

TASK 1: Form Not Submitting (HTML + JS)

```
<form id="userForm">
<input type="text" id="username">
<button onclick="validate()">Submit</button>
</form>
<script>
function validate() {
let name = document.getElementById("username").value;
if (name == "") {
alert("Name required");
}
}
</script>
```

Correct Code:

```
<form id="userForm">
<input type="text" id="username">
<button type="submit">Submit</button>
</form>
<script>
document.getElementById("userForm").addEventListener("submit", function(e) {
e.preventDefault();
let name = document.getElementById("username").value;
if (name === "") {
alert("Name required");
} else {
}
alert("Form submitted");
});
```

```
</script>
```

TASK 2: CSS Not Applying

Buggy Code:

```
<p class="msg">Hello User</p>
```

```
.msg {  
    color blue;  
    font-size: 20px;  
}
```

Correct Code:

```
.msg {  
    /* color blue; : is not present */  
    color:blue;  
    font-size: 20px;  
}
```

// Task 3

//ID to be modified with Id

```
//<button id="btn">Click Me</button>
```

```
//document.getElementById("btn").onclick = function () {
```

```
// alert("Clicked");
```

Correct Code:

```
document.getElementById("btn").onclick = function () {
```

```
// alert("Clicked");
```

```
};
```

// TASK 4: Validation Logic Error - will not work for 18

```
// let age = 18;  
// if (age > 18) {  
//   console.log("Eligible");  
// } else if (age < 18) {  
//   console.log("Not Eligible");  
// }
```

Correct Code:

```
let age = 18;  
if (age >= 18) {  
  console.log("Eligible");  
} else if (age < 18) {  
  console.log("Not Eligible");  
}
```

// TASK 5: Input Value Always Empty

```
<input type="text" id="email">  
<button onclick="check()">Check</button>  
<script>  
function check() {  
  let email = document.getElementById("email");  
  if (email === "") {  
    alert("Email required");  
  }  
}  
</script>
```

Correct Code:

```
function check() {  
  let email = document.getElementById("email").value;
```

```
if (email === "") {  
  alert("Email required");  
}  
}
```

TASK 6: CSS Layout Breaking

Buggy Code:

```
.card  
  
{  
width: 300;  
padding: 20px;  
}  
  
.card  
  
{  
width: 300px;  
padding: 20px;  
}
```

// TASK 7: Password Match Validation

// Buggy Code:

```
// let pwd = "admin123";  
// let confirmPwd = "admin123 ";  
// if (pwd == confirmPwd) {  
//   console.log("Match");  
// } else {  
//   console.log("Mismatch");  
// } -> use trim to get rid of white spaces
```

Correct Code:

```
let pwd = "admin123";  
let confirmPwd = "admin123 ";
```

```
confirmPwd=confirmPwd.trim(" ");  
if (pwd.trim() == confirmPwd) {  
    console.log("pass Match");  
} else {  
    console.log("pass Mismatch");  
}
```

TASK 8: Hidden Element Still Taking Space (CSS)

Buggy Code:

```
.alert {  
    visibility: hidden;  
}
```

Correct Code:

```
.alert {  
    display:none;  
}
```

// TASK 9: JS Error Stops Script

// Buggy Code:

```
// console.log("Start");  
// alert("Hello");  
// console.log("End"); -> end quote is not added
```

Correct Code:

```
console.log("Start");  
alert("Hello");  
console.log("End");
```

// TASK 10: Checkbox Validation

// Buggy Code:

```
// <input type="checkbox" id="terms">
// <button onclick="submitForm()">Submit</button>
// <script>
// function submitForm() {
//   if (terms.checked == false) {
//     alert("Accept terms");
//   }
// }
// }
// </script>
```

Correct Code:

```
terms=document.getElementById("terms");
function submitForm() {
  if (terms.checked === true) {
    // alert("Accept terms");
    console.log("accept");
    return;
  }
}
```

// TASK 11: Assignment instead of comparison

// Buggy Code:

```
// function validate() {
//   let mobile = document.getElementById("mobile").value;
//   if ((mobile.length = 10)) {
//     alert("Valid");
//   } else {
//     alert("Invalid");
//   }
// }
// }
```

Correct Code:

```
function validate() {  
    let mobile = document.getElementById("mobile").value;  
    if (mobile.length == 10) {  
        alert("Valid");  
    } else {  
        alert("Invalid");  
    }  
}
```

// TASK 12: Policy Holder Name Validation

// Buggy Code:

```
// function validateName() {  
// let name = document.getElementById("policyName");  
// if (name == "") {  
// alert("Policy holder name required");  
// }  
// }
```

Correct Code:

```
function validateName() {  
    let name = document.getElementById("policyName").value;  
    if (name == "") {  
        alert("Policy holder name required");  
    }  
}
```

// TASK 13: Insurance Plan Dropdown Validation

// Buggy Code:

```
// function checkPlan() {  
// let plan = document.getElementById("plan").value;  
// if ((plan = "Select Plan")) {  
// alert("Please choose a plan");  
// }  
// }
```

Correct Code:

```
function checkPlan() {  
    let plan = document.getElementById("plan").value;  
    if ((plan == "Select Plan")) {  
        alert("Please choose a plan");  
    }  
}
```

// TASK 14: Policy Number Display Bug

// Buggy Code:

```
// let policyNumber = 123456;  
// document.getElementById("policy").innerHTML = policyNo;
```

Correct Code:

```
let policyNumber = 123456;  
document.getElementById("policy").innerHTML = policyNumber;
```

// TASK 15: Claim Amount Validation - not working with null

```
// let claimAmount = "abc";  
// if (!isNaN(claimAmount)){  
//     alert("Valid claim");  
// } else {  
//     alert("Invalid claim");  
// }
```



```
// }
```

Correct Code:

```
let claimAmount = 123;

claimAmount = Number(claimAmount);

if (claimAmount && !isNaN(claimAmount)){

    // alert("Valid claim");

    console.log("valid");

} else {

    // alert("Invalid claim");

    console.log("invalid");

}
```

// TASK 16: Policy Type Comparison Error

// Buggy Code:

```
// let policyType = "Health";

// if (policyType == "health") {

// console.log("Health Policy");

// }
```

Correct Code:

```
let policyType = "Health";

if (policyType === "health") {

    console.log("Health Policy");

}
```

// TASK 17: Policy List Rendering Bug

// Buggy Code:

```
// let policies = ["Life", "Health", "Vehicle"];
// policies.forEach(function (policy) {
//   document.getElementById("list").innerHTML += policy;
// });
```

Correct Code:

```
let policies = ["Life", "Health", "Vehicle"];
policies.forEach(function (policy) {
  document.getElementById("list").innerHTML += `<li>${policy}</li>`;
});
```

// TASK 18: Premium Amount Validation

// Buggy Code:

```
// let premium = "5000";
// if (isNaN(premium)) {
//   console.log("Invalid premium");
// } else {
//   console.log("Valid premium");
// }
```

Correct Code:

```
let premium = "324";
if (Number.isNaN(Number(premium)) || premium === null || premium.length === "") {
  console.log(premium, "Invalid premium");
} else {
  console.log(premium, "Valid premium");
}
```