

# BlurSession

## Enterprise-grade privacy solution for sensitive information

### Overview

BlurSession is a lightweight, enterprise-ready JavaScript solution that automatically blurs sensitive content after a period of user inactivity. This security enhancement is ideal for organizations that handle sensitive information including financial institutions, healthcare platforms, tax software, and any application displaying Personally Identifiable Information (PII).

BlurSession addresses the critical security gap between active login and automatic timeout, providing immediate visual protection when users step away from their screens. This implementation helps protect both internal proprietary systems and external client data, demonstrating a commitment to privacy and security.

### Features

- **Automatic Privacy Protection:** Blurs sensitive content after configurable period of inactivity
- **Instant Reactivation:** Content becomes visible again with mouse movement or touch
- **Mobile-Optimized:** Fully responsive with touch event support for all devices
- **Zero Dependencies:** Pure vanilla JavaScript with no external libraries
- **Enterprise-Ready:** Production-quality code suitable for high-security environments
- **Configurable:** Easily adjust inactivity time, blur intensity, and other parameters
- **Event Callbacks:** Hook into blur activation/deactivation for custom behaviors
- **Tab Visibility Aware:** Automatically blurs when users switch to other browser tabs
- **Cross-Browser Compatible:** Works on all modern browsers

### Business & Security Value

- **Addresses Real Security Gap:** Protects against unauthorized viewing of sensitive information on unattended screens
- **Regulatory Compliance:** Supports GDPR, HIPAA, PCI-DSS, and other privacy requirements
- **Simple Implementation:** Requires minimal code changes to implement in existing applications
- **Low Resource Footprint:** Negligible impact on application performance
- **Enhanced User Trust:** Visible commitment to protecting user data

### Quick Start

1. Include the required files in your HTML:

html

```
<link rel="stylesheet" href="css/styles.css">
<link rel="stylesheet" href="css/blur-effect.css">
<script src="js/blursession.js"></script>
```

2. Add the necessary HTML structure:

html

```
<!-- Header (will remain visible) -->
<header>
  <!-- Your header content -->
</header>

<!-- Main content (will be blurred) -->
<main id="sensitive-area">
  <!-- Your sensitive content here -->
</main>

<!-- Footer (will remain visible) -->
<footer>
  <!-- Your footer content -->
</footer>

<!-- Blur overlay element (required) -->
<div id="blur-overlay"></div>

<!-- Status banner (optional but recommended) -->
<div id="status-banner">
  <div class="container">
    <span id="status-message">BlurSession active - content is protected</span>
  </div>
</div>
```

3. Initialize BlurSession with your desired configuration:

javascript

```
// Default configuration is automatically applied
// To customize:
window.blurSession = new BlurSession({
  inactivityTime: 10000, // 10 seconds (in milliseconds)
  blurIntensity: 8,      // 8px blur
  debug: true           // Enable console logging
});
```

## Configuration Options

BlurSession can be customized with the following options:

Option	Type	Default	Description
<code>inactivityTime</code>	Number	5000	Time in milliseconds before content is blurred
<code>sensitiveSelector</code>	String	<code>'#sensitive-area'</code>	CSS selector for the area to protect
<code>overlaySelector</code>	String	<code>'#blur-overlay'</code>	CSS selector for the blur overlay element
<code>blurIntensity</code>	Number	10	Blur intensity in pixels
<code>statusMessageSelector</code>	String	<code>'#status-message'</code>	CSS selector for the status message element
<code>showStatusBanner</code>	Boolean	true	Whether to show the status banner
<code>onActivate</code>	Function	null	Callback when blur mode activates
<code>onDeactivate</code>	Function	null	Callback when blur mode deactivates
<code>debug</code>	Boolean	false	Enable debug mode

## Methods

BlurSession instance provides the following methods:

Method	Description
<code>activateBlur()</code>	Manually activate blur mode (blur content)
<code>deactivateBlur()</code>	Manually deactivate blur mode (remove blur)
<code>setInactivityTime(seconds)</code>	Set the inactivity time in seconds
<code>setBlurIntensity(pixels)</code>	Set the blur intensity in pixels
<code>cleanup()</code>	Clean up event listeners and timers

## Browser Support

BlurSession is compatible with all modern browsers:

- Chrome (latest)
- Firefox (latest)
- Safari (latest)
- Edge (latest)
- Opera (latest)

For older browsers that don't support the `backdrop-filter` property, the content will be hidden using an opacity overlay as a fallback.

## Technical Environment Compatibility

- Works with any web server (Apache, Nginx, IIS)
- Compatible with all modern hosting environments
- No special server-side requirements
- Works with PHP, Node.js, ASP.NET and all other web platforms

## Implementation Examples

See the `demo.html` and `index.html` files for implementation examples. For detailed integration instructions, refer to the `IMPLEMENTATION_GUIDE.md`.

## About the Author

Hi, I'm Chad Wigington, a risk, compliance, and governance professional. I hold an MBA from Indiana University's Kelley School of Business and an undergraduate degree in Risk, Insurance, and Safety Management.

I created BlurSession to help make an impact by protecting organizations that work with Personal Identifiable Information and the public whose information could be exposed.

Connect with me on [LinkedIn](#)

## License

This project is available under the MIT License. See the LICENSE.md file for details.

Copyright (c) 2025 Chad Wigington

---

Feel free to customize and improve this solution. Please share your innovations!