Module 3

Exercise 6(TypeScript)

Q1.) Define an interface User with the following properties:

- id(number), name(string), email(string), age(number optional)

A1.) Ts

```ts
interface User{
    id:number,
    name:string,
    email:string,
    age?:number
}
```

Implementation

```ts
class UserManager{
    private users :User[]
    constructor(){
        this.users=[]
    }
    addUser(user:User):void{
```

```ts
const user:User={
    id:2,
    name:"Dhruv",
    email:"dhruv@mail.com",
    age:20
}
```

Q2.) Create a class UserManager with:

- A private array users: User[] to store user data.
- A method addUser(user: User): void that adds a new user.
- A method removeUser(id: number): void that removes a user by ID.
- A method getUser(id: number): User | undefined that retrieves a user by ID.
- A method getAllUsers(): User[] that returns all users.

A2.) Ts

```
class UserManager{
    private users :User[]
    constructor(){
        this.users=[]
    }
    addUser(user:User):void{
        this.users.push(user);
    }
    removeUser(id:number):void{
        this.users.forEach(user=>{
            if(user.id===id){
                const index=this.users.indexOf(user);
                this.users.splice(index,1);
            }
        })
    }
    getUser(id:number):User|undefined{
        for(const user of this.users){
            if(user.id===id) return user;
        }
        return
    }

    getAllUser():User[]{
        return this.users;
    }

}
```

```
const usermanager=new UserManager();

usermanager.addUser({
    id:1,
    name:"Dhruv",
    email:"dhruv@gmail.com",
    age:20
})



usermanager.addUser({
    id:2,
    name:"Aman",
    email:"aman@gmail.com",
    age:21
})


usermanager.addUser({
    id:3,
    name:"Gaurav",
    email:"gaurav@gmail.com",
    age:22
})


usermanager.addUser({
    id:4,
    name:"Tushar",
    email:"tushar@gmail.com",
})


console.log(usermanager.getAllUser());
```

```
console.log(usermanager.getUser(2));

usermanager.removeUser(3);

console.log(usermanager.getAllUser());
```

Output

```
dhruv-pandey@TTNPL-dhruv:Exercise6 (master)$npm run dev

> exercise6@1.0.0 dev
> npx tsc
```

```
};
var UserManager = /** @class */ (function () {
    function UserManager() {
        this.users = [];
    }
    UserManager.prototype.addUser = function (user) {
        this.users.push(user);
    };
    UserManager.prototype.removeUser = function (id) {
        var _this = this;
        this.users.forEach(function (user) {
            if (user.id === id) {
                var index = _this.users.indexOf(user);
                _this.users.splice(index, 1);
            }
        });
    };
    UserManager.prototype.getUser = function (id) {
        for (var _i = 0, _a = this.users; _i < _a.length; _i++) {
            var user_1 = _a[_i];
            if (user_1.id === id)
                return user_1;
        }
        return;
    };
    UserManager.prototype.getAllUser = function () {
        return this.users;
    };
    return UserManager;
}());
var usermanager = new UserManager();
usermanager.addUser({
    id: 1,
    name: "Dhruv",
    email: "dhruv@gmail.com",
```

```
dhruv-pandey@TTNPL-dhruv:Exercise6 (master)$npm run start
getAllUser()
[
  { id: 1, name: 'Dhruv', email: 'dhruv@gmail.com', age: 20 },
  { id: 2, name: 'Aman', email: 'aman@gmail.com', age: 21 },
  { id: 3, name: 'Gaurav', email: 'gaurav@gmail.com', age: 22 },
  { id: 4, name: 'Tushar', email: 'tushar@gmail.com' }
]
getUser(2)
{ id: 2, name: 'Aman', email: 'aman@gmail.com', age: 21 }
removeUser()
[
  { id: 1, name: 'Dhruv', email: 'dhruv@gmail.com', age: 20 },
  { id: 2, name: 'Aman', email: 'aman@gmail.com', age: 21 },
  { id: 4, name: 'Tushar', email: 'tushar@gmail.com' }
]
```

Q3.) Use Arrow Functions & Default Parameters

- Add a method getUser = (name: string = "Guest"): string that returns a greeting message.

A3.) Ts

```
    greetUser=(name:string="Guest"):string=>{
        return `Welcome ${name}`;
    }
}


const usermanager=new UserManager();
```

```
console.log("greetUser with parameter")
console.log(usermanager.greetUser("Dhruv"));

console.log("greetUser without parameter default used")
console.log(usermanager.greetUser());
```

Output

```
function UserManager() {
    this.greetUser = function (name) {
        if (name === void 0) { name = "Guest"; }
        return "Welcome ".concat(name);
    };
    this.users = [];
}
```

```
greetUser with parameter
Welcome Dhruv
greetUser without parameter default used
Welcome Guest
dhruv-pandey@TTNPL-dhruv:Exercise6 (master)$
```

Q4.) Use Destructuring & Spread Operator

- Create a function printUserDetails(user: User): void that logs user details using object destructuring.

A4.) Ts

```ts
printUserDetails(user:User):void{
    const {id,name,email,age}=user
    console.log(`The Id of user is ${id}`);
    console.log(`The User name is ${name}`);
    console.log(`The email of user is ${email}`);
    if(age!==undefined) console.log(`The age of the user is ${age}`);
}
```

```ts
usermanager.printUserDetails({
    id:5,
    name:"Dhruv",
    email:"dhruv",
    age:20
})


usermanager.printUserDetails({
    id:2,
    name:"Dhruv",
    email:"dhruv",
})
```

Output

```js
UserManager.prototype.printUserDetails = function (user) {
    var id = user.id, name = user.name, email = user.email, age = user.age;
    console.log("The Id of user is ".concat(id));
    console.log("The User name is ".concat(name));
    console.log("The email of user is ".concat(email));
    if (age !== undefined)
        console.log("The age of the user is ".concat(age));
};
```

```
usermanager.printUserDetails({
    id: 5,
    name: "Dhruv",
    email: "dhruv",
    age: 20
});
usermanager.printUserDetails({
    id: 2,
    name: "Dhruv",
    email: "dhruv",
});
```

```
Printing user with age

The Id of user is 5
The User name is Dhruv
The email of user is dhruv
The age of the user is 20

Printing user without age

The Id of user is 2
The User name is Dhruv
The email of user is dhruv
```

In this program we can use spread operator for making a deep copy user then destruct the copyUser we are directly destructing the user in above function