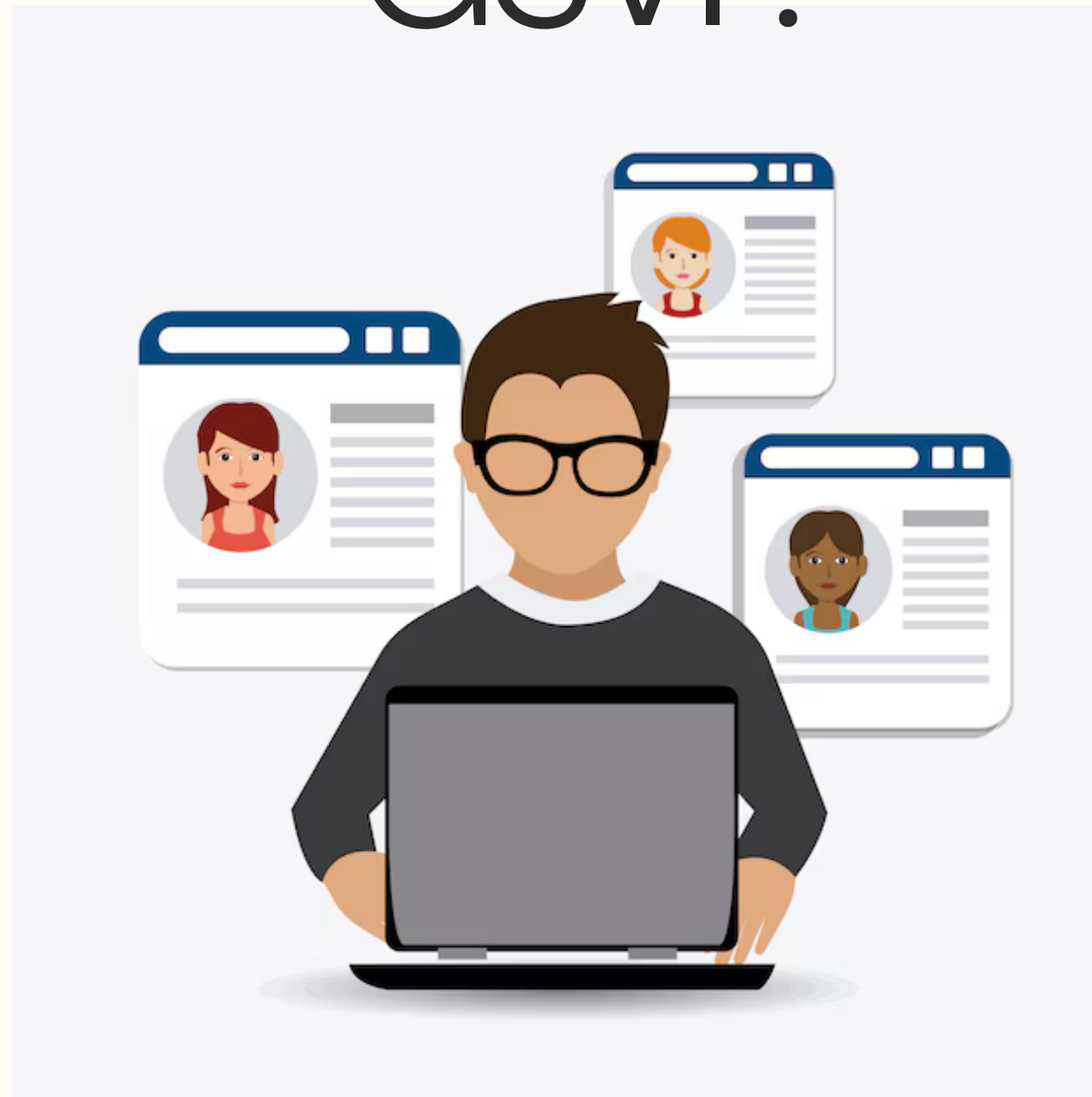


Les sites WEB réalisées pour GSVI :



Générateur de mots de passe

Objectif : Générer des mots de passe sans caractères qui portent à confusions (O et 0 I et l...)

Demande : Respect de la palette de couleur de l'entreprise en gardant toujours la même direction artistique

mdp.gsvi.fr

Installation d'un host TeamViewer

Objectif : Guider au mieux possible les utilisateurs dans l'installation d'un host TeamViewer personnalisée par l'entreprise.

Demande : Respect de la palette de couleur de l'entreprise en gardant toujours la même direction artistique et l'ajout d'un tutoriel d'installation

si.gsvi.fr

Site de surveillance de site WEB

Objectif : Avoir une page qui indique si un site web est en panne ou en fonction

Demande : Respect de la palette de couleur de l'entreprise en gardant toujours la même direction artistique

Site non publié



Générateur de Mot de Passe

KH3#Uc2h

Générer un mot de passe

[Tickets GSVI GLPI](#)
[Teamviewer Host](#)

Ajouter un Serveur +

🕒 Dernière mise à jour : 20:18:32

Statut des Serveurs

Serveurs Permanents

Global Documentation -	UP (Port 80)
Global Expedition + Bluebox -	UP (Port 80)
Global Retours -	UP (Port 80)
Economie Global Distrib -	DOWN
Paccarnet Sharepoint -	UP (Port 443)
GLPI -	UP (Port 443)
MCSUD -	UP (Port 443)
CGDPL -	UP (Port 443)
SVUL -	UP (Port 443)
SERVILOC -	UP (Port 443)
GSVI -	UP (Port 443)
SI GSVI -	UP (Port 80)
MDP GSVI -	UP (Port 80)

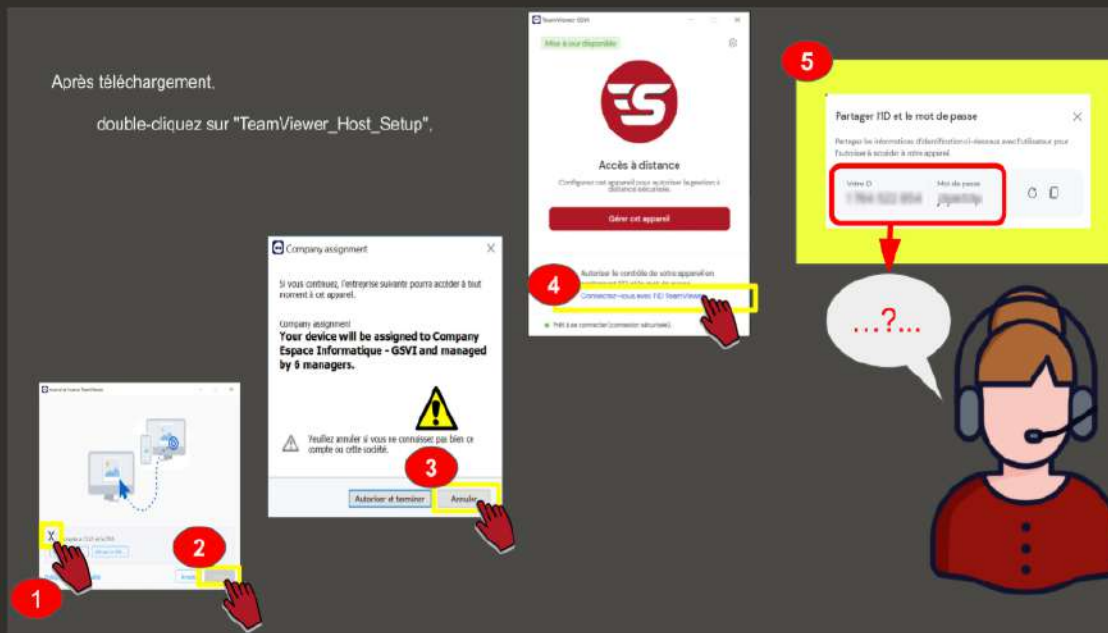
Serveurs Ajoutés



Cliquez sur le bouton ci-dessous pour télécharger TeamViewer :

Télécharger TeamViewer

Après téléchargement,
double-cliquez sur "TeamViewer_Host_Setup".



PING.HTML

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Statut des Serveurs</title>
  <style>
    body {
      background-color: #32312c;
      color: white;
      font-family: Arial, sans-serif;
      text-align: center;
      padding: 20px;
      position: relative;
    }
    .container {
      max-width: 1200px;
      margin: auto;
      display: flex;
      justify-content: space-between;
      gap: 20px;
    }
    .column {
      width: 48%; /* 2 columns side by side */
    }
    .status-box {
      padding: 10px;
      margin: 10px;
      border-radius: 5px;
      font-weight: bold;
      position: relative;
      display: flex;
      justify-content: space-between;
      text-align: left;
    }
    .up {
      background-color: green;
      color: white;
    }
    .down {
      background-color: #af1926;
      color: white;
    }
    .timestamp {
      position: absolute;
      top: 10px;
      right: 10px;
```

```
        background-color: rgba(0, 0, 0, 0.5);
        padding: 10px;
        border-radius: 5px;
        font-size: 0.9em;
    }
    .add-btn {
        position: absolute;
        top: 20px;
        left: 20px;
        background-color: rgba(0, 0, 0, 0.5);
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }
    .add-btn:hover {
        background-color: rgba(0, 0, 0, 0.7);
    }
    .add-btn span {
        margin-left: 5px;
    }
    .popup {
        display: none;
        position: fixed;
        top: 50%;
        left: 50%;
        transform: translate(-50%, -50%);
        background-color: rgba(0, 0, 0, 0.8);
        color: white;
        padding: 20px;
        border-radius: 10px;
        z-index: 1000;
        width: 300px;
    }
    .popup input {
        width: 100%;
        padding: 8px;
        margin: 10px 0;
        border-radius: 5px;
        border: 1px solid #ccc;
    }
    .popup button {
        padding: 10px 20px;
        background-color: #4CAF50;
        color: white;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }
    .popup button:hover {
```

```

        background-color: #45a049;
    }
    .popup .close-btn {
        position: absolute;
        top: 5px;
        right: 5px;
        background-color: #af1926;
        color: white;
        border: none;
        padding: 5px;
        border-radius: 50%;
        cursor: pointer;
        font-size: 1.2em;
    }
    .popup .close-btn:hover {
        background-color: #9a0e1f;
    }
    .overlay {
        display: none;
        position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
        background-color: rgba(0, 0, 0, 0.5);
        z-index: 999;
    }
</style>
</head>
<body>
    <h1>Statut des Serveurs</h1>
    <div class="container">
        <!-- Serveurs permanents (existants dès le début) -->
        <div class="column" id="permanent-servers">
            <h2>Serveurs Permanents</h2>
        </div>

        <!-- Serveurs ajoutés via le formulaire -->
        <div class="column" id="added-servers">
            <h2>Serveurs Ajoutés</h2>
        </div>
    </div>

    <!-- Add Server Button with + emoji -->
    <button class="add-btn" onclick="openPopup()">Ajouter un Serveur
    <span>✚</span></button>

    <!-- Overlay for Popup -->
    <div class="overlay" id="overlay" onclick="closePopup()"></div>

    <!-- Popup for adding server -->

```

```

<div class="popup" id="popup">
  <button class="close-btn" onclick="closePopup()">✖</button>
  <h2>Ajouter un Serveur</h2>
  <input type="text" id="new-name" placeholder="Entrez le nom du
serveur" required>
  <input type="text" id="new-ip" placeholder="Entrez l'adresse IP
du serveur" required>
  <button onclick="addServer()">Ajouter</button>
</div>

```

```

<div id="timestamp" class="timestamp">🕒 Dernière mise à jour :
--:--:--</div>

```

```

<script>
  let servers = [
    { ip: "145.239.37.162", name: "Global Documentation",
isNew: false },
    { ip: "194.60.240.176", name: "Global Expedition +
Bluebox", isNew: false },
    { ip: "194.60.240.175", name: "Global Retours", isNew:
false },
    { ip: "economie.global-distrib.com", name: "Economie Global
Distrib", isNew: false }, // Serveur permanent
    { ip: "paccarnet.sharepoint.com", name: "Paccarnet
Sharepoint", isNew: false }, // Serveur permanent
    { ip: "sigsvi.with22.glpi-network.cloud/", name: "GLPI",
isNew: false }, // Serveur permanent
    { ip: "mcsud.fr/:443", name: "MCSUD", isNew: false }, //
Serveur permanent
    { ip: "vos-pieces-camion.fr/cgdpl/", name: "CGDPL", isNew:
false }, // Serveur permanent
    { ip: "svul.pro", name: "SVUL", isNew: false }, // Serveur
permanent
    { ip: "servi-loc.com", name: "SERVILOC", isNew: false }, //
Serveur permanent
    { ip: "gsvi.com", name: "GSVI", isNew: false }, // Serveur
permanent
    { ip: "si.gsvi.fr", name: "SI GSVI", isNew: false }, //
Serveur permanent
    { ip: "mdp.gsvi.fr", name: "MDP GSVI", isNew: false } //
Serveur permanent
  ];

```

```

  async function checkStatus(ip, elementId, serverName) {
    const statusBox = document.getElementById(elementId);
    let status = `<span class='down'>DOWN</span>`;
    statusBox.innerHTML = `${serverName} - ${status}`;
    statusBox.classList.remove('up');
    statusBox.classList.add('down');
  }

```

```

        const ports = [443, 22, 21, 80, 8080, 3306]; // Commencer
par tester le port 443 pour HTTPS

        for (let port of ports) {
            try {
                const protocol = (port === 443) ? 'https://' :
'http://';
                await fetch(`${protocol}${ip}:${port}`, { method:
'HEAD', mode: 'no-cors' });
                status = `<span class='up'>UP (Port
${port})</span>`;
                statusBox.innerHTML = `${serverName} - ${status}`;
                statusBox.classList.remove('down');
                statusBox.classList.add('up');
                break;
            } catch (error) {
                // Ignore les erreurs, passer au prochain port
            }
        }
    }

    function updateStatus() {
        const permanentContainer =
document.getElementById('permanent-servers');
        const addedContainer =
document.getElementById('added-servers');
        permanentContainer.innerHTML = '<h2>Serveurs
Permanents</h2>';
        addedContainer.innerHTML = '<h2>Serveurs Ajoutés</h2>';

        servers.forEach((server, index) => {
            const serverDiv = document.createElement('div');
            serverDiv.id = `server${index + 1}`;
            serverDiv.classList.add('status-box', 'down');
            serverDiv.innerHTML = `${server.name} - <span
class='down'>DOWN</span>`; // Affichage du nom du serveur

            // Add remove button for each server
            const removeButton = document.createElement('button');
            removeButton.classList.add('remove-btn');
            removeButton.innerHTML = '✖';
            removeButton.onclick = function() {
                removeServer(index);
            };
            serverDiv.appendChild(removeButton);

            if (server.isNew) {
                addedContainer.appendChild(serverDiv);
            } else {
                permanentContainer.appendChild(serverDiv);
            }
        });
    }

```

```

        checkStatus(server.ip, `server${index + 1}`,
server.name);
    });

    const now = new Date().toLocaleTimeString();
    document.getElementById('timestamp').innerText = `
Dernière mise à jour : ${now}`;
}

function removeServer(index) {
    servers.splice(index, 1);
    updateStatus();
}

function addServer() {
    const newName =
document.getElementById('new-name').value.trim();
    const newIP =
document.getElementById('new-ip').value.trim();

    if (newName && newIP) {
        const newServer = { ip: newIP, name: newName, isNew:
true };

        servers.push(newServer);
        updateStatus();
        closePopup();
    }
}

function openPopup() {
    document.getElementById('popup').style.display = 'block';
    document.getElementById('overlay').style.display = 'block';
}

function closePopup() {
    document.getElementById('popup').style.display = 'none';
    document.getElementById('overlay').style.display = 'none';
}

updateStatus();
setInterval(updateStatus, 10000);
</script>
</body>
</html>

```

MDP.HTML

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Générateur de Mot de Passe</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #32312c;
      color: white;
      text-align: center;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }
    .container {
      background: #444;
      padding: 20px;
      border-radius: 10px;
      display: inline-block;
    }
    #password {
      font-size: 20px;
      font-weight: bold;
      background: white;
      color: #000;
      padding: 10px;
      border-radius: 5px;
      display: inline-block;
      min-width: 200px;
    }
    .btn {
      margin-top: 10px;
      background-color: #af1926;
      color: white;
      border: none;
      padding: 10px 20px;
      font-size: 16px;
      cursor: pointer;
      border-radius: 5px;
    }
```

```

    }
    .btn:hover {
        background-color: #920f1e;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Générateur de Mot de Passe</h1>
        <div id="password">Cliquez sur Générer</div>
        <br>
        <button class="btn" onclick="generatePassword()">Générer un mot
de passe</button>
        <button class="btn" onclick="copyPassword()">Copier</button>
    </div>

    <script>
        function generatePassword() {
            const length = 12;
            const uppercase = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
            const lowercase = 'abcdefghijklmnopqrstuvwxyz';
            const numbers = '0123456789';
            const special = '!@#$%^&*()_+ ';

            let allChars = uppercase + lowercase + numbers + special;
            let password = '';

            password += uppercase[Math.floor(Math.random() *
uppercase.length)];
            password += numbers[Math.floor(Math.random() *
numbers.length)];
            password += special[Math.floor(Math.random() *
special.length)];

            for (let i = 3; i < length; i++) {
                password += allChars[Math.floor(Math.random() *
allChars.length)];
            }

            password = password.split('').sort(() => Math.random() -
0.5).join('');
            document.getElementById('password').textContent = password;
        }

        function copyPassword() {
            const passwordText =
document.getElementById('password').textContent;
            navigator.clipboard.writeText(passwordText).then(() => {
                alert('Mot de passe copié !');
            });
        }
    </script>

```

```
</script>  
</body>  
</html>
```

TeamViewer.html

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Téléchargez TeamViewer GSVI</title>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
      display: flex;
      flex-direction: row;
      height: 100vh;
      color: white;
    }

    .left-section {
      background-color: #af1926;
      width: 30%;
      padding: 20px;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      text-align: center;
    }

    .left-section img {
      max-width: 80%;
      height: auto;
      margin-bottom: 20px;
    }

    .left-section p {
      margin-bottom: 20px;
    }

    .left-section .button-container {
      margin-top: 20px;
    }

    .left-section button {
      background-color: white;
      color: #af1926;
      border: none;
```

```

        padding: 10px 20px;
        font-size: 16px;
        cursor: pointer;
        border-radius: 5px;
    }

    .left-section button:hover {
        background-color: #f2f2f2;
    }

    .right-section {
        background-color: #32312c;
        width: 70%;
        padding: 20px;
        overflow-y: auto;
    }

    .right-section h2 {
        margin-top: 0;
    }

    .tutorial-image {
        width: 960px;
        height: 540px;
        margin: 85px auto;
        display: block;
    }
}
</style>
</head>
<body>
    <div class="left-section">
        
        <p>Cliquez sur le bouton ci-dessous pour télécharger TeamViewer
:</p>
        <div class="button-container">
            <button
onclick="window.location.href='https://custom.teamviewer.com/63ytbby'">
Télécharger TeamViewer</button>
        </div>
    </div>
    <div class="right-section">
        <p>Après téléchargement, double-cliquez sur
"TeamViewer_Host_Setup" puis... </p>
        
    </div>
</body>
</html>

```