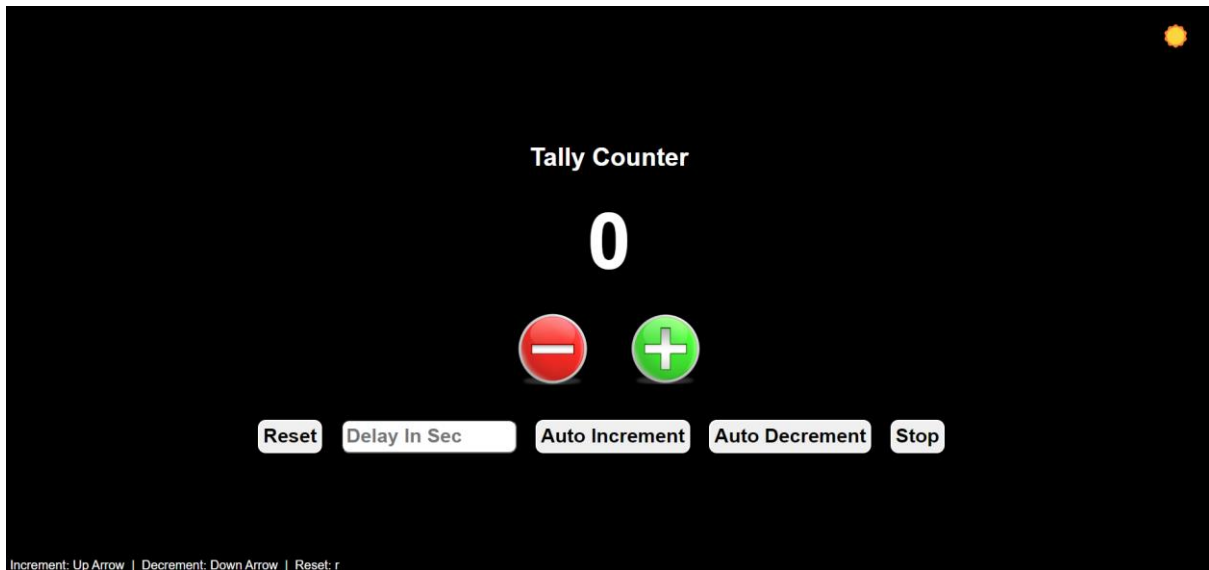


TALLY COUNTER

SNIPPET



SOURCE CODE

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tally Counter</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <button id="dark-mode-toggle"> ☀ </button>
  <h1>Tally Counter</h1>
  <div id="work-area">
    <p id="counter"></p>
    <div id="btn">
      <button id="inc"></button>
      <button id="dec"></button>
    </div>
    <button id="reset" onclick="reset()">Reset</button>
    <input type="number" id="del" placeholder="Delay In Sec" onclick="stop()">
    <button id="autoinc" onclick="autoInc()">Auto Increment</button>
    <button id="autodec" onclick="autoDec()">Auto Decrement</button>
    <button id="stop" onclick="stop()">Stop</button>
  </div>
</div>
```

```
<p id="instruction">Increment: Up Arrow    |        Decrement: Down
Arrow        |        Reset: r</p>
</div>
<script src="script.js"></script>
</body>
</html>
```

CSS:

```
body {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  font-family: Arial, sans-serif;
}

#work-area {
  text-align: center;
}

#counter {
  font-size: 100px;
  font-weight: bold;
  border: none;
  margin-bottom: 10px;
  margin-top: 10px;
}

input[type="text"] {
  font-size: 5rem;
  width: 100px;
  text-align: center;
}

#inc, #dec {
  width: 100px;
  height: 100px;
  margin: 20px;
  border: none;
  background-color: transparent;
  border-radius: 50px;
}

button img {
  width: 100%;
  height: 100%;
  object-fit: contain;
  padding: 0;
}
```

```
button:hover {  
    cursor: pointer;  
}
```

```
button:active {  
    transform: scale(0.95);  
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);  
}
```

```
#reset, #autoinc, #autodec, #stop {  
    margin-top: 20px;  
    margin-right: 20px;  
    font-size: 25px;  
    font-weight: bold;  
    border-radius: 10px;  
    padding: 5px;  
}
```

```
#del {  
    width: 210px;  
    margin-top: 20px;  
    margin-right: 20px;  
    font-size: 25px;  
    font-weight: bold;  
    border-radius: 10px;  
    padding: 5px;  
}
```

```
#instruction {  
    position: absolute;  
    bottom: 5px;  
    left: 5px;  
    margin-bottom: 0;  
}
```

```
#dark-mode-toggle {  
    position: absolute;  
    top: 20px;  
    right: 20px;  
    background-color: transparent;  
    border: none;  
    font-size: 30px;  
}
```

```
#dark-mode-toggle:active {  
    box-shadow: none;  
}
```

JavaScript:

```
var count
```

```
if(localStorage.getItem('count') != null){
    count = localStorage.getItem('count');
}
else{
    count = 0;
}
document.getElementById('counter').innerHTML = count;

function inc(){
    count++;
    document.getElementById('counter').innerHTML = count;
    localStorage.setItem('count', count);
}

function dec(){
    if(count == 0){
        return
    }
    count--;
    document.getElementById('counter').innerHTML = count;
    localStorage.setItem('count', count);
}

function reset(){
    count = 0;
    document.getElementById('counter').innerHTML = count;
    localStorage.setItem('count', count);
}
```

```
var intervalId
var del
function autoInc (){
    stop()
    del = parseInt(document.getElementById('del').value)*1000;
    intervalId = setInterval(function(){
        inc();
    }, del);
}
```

```
function autoDec (){
    stop()
    del = parseInt(document.getElementById('del').value)*1000;
    intervalId = setInterval(function(){
        dec();
    }, del);
}
```

```
    if(count == 0){
        clearInterval(intervalId);
    }
}, del);
}
```

```
function stop(){
    clearInterval(intervalId);
}
```

```
const darkModeToggle = document.getElementById('dark-mode-toggle');
var buttons = document.querySelectorAll('#reset, #autoinc, #autodec, #stop');
```

```
if (localStorage.getItem('darkMode') === 'enabled') {
    enableDarkMode();
} else {
    disableDarkMode();
}
```

```
darkModeToggle.addEventListener('click', () => {
    if (localStorage.getItem('darkMode') !== 'enabled') {
        enableDarkMode();
    } else {
        disableDarkMode();
    }
});
```

```
function enableDarkMode() {
    darkModeToggle.innerHTML = '☀️';
    document.body.style.backgroundColor = 'black';
    document.body.style.color = 'white';
    buttons.forEach(button => {
        button.style.border = '2px solid white';
    });
    localStorage.setItem('darkMode', 'enabled');
}
```

```
function disableDarkMode() {
    darkModeToggle.innerHTML = '🌑';
    document.body.style.backgroundColor = 'white';
    buttons.forEach(button => {
        button.style.border = '2px solid black';
    });
    document.body.style.color = 'black';
    localStorage.setItem('darkMode', 'disabled');
}
```

```
document.addEventListener('keydown', function(event) {
```

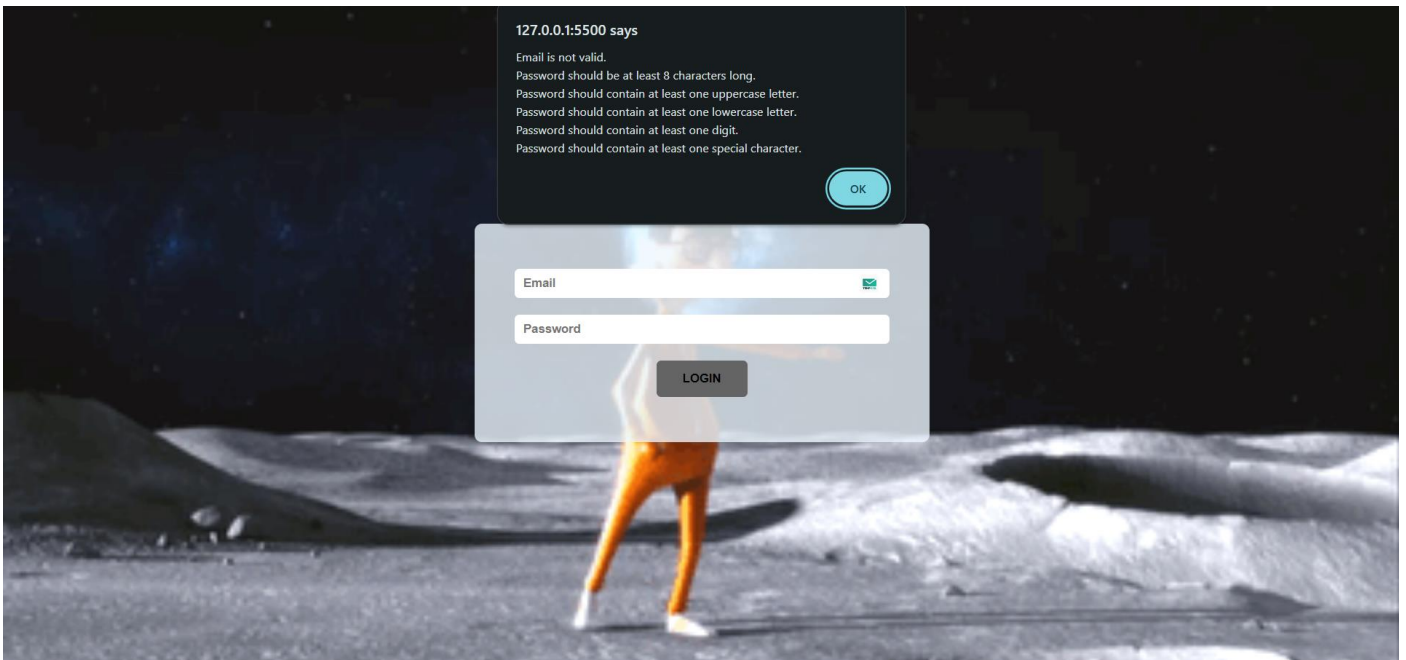
```
if(event.key == 'ArrowUp'){
    inc();
}
if(event.key == 'ArrowDown'){
    dec();
}
if(event.key == 'r' || event.key == 'R'){
    reset();
}
});

document.getElementById('del').addEventListener('keydown', function(event) {
    if(event.key == 'Enter'){
        autoInc();
    }
});
```

Github: <https://github.com/DMegatron/Tally-Counter>

FORM VALIDATION

SNIPPET



SOURCE CODE

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <center>
    <form action="https://google.com" class="form" onsubmit="return validate()">
      <input class="email" type="text" placeholder="Email">
      <br>
      <input class="password" type="password" placeholder="Password">
      <br>
      <input class="btn" type="submit" value="LOGIN">
    </form>
  </center>
  <script src="script.js"></script>
</body>
</html>
```

CSS:

```
body {  
  background-image: url(images/vector\ gif.gif);  
  background-size: cover;  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: center;  
  height: 97vh;  
}
```

```
.form {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: center;  
  height: 240px;  
  width: 500px;  
  background-color: rgba(237, 248, 255, 0.8);  
  border-radius: 8px;  
  border: 0;  
}
```

```
.email, .password {  
  height: 30px;  
  width: 80%;  
  background-color: rgb(255, 255, 255);  
  border: 0;  
  border-radius: 5px;  
  padding-left: 10px;  
  font-weight: 600;  
}
```

```
.email:focus, .password:focus {  
  border: 1px solid rgb(0, 0, 0);  
}
```

```
.btn {  
  height: 40px;  
  width: 20%;  
  border-radius: 5px;  
  border: 0;  
  font-weight: 600;  
  color: rgb(70, 70, 70);  
  background-color: rgb(179, 179, 179);  
}
```



```
.btn:hover {  
  background-color: rgb(100, 100, 100);  
  color: black;  
  cursor: pointer;  
}
```

JavaScript:

```
var email;  
var password;  
var errorMessages = [];  
  
function validate(event) {  
  email = document.querySelector('.email');  
  password = document.querySelector('.password');  
  errorMessages = [];  
  
  // removeErrorMessages(email);  
  // removeErrorMessages(password);  
  
  if (!(email.value.includes('@') && email.value.includes('.'))) {  
    errorMessages.push("Email is not valid.");  
    email.focus();  
  }  
  
  if (password.value.length < 8) {  
    errorMessages.push("Password should be at least 8 characters long.");  
  }  
  
  if (!/[A-Z]/.test(password.value)) {  
    errorMessages.push("Password should contain at least one uppercase letter.");  
  }  
  
  if (!/[a-z]/.test(password.value)) {  
    errorMessages.push("Password should contain at least one lowercase letter.");  
  }  
  
  if (!/[0-9]/.test(password.value)) {  
    errorMessages.push("Password should contain at least one digit.");  
  }  
  
  if (!/^[a-zA-Z0-9]/.test(password.value)) {  
    errorMessages.push("Password should contain at least one special character.");  
  }  
  
  if (errorMessages.length > 0) {  
    password.focus();  
    showError(errorMessages.join('\n'));
```

```

        errorMessages = [];
        return false;
    }

    return true;
}

function showError(errorMessage) {
    alert(errorMessage);
    // input.style.borderWidth = "2px";
    // input.style.borderColor = "red";
    // input.style.borderStyle = "solid";

    // var errorMsg = document.createElement('div');
    // errorMsg.innerText = errorMessage;
    // errorMsg.classList.add('error-message');
    // errorMsg.style.color = "red";
    // errorMsg.style.fontFamily = "Arial";
    // errorMsg.style.width = "81%";
    // errorMsg.style.marginTop = "5px";
    // errorMsg.style.textAlign = "left";
    // input.insertAdjacentElement('afterend', errorMsg);
    // input.focus();
}

// function removeErrorMessages(input) {
//     if (input.nextElementSibling && input.nextElementSibling.classList.contains('error-message')) {
//         input.nextElementSibling.remove();
//     }
// }

```

GitHub: <https://github.com/DMegatron/Form-Validation>