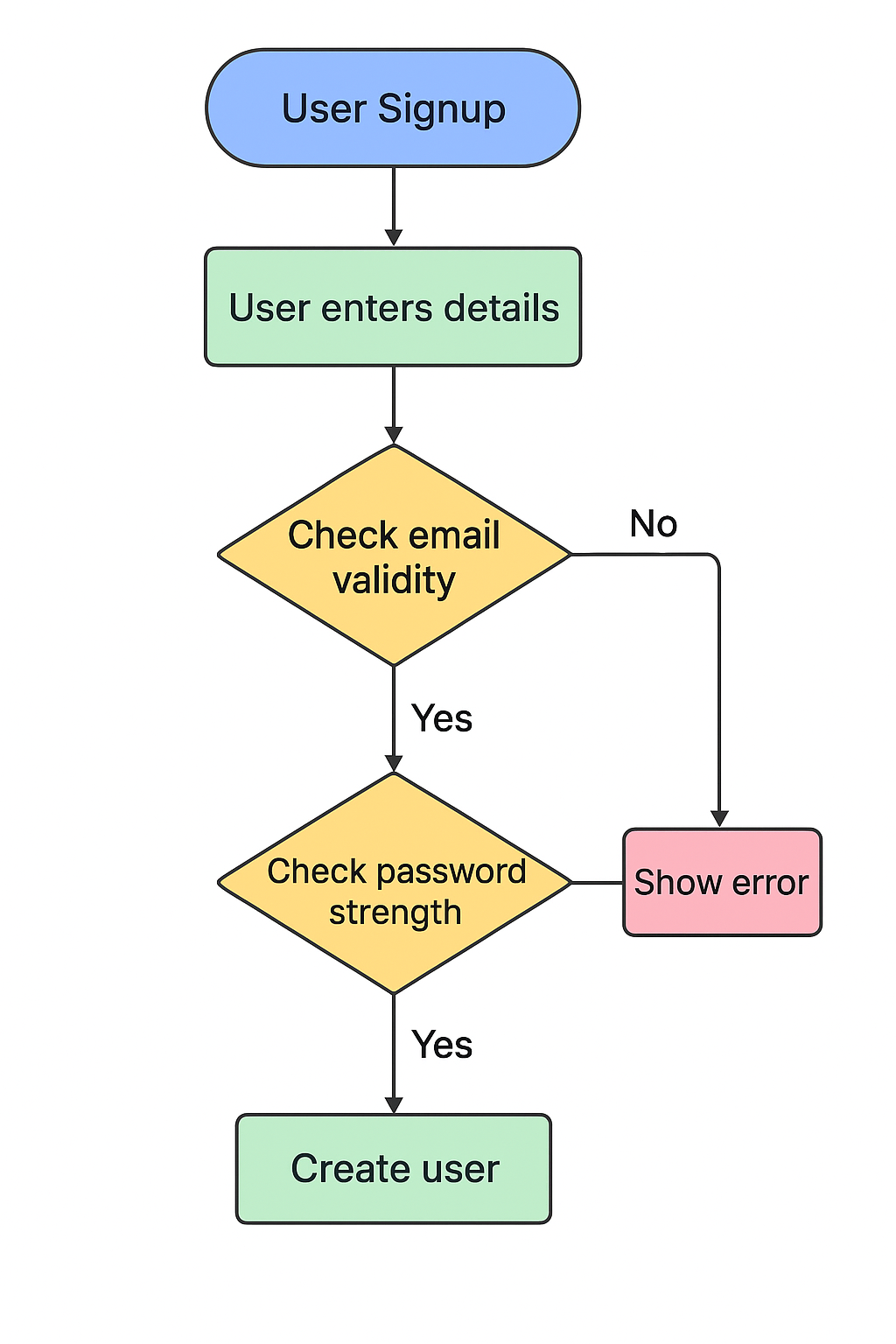
**System Design**

1. **Draw a flowchart for the “User Signup Process”.**



1. **Read about JavaScript Data Types (primitive vs non-primitive).**

* Primitive Data Types (immutable, stored by value):
* String → "Hello"
* Number → 42, 3.14
* Boolean → true, false
* Null → null (intentional empty value)
* Undefined → variable declared but not assigned
* Symbol → unique identifiers
* BigInt → very large integers (123n)
* Primitives are compared by value.
* Primitives = immutable, by value.
* Non-Primitive Data Types (mutable, stored by reference):
* Object → {name: "John"}
* Array → [1,2,3]
* Function → function() { ... }
* Date, RegExp, etc.
* Non-primitives are compared by reference (two different objects with the same content are not equal).
* Non-primitives = objects, by reference.

1. **Understand the difference between == and ===.**

== → loose equality (performs type coercion before comparing).

=== → strict equality (compares value + type, no coercion).

5 == "5" // true (string "5" is coerced into number 5)

5 === "5" // false (number vs string, types different)

1. **Note down 3 examples where type coercion happens.**

* Type coercion often happens in math operations, string concatenation, and loose equality checks.

1. String + Number : console.log("5" + 1); // "51" (number converted to string)
2. Boolean in Math :console.log(true + 1); // 2 (true → 1, false → 0)
3. Equality Check : console.log(0 == false); // true (0 coerced to false)
   * + - 1. console.log(" " == 0); // true (empty string coerced to 0)