

Name – Deore harshali

Branch – MCA

Email – harshalideore2002@gmail.com

TypeScript Type Inference

Scenario: Consider the following TypeScript function:

```
typescript Copy code  
  
function getUserInfo(userId: number): Promise<{ name: string; age: number }> {  
  // Simulate an async API call  
  return fetch(`/api/users/${userId}`)  
    .then(response => response.json());  
}
```

1. What TypeScript feature is being used to specify the return type of the `getUserInfo` function?
2. How would you modify this function to handle cases where the API response might be missing the `age` property.

- 1) The typescript feature that is being used here is to specify the return type of `getUserInfo` function is type inference. Type inference allows compiler to automatically detect the type of a variable or expression based on its usage and context.
Here , the type of the function's return type is inferred from the type of the value returned by the then chain, which is a `Promise<{ name: string; age: number }>`

- 2) To handle the missing age property , we can make age property as optional .

`Promise<{ name: string; age?: number }>`

We can extend the then chain to include a callback which will check id age property is present in response data.

If the age is missing we can give a warning like message . SO here API will handle the missing Age propert and give response properly.