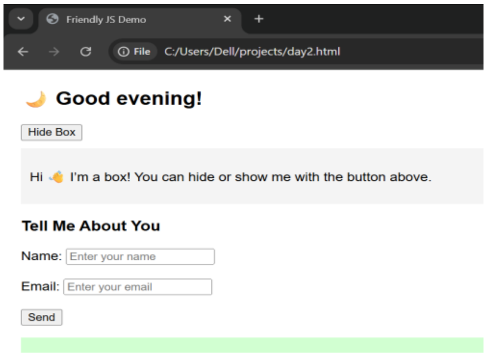
AeroAspire -SDE Intern Training

John Nikhil G

Day 2-Week1

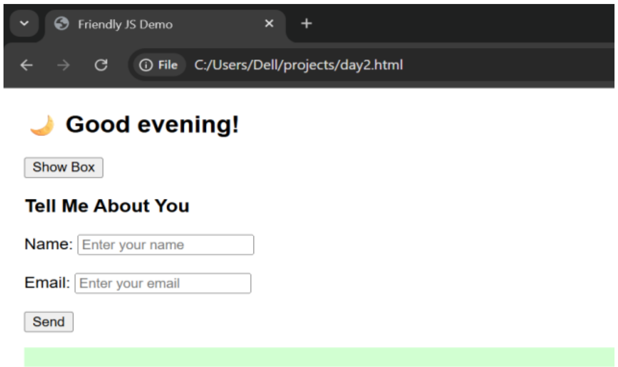
- **The toggle section is closed which says (show box) and when it is opened it says(hide box).**
- **Here the use of a simple toggle button is used in order to create an option of either on/off type condition.**



🌙 Good evening!

Hi 👋 I'm a box! You can hide or show me with the button above.

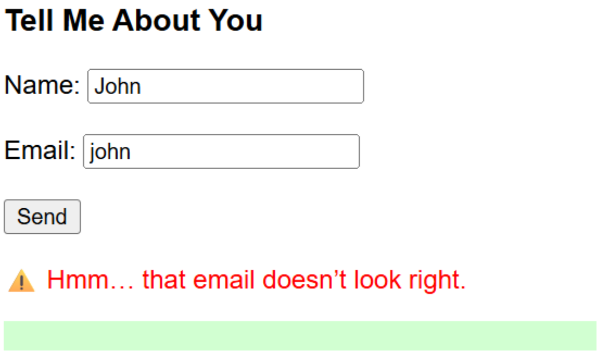
Tell Me About You



🌙 Good evening!

Tell Me About You

- **About the content supposed to fill in, there are two boxes which are name and email.**
- **The condition is if only either one of them is filled it shows an error that says enter input in the box which has no content.**
- **There is also a condition present in the content filled in the email box, if (@) is not present in the box it declines the input and returns output as to fill in the individual′ s email correctly.**

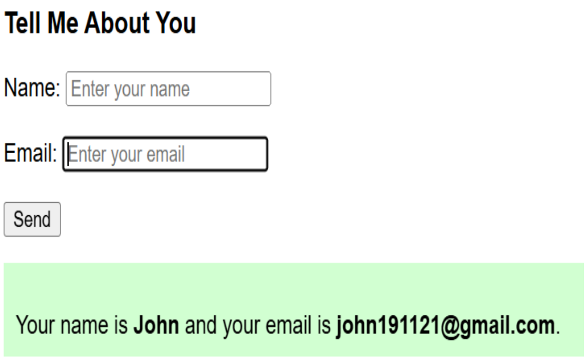


Tell Me About You

Name: John

Email: john

⚠ Hmm… that email doesn’t look right.



Tell Me About You

Name: Enter your name

Email: Enter your email

Your name is **John** and your email is **john191121@gmail.com**.

QUESTIONS:

Q: Difference between let, const, var?

- let and const are block-scoped, whereas var is function-scoped.
- With let and const, you must declare before using them, but var can be used even before declaration.
- Another difference is that let and const don't bind to the this keyword, but var does.
- Also, var gets hoisted, while let and const don't.

Q: What is an event listener?
An event listener is simply a way to tell JavaScript: *"When this event happens, run this function."*
So when an event like a click, hover, or keypress occurs, the listener catches it and triggers the function you wrote.

Q:Walkthrough when you click a button:-

- You click → an event (click) happens.
- The event listener detects that click.
- It then calls the function you linked to it.
- The function runs, and the work gets done.

Q: How does JS know which DOM element, which listener, and which callback to use?
JavaScript maintains a list of event listeners tied to each DOM element. When the event fires, it looks up the list and runs the specific function you attached for that event.

Q: What's the difference between event capturing and bubbling?

- **Capturing:** The event moves from the top (document) down to the target element.
- **Bubbling:** The event starts at the target element and then travels upward to its parent elements.

Q: How do you debug JS errors in browser dev tools?
Open **Inspect → Console tab** in the browser. You'll see errors, warnings, and logs.
You can also go to the **Sources tab** to set breakpoints, check variable values, and see event listeners in action.

Q: What is event.preventDefault() and when do you use it?
event.preventDefault() stops the browser's default action.
Example: If you click a link, normally it would take you to another page. If you use preventDefault(), you can stop that and maybe run your own code first. Same for forms—it prevents automatic submission so you can validate the data before sending.

Q: How do you validate form fields in JS (required, email format, etc.)?

- Check if the required fields have values.
- For email, ensure it has the proper format (like containing @).
- If a field is empty or invalid, return false or show an error message.

For example, in my code I checked email validity using email.includes("@").

Q: Describe the data flow from user input → validation → UI feedback.

1. User types data into the form.
2. The code checks if the data is valid (required fields filled, email has the right format, etc.).
3. If invalid → show an error message or highlight the field.
4. If valid → continue with normal flow (like showing success message or processing data).
   The UI feedback could be a red error message or some CSS changes to catch the user's attention.