```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Callback & Callback Hell</title>
</head>

<body>

    <h2>Register New User</h2>

    <form id="userForm">

        <input type="text" id="name" placeholder="Enter Name" autocomplete="off" autofocus onfocus="this.value=''" required>
<br><br>

        <input type="number" id="age" placeholder="Enter Age" autocomplete="off" onfocus="this.value=''" required><br><br>

        <input type="text" id="designation" placeholder="Enter Designation" autocomplete="off" onfocus="this.value=''"
required><br><br>

        <button type="submit">Add User</button>

    </form>

    <h2>Registered Users</h2>

    <div id="users"></div>

    <script src="script.js"></script>

</body>

</html>
```

```
/*

==================================

Initial Users Array (Database)

==================================

*/

// We create an array of objects called 'users'. Each object represents one registered user with name, age and designation.

// Check if 'users' already exist in localStorage

const savedUsers = localStorage.getItem("users");

// If 'users' exists in localStorage, use it, otherwise create an empty array

// If found → parse and use them, otherwise use default

const users = savedUsers ? JSON.parse(savedUsers) : [

    { name: "Ajay Suneja", age: 30, designation: "Front End Web Developer" },

    { name: "Anshika Gupta", age: 25, designation: "Back End Web Developer" }
];

/*

==========================

Render Users to the DOM

==========================

*/

```

```
// This function takes the current 'users' array and displays it inside the <div id="users">.

const renderUsers = () => {

    // Get the 'users' container div from HTML

    const usersDiv = document.getElementById("users");

    // Clear the div before re-rendering (so old content doesn't duplicate)

    usersDiv.innerHTML = "";

    // Loop over each user in the 'users' array

    users.forEach((user) => {

        // Append a user card (using template literals to insert values)

        usersDiv.innerHTML += `<div class="user-card">

            <strong>Name:</strong> ${user.name}<br>

            <strong>Age:</strong> ${user.age}<br>

            <strong>Designation:</strong> ${user.designation}

        </div>`
    })
}

/*

==================================

Add User Function (Asynchronous)

==================================

```

```
*/

/*

This function simulates saving a new user (like to a database or server). It accepts:

1) 'newUser' → the new user object to add

2) 'renderUsers' → a callback function to run AFTER the user is added

*/

const addUser = (newUser, renderUsers) => {

    // Simulate server delay using setTimeout (2 seconds)

    setTimeout(() => {

        // Add the new user into the 'users' array

        users.push(newUser);

        // Save the updated 'users' array into localStorage

        localStorage.setItem("users", JSON.stringify(users));

        // Call the callback function (refresh the user list on the page)

        renderUsers();

    }, 2000); // 2000ms = 2 seconds
}

/*

==========================

Initial Render of Users
```

```
114
115 ==========================
116
117 */
118
119 // Call renderUsers() once when the page loads, so that the initial users appear in the "Registered Users" section.
120
121 renderUsers();
122
123 /*
124
125 ==========================
126
127 Form Submission Handling
128
129 ==========================
130
131 */
132
133 // Get the form with id="userForm" and attach an event listener. When user submits the form, run the function below
134
135 document.getElementById("userForm").addEventListener("submit", (e) => {
136
137     // Prevent the form's default behavior (page reload)
138
139     e.preventDefault();
140
141     // Create a new user object with values from the form inputs
142
143     const newUser = {
144
145         name: document.getElementById("name").value,          // name input
146
147         age: document.getElementById("age").value,            // age input
148
149         designation: document.getElementById("designation").value // designation input
150     };
151
```

```
    // Change button text to show "Adding..."

    e.target.querySelector("button").textContent = "Adding...";

    // Disable the button temporarily (prevent multiple submissions while waiting)

    e.target.querySelector("button").disabled = true;

    /*

    Call addUser() function. Pass:

    1) newUser (the object we just created)

    2) a callback function that runs AFTER user is added

    */

    addUser(newUser, () => {

        // Re-render all users (to show the new one on screen)

        renderUsers();

        // Reset form inputs (clear fields)

        e.target.reset();

        // Change button text back to "Add User"

        e.target.querySelector("button").textContent = "Add User";

        // Enable the button again (so user can add more entries)

        e.target.querySelector("button").disabled = false;
    });
});
```