

## A quoi ça représente ?

## Visible par qui ?

Partie présentation du projet	<div>accueil.css</div> <div>accueil.html</div> <div>basededonnees.php</div>	Visible par tout le monde
Visualisation des données en temps réel	<div>ecran-contrôle.css</div> <div>ecran-contrôle.html</div> <div>ecran-contrôle.js</div>	
Gestion des utilisateurs par l'administrateur	<div>gest-user.css</div> <div>gest-user.html</div> <div>gest-user.js</div>	Visible uniquement par l'administrateur
Graphique Chart.js pour crée l'historique de valeurs	<div>historique.css</div> <div>historique.html</div> <div>historique.js</div>	Visible par tout le monde
Partie connexion	<div>login-admin.css</div> <div>login-admin.html</div> <div>login.css</div> <div>login.html</div> <div>main.css</div>	Visible par tout le monde
Style de chaque page du site web		

## Tâches effectuée chaque séances

## Tâble des matières

```
1  Lundi 25 mars (4h) :
2  - Maquette du site web
3  - Connexion au compte administrateur
4
5  Mardi 26 mars (4h) :
6  - Modification de la maquette
7  - Header + Nav + Footer
8
9  Vendredi 29 mars (3h) :
10 - Maquette définitive
11 - Passage du site web en sombre
12 - Partie présentation (en cours)
13
14 Mardi 2 avril (4h) :
15 - Partie présentation du site web terminée
16 - Graphique avec chart.js
17 - Lien entre mes pages web (en cours)
18
19 Mercredi 3 avril (3h)
20 - Partie écran de contrôle du site web terminée (chart.js)
21 - Amélioration de la barre de navigation
22
23 Vendredi 5 avril (4h)
24 - Je me suis documenté d'avantage sur des modèles de boites pour la structure de la page
25 - Prise de note sur les site web des autres groupes
26 - Modification des boutons de navigation
27
28 Lundi 8 avril (4h)
29 - Remise de projet :0)
30 - Refont totale du diaporama qui accompagnera la revue de projet du mardi 16/04
31 - Modification de partie connexion user/admin
32
33 Vendredi 12 avril (4h)
34 - Création de la petite connexion admin/footer à connexion admin simple
35 - Ajout d'un historique de valeur
36
37 Lundi 15 avril (4h)
38 - Gestion de comptes user/admin
39 - Mise à jour de la présentation orale du lendemain
40
41 Mardi 16 avril (4h)
42 - Identification + Commentaire des scripts
43 - Entraînement au parsing à l'oral
44 - Vérification de toutes les fonctionnalités du site web
45
46
47 si possible, design / couleur / responsive à améliorer
48 en attente de la base de données pour rendre les liens de pages fonctionnels
49
50
```

```
connexion user ou admin

si connexion user =
    interface web user :presentation
    valeur en temps réel
    historique valeur
    connexion admin

si connexion admin =
    interface web admin :presentation
    valeur en temps réel
    historique valeur
    gestion utilisateurs
    deconnexion
```

## Script Javascript qui permet de visualisé les données en temps réel

```
ecran-contrôle.js X
admin > ecran-contrôle.js > ...
1 // Fonction pour générer une valeur aléatoire entre min (inclus) et max (inclus)
2 function generateRandomValue(min, max) {
3     return Math.floor(Math.random() * (max - min + 1)) + min;
4 }
5
6 // Fonction pour mettre à jour les valeurs de température et d'humidité
7 function updateValues() {
8     // Définition des valeurs minimales et maximales pour la température et l'humidité
9     const minTemperature = 20;
10    const maxTemperature = 60;
11    const minHumidity = 40;
12    const maxHumidity = 100;
13
14    // Génération de valeurs aléatoires pour la température et l'humidité
15    const temperature = generateRandomValue(minTemperature, maxTemperature);
16    const humidity = generateRandomValue(minHumidity, maxHumidity);
17
18    // Détermination de la couleur de la température en fonction de sa valeur
19    let temperatureColor;
20    if (temperature >= 20 && temperature < 35) {
21        temperatureColor = "green"; // Vert pour une température modérée
22    } else if (temperature >= 35 && temperature < 50) {
23        temperatureColor = "orange"; // Orange pour une température élevée
24    } else {
25        temperatureColor = "red"; // Rouge pour une température critique
26    }
27
28    // Détermination de la couleur de l'humidité en fonction de sa valeur
29    let humidityColor;
30    if (humidity >= 40 && humidity < 65) {
31        humidityColor = "green"; // Vert pour une humidité normale
32    } else if (humidity >= 65 && humidity < 90) {
33        humidityColor = "orange"; // Orange pour une humidité élevée
34    } else {
35        humidityColor = "red"; // Rouge pour une humidité critique
36    }
37
38    // Mise à jour des éléments HTML affichant la température et l'humidité
39    document.getElementById("temperature").innerText = temperature + "°C";
40    document.getElementById("temperature").style.color = temperatureColor; // Changement de la couleur de la température
41
42    document.getElementById("humidity").innerText = humidity + "%";
43    document.getElementById("humidity").style.color = humidityColor; // Changement de la couleur de l'humidité
44
45
46 // Appel à la fonction updateValues toutes les 1,5 seconde (1500 millisecondes)
47 setInterval(updateValues, 1500);
48
```

## Script Javascript qui permet de ajouter et supprimer des utilisateurs, avec la possibilité de leur donné l'accès au compte administrateur

```
gest-user.js X
admin > gest-user.js > @addEventListener("change") callback
1 // Lorsque le formulaire d'inscription est soumis...
2 document.getElementById("signup-form").addEventListener("submit", function(event) {
3     event.preventDefault(); // Empêche le comportement par défaut du formulaire (rechargement de la page)
4
5     // Récupère les valeurs des champs du formulaire
6     var username = document.getElementById("username").value;
7     var password = document.getElementById("password").value;
8     var isAdmin = document.getElementById("admincheckbox").checked; // Vérifie si la case "admin" est cochée
9     var adminEmail = document.getElementById("adminEmail").value;
10
11     // Enregistre les données de l'utilisateur localement ou les envoie à un serveur
12     var userData = {
13         username: username,
14         password: password,
15         isAdmin: isAdmin,
16         adminEmail: isAdmin ? adminEmail : "" // Si l'utilisateur est admin, enregistre l'e-mail, sinon une chaîne vide
17     };
18
19     console.log("Données de l'utilisateur:", userData); // Affiche les données de l'utilisateur dans la console
20
21     // Affiche les informations de l'utilisateur dans la section appropriée de la page
22     document.getElementById("displayUsername").textContent = userData.username;
23     document.getElementById("displayPassword").textContent = userData.password;
24     document.getElementById("displayAdmin").textContent = userData.isAdmin ? "Oui" : "Non";
25     // Affiche "Oui" si l'utilisateur est admin, sinon "Non"
26     document.getElementById("displayEmail").textContent = userData.adminEmail;
27
28     // Efface les champs du formulaire après soumission
29     document.getElementById("username").value = "";
30     document.getElementById("password").value = "";
31     document.getElementById("admincheckbox").checked = false;
32     document.getElementById("adminEmail").value = "";
33
34     // Affiche la section d'informations de l'utilisateur
35     document.getElementById("user-info").style.display = "block";
36
37     // Affiche le bouton pour retirer l'accès
38     document.getElementById("removeAccessBtn").style.display = "block";
39
40     // Ajuste les largeurs des éléments après soumission
41     document.querySelector(".input-container").style.width = "60%";
42     document.querySelector(".user-info-card").style.width = "40%";
43 });
44
45 // Lorsque la case "admin" est cochée ou décochée ces actions sont exécutées
46 document.getElementById("admincheckbox").addEventListener("change", function() {
47     var adminEmailInput = document.getElementById("adminEmail");
48     if (this.checked) {
49         adminEmailInput.style.display = "block"; // Affiche le champ d'e-mail admin si la case est cochée
50     } else {
51         adminEmailInput.style.display = "none"; // Sinon, cache le champ d'e-mail admin
52     }
53 });
54
55 // Lorsque le bouton de suppression d'accès est cliqué ces actions sont exécutées
56 document.getElementById("removeAccessBtn").addEventListener("click", function() {
57     document.getElementById("signup-form").reset();
58     document.getElementById("user-info").style.display = "none";
59     document.getElementById("removeAccessBtn").style.display = "none";
60
61     // Ajuste les largeurs des éléments après la suppression de l'accès
62     document.querySelector(".input-container").style.width = "60%";
63     document.querySelector(".user-info-card").style.width = "40%";
64 });
65
```

## Script Javascript qui permet de visualisé les 10 dernieres valeurs enregistrés à travers le capteur DHT22

```
historique.js X
admin > historique.js > onload
1 // Fonction pour générer une valeur aléatoire
2 function getRandomValue(min, max) {
3     return Math.random() * (max - min) + min;
4 }
5
6 // Configuration du graphique de température
7 var temperatureConfig = {
8     type: 'line',
9     data: {
10         labels: [],
11         datasets: [{
12             label: 'Température (°C)',
13             backgroundColor: 'rgb(255, 99, 132)',
14             borderColor: 'rgb(255, 99, 132)',
15             data: [],
16             fill: false
17         }]
18     },
19     options: {
20         responsive: true,
21         title: {
22             display: true,
23             text: 'Graphique de Température'
24         },
25         scales: {
26             xAxes: [{
27                 display: true
28             }],
29             yAxes: [{
30                 display: true,
31                 scaleLabel: {
32                     display: true,
33                     labelString: 'Température (°C)'
34                 }
35             }]
36         }
37     }
38 };
39
40 // Configuration du graphique d'humidité
41 var humidityConfig = {
42     type: 'line',
43     data: {
44         labels: [],
45         datasets: [{
46             label: 'Humidité (%)',
47             backgroundColor: 'rgb(54, 162, 235)',
48             borderColor: 'rgb(54, 162, 235)',
49             data: [],
50             fill: false
51         }]
52     },
53     options: {
54         responsive: true,
55         title: {
56             display: true,
57             text: 'Graphique d\'Humidité'
58         },
59         scales: {
60             xAxes: [{
61                 display: true
62             }],
63             yAxes: [{
64                 display: true,
65                 scaleLabel: {
66                     display: true,
67                     labelString: 'Humidité (%)'
68                 }
69             }]
70         }
71     }
72 };
73
74 // Création des graphiques et ajout des animations
75 window.onload = function() {
76     var temperatureCtx = document.getElementById("temperatureChart").getContext('2d');
77     var humidityCtx = document.getElementById("humidityChart").getContext('2d');
78     window.temperatureChart = new Chart(temperatureCtx, temperatureConfig);
79     window.humidityChart = new Chart(humidityCtx, humidityConfig);
80
81     // Boucle de mise à jour
82     setInterval(function() {
83         var timestamp = new Date().toLocaleTimeString(); // Heure actuelle
84         var temperatureValue = getRandomValue(20, 60); // Température aléatoire entre 20°C et 60°C
85         var humidityValue = getRandomValue(40, 100); // Humidité aléatoire entre 40% et 100%
86
87         // Ajouter les nouvelles valeurs aux graphiques
88         addData(window.temperatureChart, timestamp, temperatureValue);
89         addData(window.humidityChart, timestamp, humidityValue);
90     }, 1000); // 1000 millisecondes = 1 seconde
91
92     // Fonction pour ajouter une nouvelle valeur au graphique en limitant à 10 valeurs
93     function addData(chart, label, value) {
94         if (chart.data.labels.length >= 10) {
95             chart.data.labels.shift(); // Supprimer la première valeur du tableau des labels
96             chart.data.datasets.forEach((dataset) => {
97                 dataset.data.shift(); // Supprimer la première valeur du tableau des données
98             });
99         }
100         chart.data.labels.push(label);
101         chart.data.datasets.forEach((dataset) => {
102             dataset.data.push(value);
103         });
104         chart.update();
105     }
106 }
```