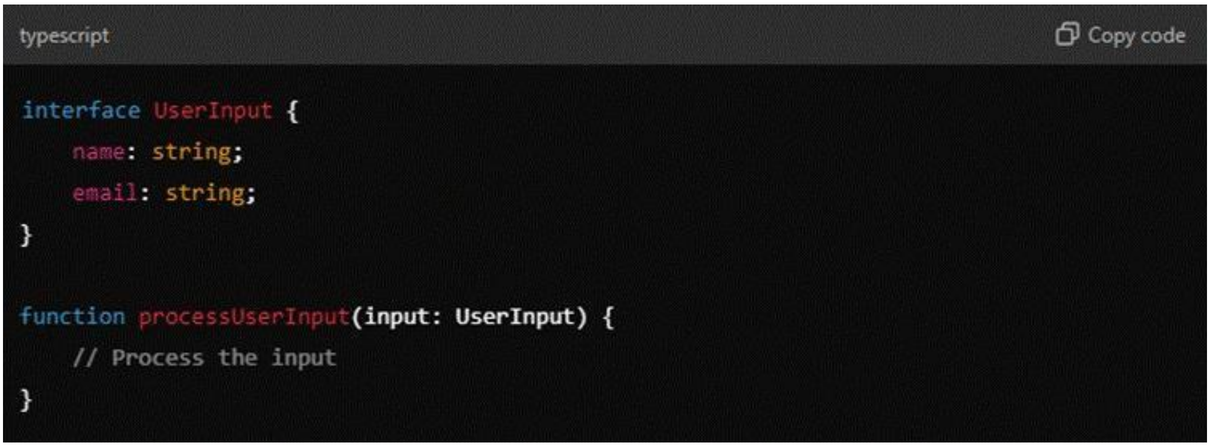
Name – harshali deore
Branch – MCA
Email – harshalideore2002@gmail.com

**8. Data Validation in TypeScript**

**Scenario:** You have the following TypeScript function that processes user input:



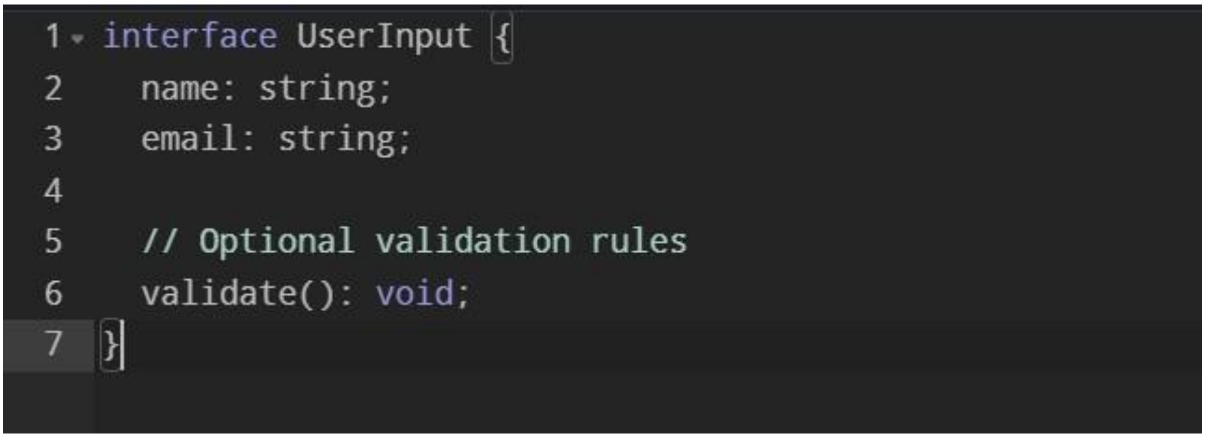
```typescript
interface UserInput {
    name: string;
    email: string;
}

function processUserInput(input: UserInput) {
    // Process the input
}
```

1. How would you enhance the UserInput interface to include validation rules (e.g., email format)?
2. Describe a strategy for validating user input in a TypeScript application.

------>

1) To add validation rules to the **UserInput** interface, we can use TypeScript's features to ensure that the **name** and **email** properties follow certain rules.



```typescript
interface UserInput {
  name: string;
  email: string;

  // Optional validation rules
  validate(): void;
}
```

The validate method can be added to the interface as an optional function that performs custom validation checks. You can implement this method to ensure that the name and email properties adhere to specific requirements, such as:

- **Name:** Non-empty string, we might want to ensure it is always a string with a minimum or maximum length.
- **Email:** Valid email format using regular expressions or libraries like validator.

2) **Define What We Expect**: First, we create a structure (like an interface) that tells us what kind of data we want. For example, if we're asking for a user’s name and email, we say that both should be strings.
**Check the Input**: When the user gives us their input, we check to make sure it matches what we expect. For example, we make sure the name is a string, not a number, and the email is in the correct format.
**Add Extra Rules**: We can add more rules to make sure the input is valid. For example:
The name shouldn't be too short or too long.
The email should have the "@" symbol to be a proper email.
**Handle Mistakes**: If the input doesn't follow the rules, we tell the user what's wrong and ask them to fix it.
**Use Tools**: We can also use tools or libraries that make this process easier by automatically checking if the input meets our rules.