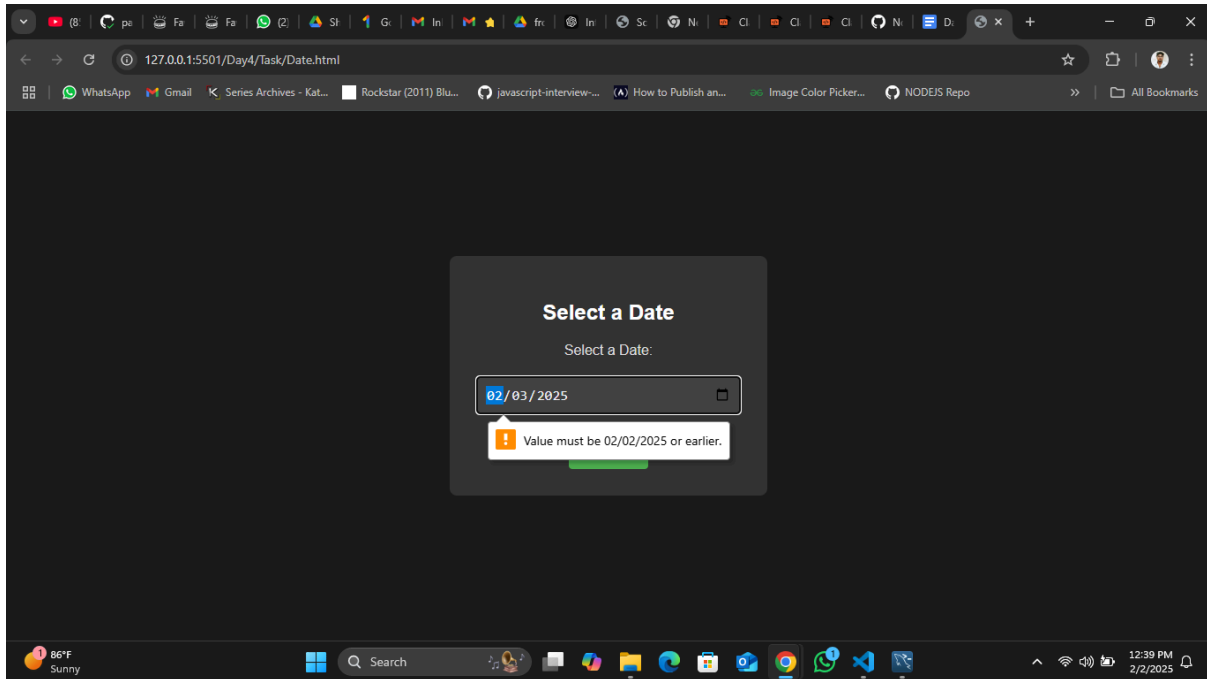
If a date is selected, an alert is shown with the selected date.



If no date is selected, an alert asks the user to please fill out this field.



If a date is selected, and not a valid date an alert asks the user to select a valid date.



VALIDATE EMAIL:

Q. Validate Email: If it ends with another words output should not valid mail id. If it ends with @gmail.com the output should be valid email

HTML Form:

```
<form onsubmit="event.preventDefault(); validateEmail();">
  <h2>Email Validation</h2>
  <label for="email">Enter your email:</label><br>
  <input type="email" id="email" name="email" required><br><br>
  <input type="submit" value="Validate Email">
</form>
```

- The `<form>` element contains an `onsubmit` attribute. Instead of submitting the form, it prevents the default form submission behavior using `event.preventDefault()` and calls the JavaScript function `validateEmail()`.
- The `<label>` gives context for the email input field.
- The `<input>` field is where the user enters their email. It has a `required` attribute, meaning the user cannot submit an empty email.
- The submit button triggers the email validation.

JavaScript Email Validation:

```
function validateEmail() {
  // Get the email input value
  var email = document.getElementById('email').value;
```

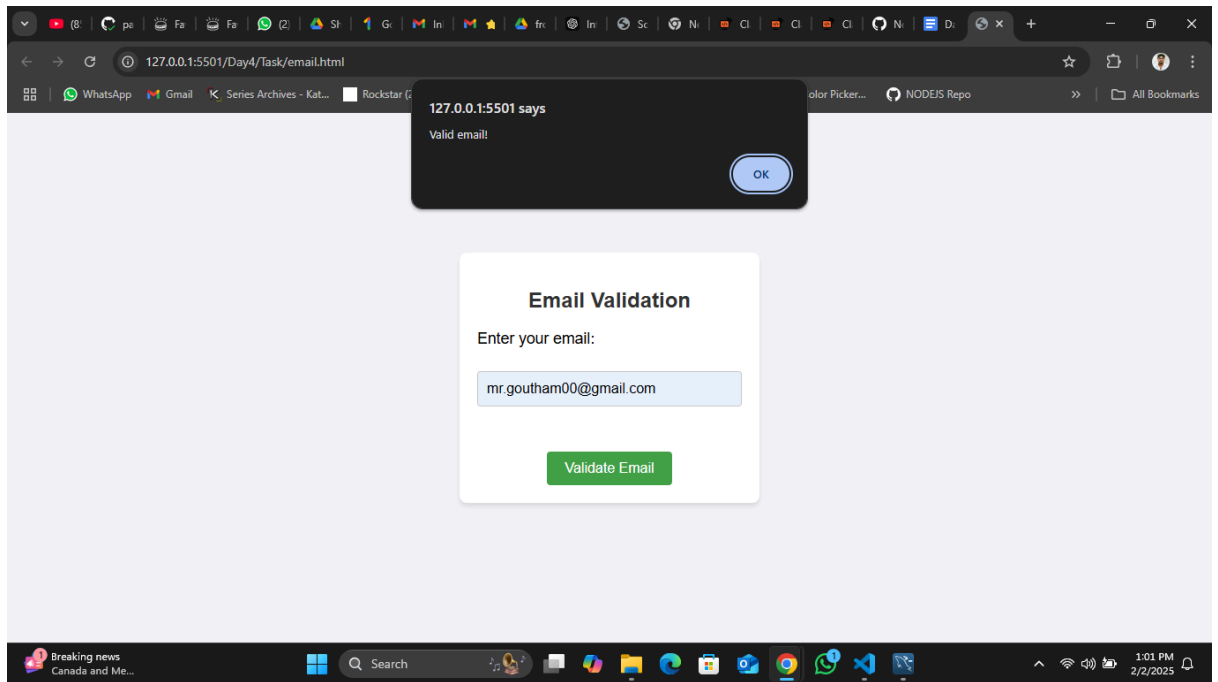
```
// Check if the email ends with '@gmail.com'
if (email.endsWith('@gmail.com')) {
    alert("Valid email!");
} else {
    alert("Invalid email! The email should end with
@gmail.com.");
}
}
```

- Get the email value: The `document.getElementById('email').value` gets the value entered in the email input field.
- Email Validation: The `endsWith('@gmail.com')` method checks if the email ends with "@gmail.com".
- If true, an alert pops up saying "Valid email!".
- If false, an alert pops up saying "Invalid email! The email should end with @gmail.com."

If a user enters an email in the input field.

Upon clicking Submit, the `validateEmail()` function is called.

If the email ends with "@gmail.com", an alert shows "Valid email!".



If the email does not end with "@gmail.com", an alert shows "Invalid email! The email should end with @gmail.com."

