



# Greenstem - Official Website

## Greenstem Software Business Sdn Bhd Official Website

- [Introduction](#)
- [Repository](#)
- [Technology Stack](#)
  - [Frontend](#)
    - Bootstrap 5.3
    - JQuery 3.7.1
    - Swiper 11.0.5
    - Animate.css 4.1.1
    - Fancybox 5.0
  - [Backend](#)
    - PHP 7.2
- [Folder Structure](#)
- [Design Guidelines](#)
  - Color Schema
  - Typography
  - Buttons and links
  - Image and Icons
  - Page Layout
  - General Animation
- [Responsiveness](#)
- [Report](#)

- Lighthouse Report
- Structured Data Test
- Detail for each Page
  - Navigation Bar
  - Footer
  - Index/Home Page
  - Product Page
  - About Us Page
  - Tutorial Page
  - Contact Us Page
  - Additional JavaScript Code
- Backup
- Special Thank

## Introduction

This document provides comprehensive guidance on the development of the official website. The website is designed to showcase essential information about Greenstem Software Business Sdn Bhd. The website will include sections such as About Us, Products and Services, Tutorial Video and Contact Us. It will feature a clean and modern design with a user-friendly interface. The goal is to effectively communicate the company's mission, values, and expertise to potential clients and partners.

## Repository

The source code for this project is hosted on our cPanel account. As this is a private hosting environment, you can access the live website at <https://www.greenstem.com.my/>.

For access to the source code, please refer to Ms. Angie or the designated administrator for login credentials and repository details.

# Technology Stack

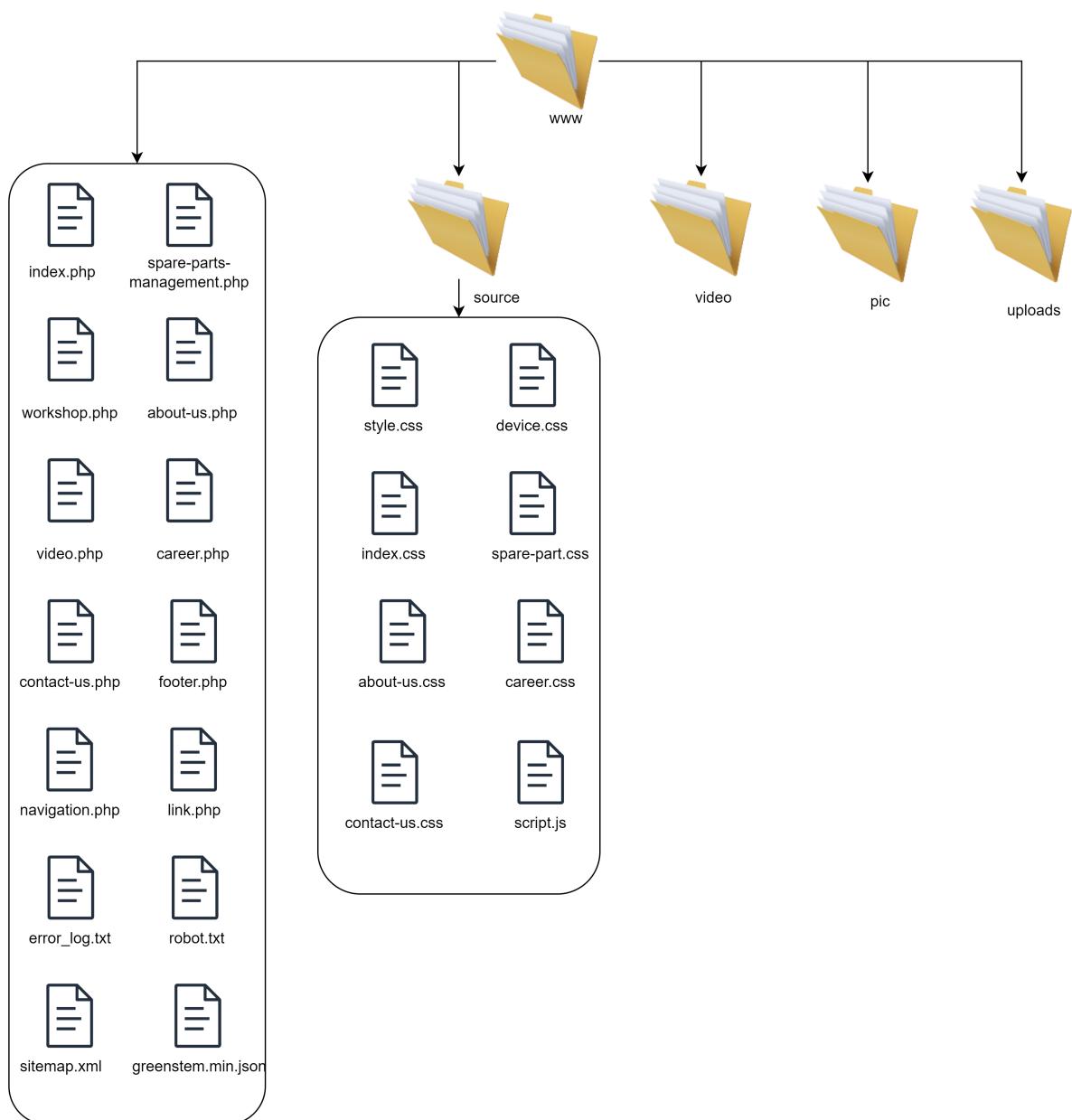
## ▼ Frontend

- Bootstrap 5.3 - a standard **responsive web design framework**, provides custom user interface for different device screen sizes.  
[Official] <https://getbootstrap.com/>
- JQuery 3.7.1 - a classic JavaScript library for effect and delay looping images.  
[Official]  
<https://jquery.com/>
- Swiper 11.0.5 - a modern, mobile-friendly, and highly customizable JavaScript library for creating touch-enabled sliders and carousels.  
[Official] <https://swiperjs.com/>
- Animate.css 4.1.1 - a lightweight and easy-to-use CSS animation library that adds a variety of cross-browser animations to your HTML elements.  
[Official] <https://animate.style/>
- Fancybox 5.0 - a versatile and feature-rich JavaScript library that provides an elegant solution for displaying images, videos, HTML content, and more in a responsive lightbox/modal overlay. [Official]  
<https://fancyapps.com/fancybox/>

## ▼ Backend

- PHP 7.2 - a typical **backend programming language**, can change the version in cPanel.  
[Official]  
<https://www.php.net/>

# Folder Structure



# Design Guidelines

## ▼ Color Schema

- Background : #29A87A (green), #6046FF (purple)
- Text Color : white, black;

## ▼ Typography

- Font Family
  - Primary : 'Istok Web', sans-serif
  - Secondary : 'Nunito', sans-serif

<https://fonts.google.com/>

- Font Size

The font size is set using `vw` (view width) units, allowing it to scale proportionally with the width of the viewport. This approach ensures a responsive layout that adapts well to various screen sizes.

```
h4 {  
    font-size: 1.18vw;  
    font-style: normal;  
    font-weight: 700;  
    line-height: normal;  
}  
  
h5 {  
    color: #29A87A;  
    font-size: 1.125vw;  
    font-weight: 700;  
}  
  
p,  
pre {  
    font-size: 1vw;  
    font-weight: 400;
```

```
h1 {  
    font-size: 2.3vw;  
    font-weight: 700;  
    line-height: normal;  
}  
  
h2 {  
    font-size: 1.825vw;  
    font-weight: 700;  
    line-height: normal;  
}  
  
h3 {  
    font-size: 1.75vw;  
    font-style: normal;  
    font-weight: 600;  
    line-height: normal;
```

```
    font-style: normal;  
    line-height: normal;  
}
```

```
    margin-bottom: 0;  
}
```

## ▼ Buttons and links

- All links (`<a>` elements) feature a consistent and clean styling, maintaining a uniform appearance without additional hover effects.

```
a {  
    text-decoration: none;  
    color: #ffffff;  
    cursor: pointer;  
}
```

- All buttons are equipped with a hover effect triggered by the `onmouseover` event, which calls the `changeButtonStyle` function. This function dynamically alters the color and background color of the button. For example, you can implement the function as demonstrated

```
<button type="submit" onmouseover="changeButtonStyle(this, true)">Send</button>
```

```
function changeButtonStyle(button, onHover) {  
    var color = button.style.color || getComputedStyle(button).color;  
    var bgColor = button.style.backgroundColor || getCom
```

```

putedStyle(button).backgroundColor;
    var border = button.style.border || getComputedStyle
(button).border;

    if (onHover) {
        button.style.color = bgColor;
        button.style.backgroundColor = color;
        button.style.border = `2.5px solid ${bgColor}`;
    }

    button.addEventListener('mouseout', function () {
        button.style.backgroundColor = bgColor;
        button.style.color = color;
        button.style.border = border;
    });
}

```

## ▼ Image and Icons

- Icons

Icons are sourced from the Google Material Icons library.

<https://fonts.google.com/icons>

- Images

All images are styled to maintain a consistent aspect ratio, ensuring that their proportions are preserved and preventing distortion in their appearance.

```

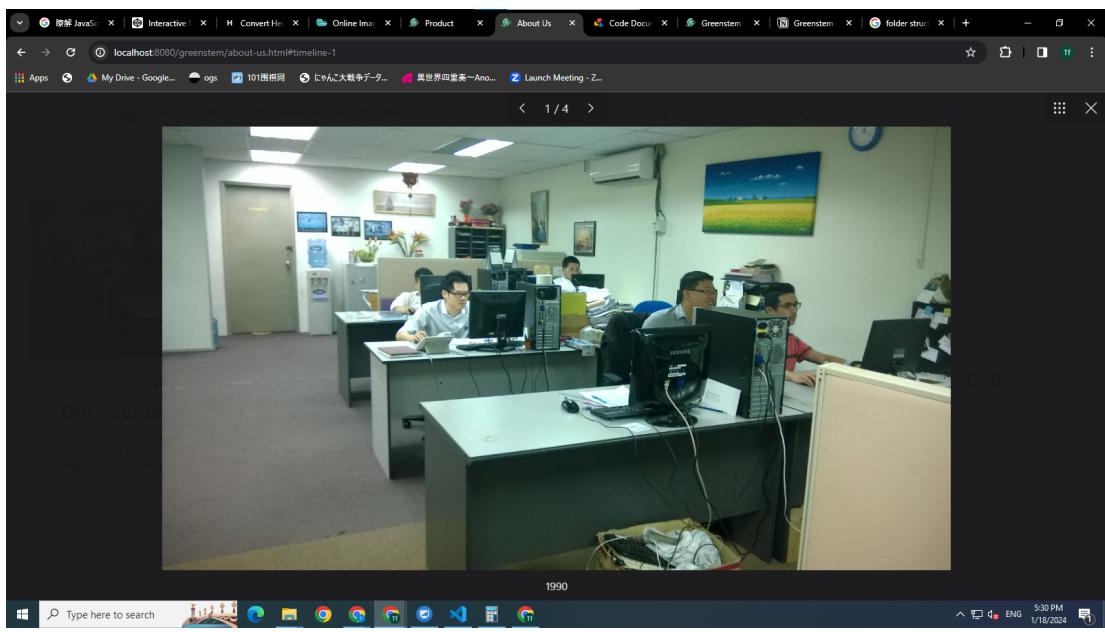
img {
    width: 100%;
    height: auto;
}

```

- The images used for the event gallery should follow the naming format `i.jpg`, starting from 1 and increasing incrementally. Similarly, for the general gallery, use the format `gallery_i.jpg`, with 'i' starting from 1 and increasing

sequentially. This structure allows for efficient implementation using a for loop function.

- All images, including background images, are equipped with the capability to zoom in when clicked, utilizing the Fancybox function. To enable this feature, add the 'gallery' class to the element and include the attributes `data-fancybox=""` with the specified category, `data-caption=""` to add a caption, and `data-src=""` for background images to specify the source.



effect

```
<div  
    class="project-pic four  
    gallery"  
    loading="lazy"  
    data-fancybox="gallery2"  
    data-src="../greenstem/p  
    ic/project-4.JPG"  
></div>
```

```

```

## ▼ Page Layout

- Navigation, footer, and all third-party libraries are consistent across all pages. To enhance readability and ease of modification, these common elements are encapsulated in separate files: 'navigation.php', 'footer.php', and 'link.php'. These files are then imported into each page, ensuring uniformity and simplifying maintenance.

```
<?php include("link.php"); ?>
```

- The page layout consists of an HTML structure divided into head and body sections. The head contains the title, links to libraries, and styles. The body is further segmented into three main parts: a navigation bar, a content container, and a footer. The navigation bar provides general navigation elements, while the content container holds specific content sections. Each section is enclosed in a div container identified by its title. The JavaScript file link is positioned at the end of the body, while any special JavaScript code is placed at the top of the body for optimal script execution.

```

<body>
    <script>
        //to make the header 1 benn expended as default (why select us)
        $(document).ready(function () {
            $("#header1").click();
        });
    </script>

    <?php
    include("navigation.php");
    ?>

    <div class="content container-fluid">
        <div class="wave-container">...
        </div>

        <div class="title">...
        </div>

        <div class="feature">...
        </div>

        <div class="why-select-us animation">...
        </div>

        <div class="product-container">...
        </div>

        <div class="after-sales-service">...
        </div>

        <div class="project-preview">...
        </div>

        <div class="feedback">...
        </div>

    </div>

    <?php
    include('footer.php');
    ?>

    <script src="/source/script.js"></script>
</body>

```

## ▼ General Animation

All elements are configured to have an entering animation when first loaded onto the screen by appending the 'animation' class to their respective HTML classes. This ensures a visually appealing transition for users upon the initial display of each element.

To modify the entering animation effect, simply adjust the class in the JavaScript code. For example, changing

```
entry.target.classList.add('animate__animated',
'animate__fadeInUp');
```

to a different effect

```
.animation {
    animation-duration: 0.5s;
    opacity: 0;
}
```

```
<div class="why-select-us animation"></div>
```

from the [Animate.css library](#), such as `animate__fadeInDown`, will apply an alternative animation upon the element's entrance.

```
const the_animation = document.querySelectorAll('.animation');

the_animation.forEach((element) => {
    const threshold = 0.5; // Default threshold is 0.5
    const observer = new IntersectionObserver(entries) => {
        entries.forEach((entry) => {
            if (entry.isIntersecting) {
                entry.target.classList.remove('animation');
                entry.target.classList.add('animate__animated', 'animate__fadeInUp');
                observer.unobserve(entry.target); // Stop observing the element
            }
        });
    }, { threshold: threshold });

    observer.observe(element);
});
```

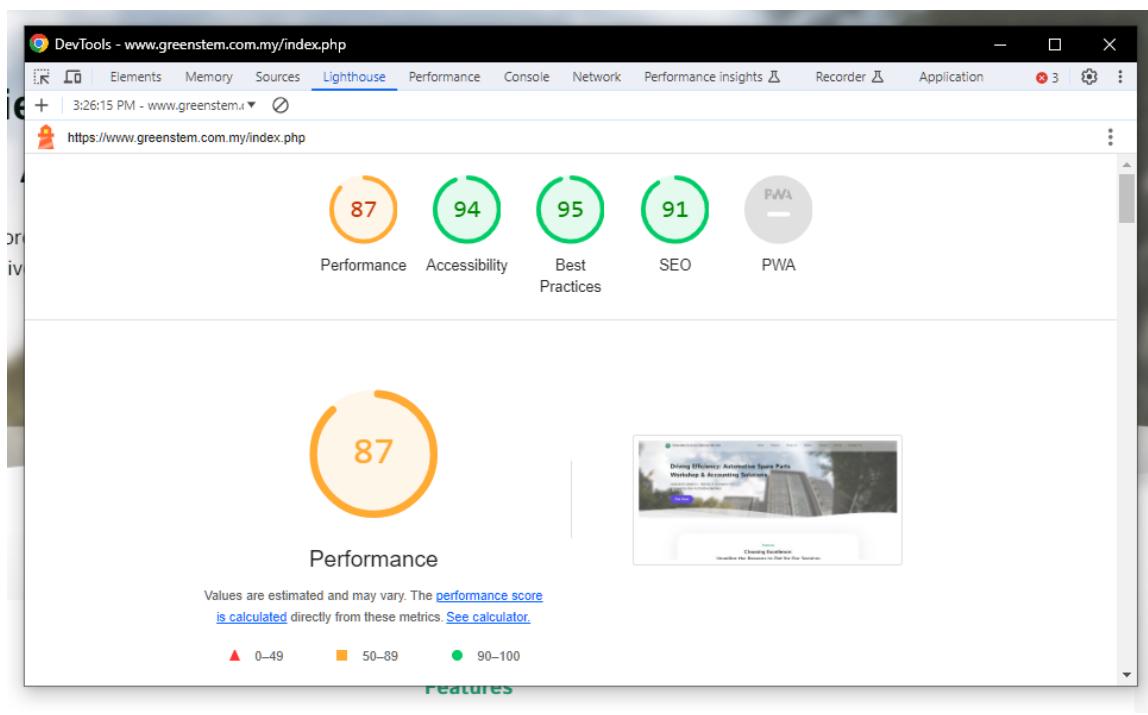
## Responsiveness

The website is designed to cater to two types of views: mobile (maximum width = 991.98px) and desktop. Each page has a general style defined in 'style.css', and additional customization is implemented in respective '(pagename).css' files for specific pages. For mobile views, the 'device.css' stylesheet is applied to ensure optimal styling and responsiveness.

# Report

- Lighthouse Report

The website underwent a Lighthouse performance audit. The results provide insights into various aspects such as performance, accessibility, best practices, SEO, and more. Each metric is thoroughly examined to ensure optimal website performance and user experience.



run on 2/2/2024

- **Structured Data Test**

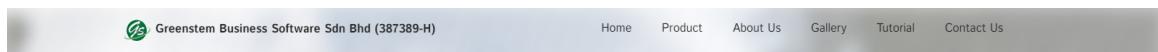
A Structured Data Test assesses the website's implementation of structured data, ensuring that the information is presented in a format that search engines can easily understand. This test verifies the accuracy and effectiveness of the structured data markup, contributing to improved search engine visibility and rich result display.

The screenshot shows the Google Rich Results Test tool interface. At the top, a green checkmark icon indicates "4 valid items detected". Below it, a message states "Valid items are eligible for Google Search's rich results. [Learn more](#)". There are two buttons: "VIEW TESTED PAGE" and "PREVIEW RESULTS". Under the "Details" section, there is a "Crawl" section with a green checkmark and the text "Crawled successfully on Feb 2, 2024, 3:24:05 PM". Below that, under "Detected structured data", there are two items: "FAQ" with "1 valid item detected" and "Local businesses" with "1 valid item detected" and a note "Non-critical issues detected".

<https://search.google.com/test/rich-results/result?id=VT6LvTjz9jT1VIIM7AxAyg>

## Detail for each Page

### ▼ Navigation Bar



The navigation bar is built using Bootstrap and incorporates custom CSS styles and JavaScript for additional effects. It includes links to various pages, providing seamless navigation throughout the website.

The navigation bar's behavior is dynamically controlled by the JavaScript code. When users scroll down, the navigation bar smoothly hides (`transform: translateY(-100%)`), optimizing screen space. Conversely, when users scroll up or hover near the top of the page, the navigation bar gracefully reappears (`transform: translateY(0)`), ensuring easy access to navigation options.

```
document.addEventListener('DOMContentLoaded', function () {
    const navbar = document.getElementById('navigation');
```

```
let lastScrollPosition = 0;

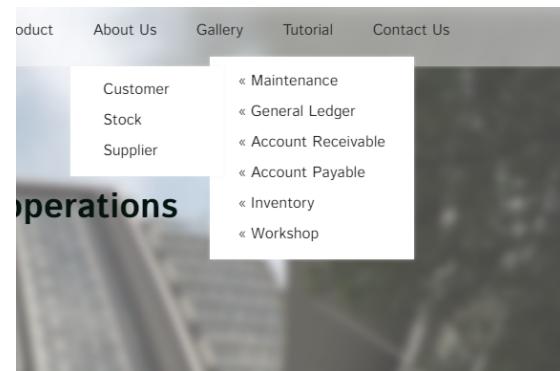
function checkScroll() {
    const scrollTop = window.scrollY;

    if (scrollTop > lastScrollPosition) {
        // Scrolling down
        navbar.style.transform = 'translateY(-10
0%)';
    } else {
        // Scrolling up
        navbar.style.transform = 'translateY(0)';
    }

    lastScrollPosition = scrollTop;
}

window.addEventListener('scroll', checkScroll);
window.addEventListener('mousemove', function (even
t) {
    // Show the navigation bar when hovering near th
e top
    if (event.clientY <= 10) {
        navbar.style.transform = 'translateY(0)';
    }
});
});
```

The Tutorial link, initially set to expand by click, has been modified to expand by hover. This behavior is achieved through the implementation of the following code:



```

document.addEventListener('DOMContentLoaded', function () {
    var navbar = document.getElementById('navigation');
    var productDropDown = document.getElementById('navdropdownProduct');
    var product_submenu = productDropDown.nextElementSibling;
    var tutorialDropDown = document.getElementById('navdropdownTutorial');
    var tutorial_submenu = tutorialDropDown.nextElementSibling;

    // Function to check if an element is part of the dropdown or its submenu
    function isDropdownElement(element) {
        return (
            element.classList.contains('nav-item') ||
            element.closest('.nav-item') !== null
        );
    }

    function showDropdown(dropdown, submenu) {
        dropdown.setAttribute('aria-expanded', 'true');
        submenu.style.display = 'block';
    }
}

```

```

        function hideDropdown(dropdown, submenu) {
            dropdown.setAttribute('aria-expanded', 'false');
            submenu.style.display = 'none';
        }

        navbar.addEventListener('mouseleave', function (event) {
            if (!isDropdownElement(event.relatedTarget)) {
                // Close the dropdowns
                hideDropdown(productDropDown, productSubMenu);
                hideDropdown(tutorialDropDown, tutorialSubMenu);
            }
        });

        productDropDown.addEventListener('mouseenter', function () {
            // Show the "Product" dropdown when entering the navbar
            showDropdown(productDropDown, productSubMenu);
            hideDropdown(tutorialDropDown, tutorialSubMenu);
        });

        productDropDown.addEventListener('mouseleave', function () {
            // Close the "Product" dropdown when leaving the navbar
            hideDropdown(productDropDown, productSubMenu);
        });

        tutorialDropDown.addEventListener('mouseenter', function () {
            // Show the "Tutorial" dropdown when entering the navbar
            showDropdown(tutorialDropDown, tutorialSubMenu);
            hideDropdown(productDropDown, productSubMenu);
        });
    
```

```

    tutorialDropDown.addEventListener('mouseleave', function () {
        // Close the "Tutorial" dropdown when leaving the navbar
        hideDropdown(tutorialDropDown, tutorial_submenu);
    });

    // Handle mouseover and mouseout events on the submenus
    product_submenu.addEventListener('mouseover', function () {
        // Keep the "Product" dropdown open when over the submenu
        showDropdown(productDropDown, product_submenu);
    });

    product_submenu.addEventListener('mouseout', function (event) {
        // Close the "Product" dropdown only if the mouse is not over the submenu or the dropdown button
        if (!isDropdownElement(event.relatedTarget)) {
            hideDropdown(productDropDown, product_submenu);
        }
    });
}

tutorial_submenu.addEventListener('mouseover', function () {
    // Keep the "Tutorial" dropdown open when over the submenu
    showDropdown(tutorialDropDown, tutorial_submenu);
});

tutorial_submenu.addEventListener('mouseout', function (event) {
    // Close the "Tutorial" dropdown only if the mouse is not over the submenu or the dropdown button
}

```

```
        if (!isDropdownElement(event.relatedTarget)) {
            hideDropdown(tutorialDropDown, tutorialSubme
nu);
        }
    });
});
});
```

The JavaScript code enhances the functionality of the Tutorial link, transforming it into a multi-drop-down button that expands on hover instead of click. The code employs event listeners to manage the dropdown behavior effectively. When users hover over the dropdown button, it dynamically expands, revealing a submenu. Importantly, the dropdown remains open when the mouse is over the submenu, ensuring a seamless and user-friendly experience. On mouse leave events, the dropdown reverts to its closed state, providing intuitive navigation. This modification optimizes the user interface, offering a responsive and efficient way for users to explore the content associated with the Tutorial link.

```
document.querySelectorAll('.nav-dropdown').forEach(function (element) {
    element.addEventListener('click', function (e) {
        e.stopPropagation();
    });
})
```

The JavaScript code utilizes the `querySelectorAll` method to target all elements with the class 'nav-dropdown.' Each element is assigned an event listener to prevent the automatic closure of the dropdown menu when clicking inside its container. This is achieved by using the `stopPropagation` method, ensuring that click events within the dropdown container do not propagate to the document and, consequently, do not trigger the closing of the dropdown.

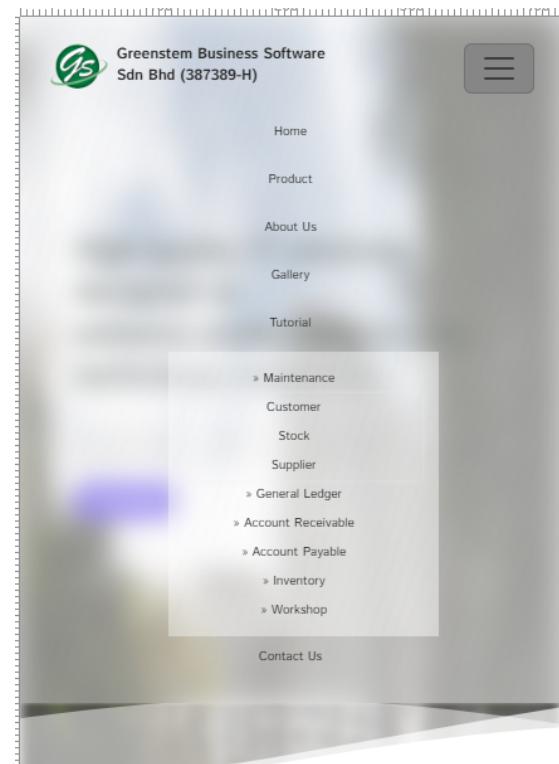
All navigation links in the menu are designed with a hover effect that elegantly underlines the link text. The following CSS code is responsible for this effect:



```
nav .nav-link:hover {  
    transition: all 300ms linear;  
}  
  
nav .nav-link::after {  
    display: block;  
    margin: 0 auto;  
    margin-top: 5px;  
    width: 0;  
    height: 2px;  
    background-color: #282828;  
    content: " ";  
    opacity: 0;  
    -webkit-transition: width 0.6s, opacity 0.8s;  
    -moz-transition: width 0.6s, opacity 0.8s;  
    transition: all 300ms linear;  
}  
  
nav .nav-link:hover::after {  
    opacity: 1;  
    width: 100%;  
}
```

By default in Bootstrap, when the screen width falls below 992px, the navigation links are automatically transformed into buttons. This

responsive behavior ensures an optimized user experience on smaller screens, allowing for easier navigation through a button-based interface.



## ▼ Footer

(6)03 6263 3933  
Hunting Line

admin@greenstem.com.my  
Free Consult

A-3-3A, Block A, Ativo Plaza,  
No. 1, Jalan PJU 9/1, Bandar Sri  
Damansara, 52200 Kuala Lumpur.

---

**Greenstem Business Software Sdn Bhd**

Elevate your business with Greenstem, delivering top-tier web, mobile, and desktop software solutions, including Accounting and Inventory control. Committed to customer satisfaction and continual innovation, we're your dedicated partner for success.

**PAGES**

- > Home
- > Product
- > About Us

**TUTORIAL**

- > Maintenance
- > General Ledger
- > Account Receivable
- > Account Payable
- > Inventory
- > Workshop

**Social Media**

©2024 Greenstem Business Software Sdn Bhd (387389-H) - All rights reserved

The footer is thoughtfully crafted to house essential company contact details, including the phone number, email address, and physical location. Each piece of information is interactive, allowing users to initiate a phone call, send an email, or view the location on a map by simply clicking on the respective elements. Additionally, the footer provides links to other pages within the website and integrates social media links for seamless connectivity. The concluding element is the copyright notice, providing legal acknowledgment of the company's intellectual property.

The Tutorial link is designed as a dropdown, activated by a click event that invokes the `showDropdown` function. When a user clicks the Tutorial link, this function is called, dynamically expanding the dropdown by adding the 'expanded' class to the selected button. To ensure a clean user experience, the function also ensures that only one dropdown is expanded at a time. This is achieved by removing the 'expanded' class from other buttons with the class 'dbtn'. The addition of 'dbtn' to the class list facilitates the identification of Tutorial link elements associated with the dropdown functionality.

```
function showDropdown(selected
Button){
    var buttons = document.querySelectorAll(".dbtn");

    //to make sure only one button been expended by removing
    //the expended class to other button
    buttons.forEach(function(button){
        if (button !== selecte
dButton){
            button.classList.r
emove('expanded');
        }
    })
}
```

## TUTORIAL

- > Maintenance
- > General Ledger
- > Account Receivable
  - > Delivery Order
  - > Direct Invoice
  - > DO-Invoice Generator
  - > Invoice Posting
  - > Receivable Credit Note
  - > Receivable Debit Note
  - > Receivable Receipt
  - > Receivable Report
- > Account Payable
- > Inventory
- > Workshop

```
<li onclick="showDr
opdown(this)" id="m
aintenanceBtn" clas
s="dbtn video">
```

```
//toggle the expanded class to selected button
selectedButton.classList.toggle('expanded');
}
```

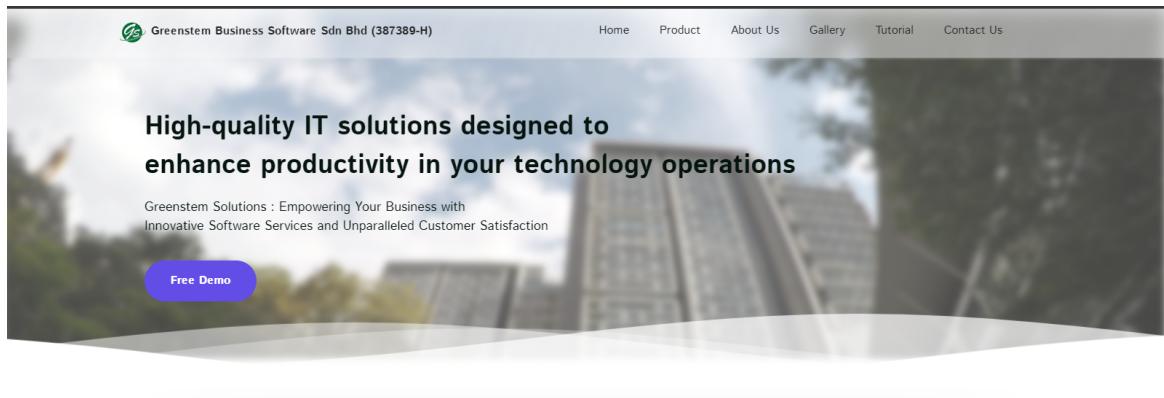
The copyright statement is dynamically generated using JavaScript to always display the current year, ensuring that the copyright information is automatically updated without manual intervention.

©2024 Greenstem Business Software Sdn Bhd (387389-H) - All rights reserved

```
document.addEventListener('DOMContentLoaded', function
() {
    var year = new Date().getFullYear();

    document.getElementById("copyRight").textContent =
'©' + year + ' Greenstem Business Software Sdn Bhd (3873
89-H) - All rights reserved';
})
```

## ▼ Index/ Home Page



The background effect is elegantly implemented using the `::before` pseudo-element of the 'wave-container.' The visual enhancement involves applying a filter of `blur(5px)` for a subtle blurring effect and setting the opacity to 75%. This combination achieves a visually pleasing and softened background, adding a touch of sophistication to the overall design.

```
.wave-container::before{  
    width: 100%;  
    height: 100%;  
    position: absolute;  
    top: 0;  
    left: 0;  
    content: '';  
    background-image: url  
( './greenstem/pic/GOPR45  
65.JPG' );  
    background-repeat: no  
-repeat;  
    background-size: 100%  
125%;  
    background-attachmen  
t: fixed;  
    filter: blur(5px);  
    opacity: 0.75;  
}
```

The captivating wave effect is achieved through an SVG element, enhanced with CSS animation. The timing of the animation is adjustable using CSS, providing flexibility to control the speed and fluidity of the waves. Additionally, the color and height of the waves can be fine-tuned by adjusting the values within the SVG,

allowing for easy customization to seamlessly integrate with the overall design aesthetics.

```
<svg class="waves" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
      viewBox="0 24 150 28" preserveAspectRatio="none" shape-rendering="auto">
  <defs>
    <path id="gentle-wave"
          d="M-160 44c30 0 58-18 88-18s 58
          88 18 58-18 88-18 58 18 88 18 v44h352z" />
  </defs>
  <g class="parallax">
    <use xlink:href="#gentle-wave" x="4
8" y="0" fill="rgba(255,255,255,0.7)" />
    <use xlink:href="#gentle-wave" x="4
8" y="3" fill="rgba(255,255,255,0.5)" />
    <use xlink:href="#gentle-wave" x="4
8" y="5" fill="rgba(255,255,255,0.3)" />
    <use xlink:href="#gentle-wave" x="4
8" y="7" fill="#fff" />
  </g>
</svg>
```

```
.parallax>use {
  animation: move-forever 25s cubic-bezier(.55, .5, .4
5, .5) infinite;
}

.parallax>use:nth-child(1) {
  animation-delay: -2s;
  animation-duration: 7s;
}

.parallax>use:nth-child(2) {
  animation-delay: -3s;
  animation-duration: 10s;
}
```

```

.parallax>use:nth-child(3) {
    animation-delay: -4s;
    animation-duration: 13s;
}

.parallax>use:nth-child(4) {
    animation-delay: -5s;
    animation-duration: 20s;
}

@keyframes move-forever {
    0% {
        transform: translate3d(-90px, 0, 0);
    }

    100% {
        transform: translate3d(85px, 0, 0);
    }
}

```

**WHY US**

**Successful Empowering 1000+**

**Spare Parts Companies**

Choose Greenstem for unparalleled software solutions that elevate your business. Our commitment to customer satisfaction, innovative approach, and a seasoned team set us apart. Partner with us for excellence and drive your success forward.

- + Dedicated Teams
- + Certified Professional
- Expert Support Team

Rely on Greenstem's seasoned support team for a seamless experience. With a wealth of expertise, we guarantee dedicated assistance to enhance your journey with our web, mobile, and desktop solutions.



The "Why Select Us" section is designed to showcase the company's strengths. Upon clicking on each accordion header, the `"toggleAccordion"` function is invoked, revealing detailed content along with a representative image. The image display is facilitated by specifying the image's ID as the attribute `'data-target-pic.'` This interactive approach ensures a user-centric experience, allowing visitors to explore specific strengths with corresponding visuals at their own pace.

```
<div class="accordion-item">
  <div
    class="accordion-header"
    id="header3"
    onclick="toggleAccordion(this, 'collapse3')"
    data-target-pic="pic"
  >
    <span class="accordion-symbol material-symbols-outlined">add</span>
    <span class="accordion-text">Expert Support Team</span>
  </div>

  <div class="accordion-body" id="collapse3">
    Rely on Greenstem's seasoned support team for a seamless experience. With a wealth of expertise, we guarantee dedicated assistance to enhance your journey with our web, mobile, and desktop solutions.
  </div>
</div>
```

```
<div class="pic" id="pic1" loading="lazy"></div>
```

```
#pic1 {
  background: url("../greenstem/pic/team.jpg") 50% / cover no-repeat;
  border-radius: 5px;
  width: 80%;
}
```

```

// Function to toggle the accordion item and adjust associated styles
function toggleAccordion(selectedButton, selectedContent) {
    // Get references to the selected content, all accordion items, and all images
    var content = document.getElementById(selectedContent);
    var category = document.querySelectorAll('.accordion-item');
    var pics = document.querySelectorAll('.pic');
    var targetPic = document.getElementById(selectedButton.getAttribute('data-target-pic'));

    // Close all accordion items
    category.forEach(item => {
        var body = item.querySelector('.accordion-body');
        body.classList.remove('expanded');
        body.style.maxHeight = '0';
    });

    // Toggle the selected accordion item's visibility
    content.classList.toggle('expanded');
    content.style.maxHeight = content.classList.contains('expanded') ? content.scrollHeight + 'px' : '0';

    // Adjust image widths based on the selected accordion item
    pics.forEach(function(pic) {
        if (pic !== targetPic) {
            pic.style.width = '10%';
        } else {
            pic.style.width = '80%';
        }
    });

    // Update symbols to indicate expand or collapse state
}

```

```

te
    var selectedSymbol = selectedButton.querySelector('.accordion-symbol');
    if (!(selectedSymbol.textContent === 'remove')) updateAccordionSymbol(selectedSymbol);
}

// Function to update the selected accordion symbol
function updateAccordionSymbol(selectedSymbol) {
    // Get all symbols and update their content and styling
    var symbols = document.querySelectorAll('.accordion-symbol');
    symbols.forEach(symbol => {
        symbol.textContent = symbol === selectedSymbol ? 'remove' : 'add';
        symbol.classList.toggle('expanded', symbol === selectedSymbol);
    });
}

// Function to expand the accordion content when clicking on an image
document.addEventListener('DOMContentLoaded', function () {
    // Get all images
    var pics = document.querySelectorAll('.pic');

    // Attach click event listeners to each image
    if (pics) {
        pics.forEach(function (pic) {
            pic.addEventListener('click', () => {
                // Get the associated accordion header and simulate a click event
                var headerId = pic.getAttribute('data-header');
                var header = document.getElementById(headerId);

```

```
        $(document).ready(function () {
            $(header).click();
        });
    });
}
});
```

The animation code for the number '1000' is customizable, and you can adjust its details by modifying the settings in the JavaScript code.

```
let start = 20; // Set the starting value
let end = 1000; // Set the ending value
let ticks = 10; // Set the number of animation steps
let speed = 100; // Set the speed of the animation
```

```
document.addEventListener("DOMContentLoaded", function () {
    // Select the element with the id 'number'
    let number = document.getElementById('number');

    // Set the intersection threshold for triggering the animation
    const threshold = 1;

    // Flag to track if the animation has been executed to prevent multiple executions
    let animationExecuted = false;

    // Check if the element with id 'number' exists
    if (number) {
        // Create an IntersectionObserver to handle element visibility changes
```

```

        const observer = new IntersectionObserver((entries) => {
            // Iterate through each entry (element visibility status)
            entries.forEach((entry) => {
                // Check if the element is currently intersecting with the viewport and animation hasn't been executed
                if (entry.isIntersecting && !animationExecuted) {
                    // Set the flag to indicate that the animation is being executed
                    animationExecuted = true;

                    // Define parameters for the random number animation
                    let start = 20;
                    let end = 1000;
                    let ticks = 10;
                    let speed = 100;

                    // Generate an array of random numbers within the specified range
                    let randomNumbers = [end];
                    for (let i = 0; i < ticks - 1; i++) {
                        randomNumbers.unshift(
                            Math.floor(Math.random() *
                            (end - start + 1) + start)
                        );
                    }

                    // Sort the array in ascending order
                    randomNumbers.sort((a, b) => { return a - b });
                }
            });
        });

        // Initialize a counter for the animation steps
    
```

```

        let x = 0;

        // Set up an interval to update the
        'number' element with random numbers
        let interval = setInterval(function
() {
            // Update the 'number' element w
            ith the next random number
            if (number) {
                number.innerHTML = randomNum
bers.shift();
            }

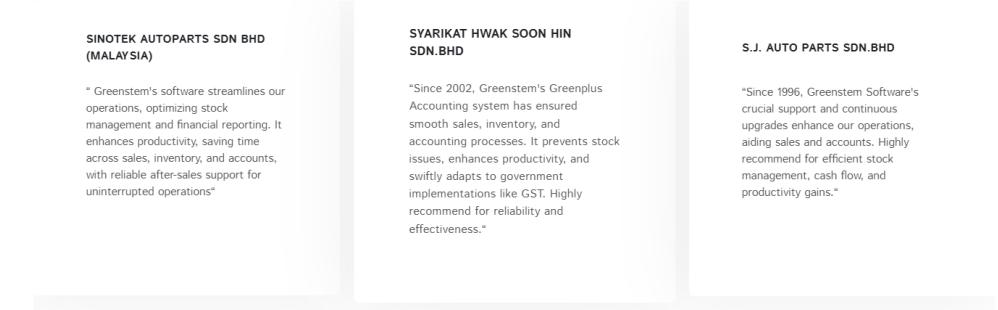
            // Check if the animation steps
            are complete
            if (++x === ticks) {
                // Clear the interval to sto
                p the animation
                window.clearInterval(interva
1);
            }
        }, speed);
    }
});

}, { threshold: threshold })); // Configure the o
bserver with the specified threshold

// Start observing the 'number' element for visi
bility changes
observer.observe(number);
}
});

```

## Let's See What Our Clients Say



The "Testimonial" section is implemented as a carousel, leveraging the Swiper.js library for a dynamic and interactive testimonial slideshow. To include additional slides, simply create a new container with the class name `"swiper-slide"` within the `"swiper-wrapper"` container. This allows for easy expansion and customization of the testimonial carousel by adding more slides as needed

```
<div class="swiper-slid  
e">  
  <span class="review-hea  
der">  
    HUA HONG AUTO SUPPLIE  
S <br />  
    (KUALA LUMPUR) SDN.BH  
D  
  </span>  
  
  <span class="review-bod  
y">  
    "Greenstem's software  
streamlines operations, i  
mproving stock management  
and  
    financial reporting.  
It enhances productivity  
across sales, inventory,  
and  
    accounts, with reliab  
le after-sales support fo
```

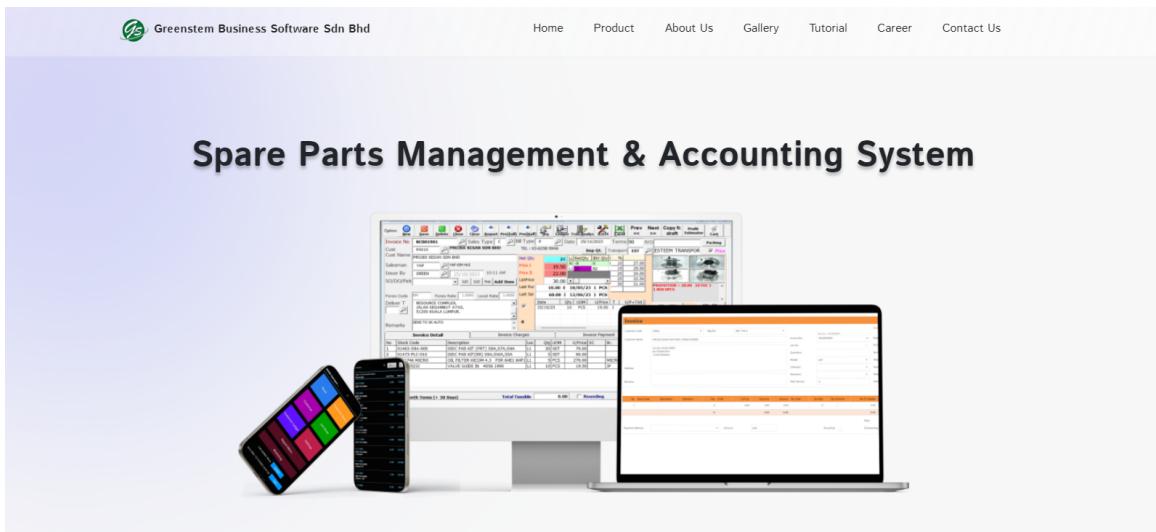
```
new Swiper(".customer-rev  
iew", {  
  slidesPerView: isPhone  
? 1 : 3,  
  spaceBetween: 30,  
  freeMode: true,  
  pagination: {  
    el: ".swiper-paginati  
on",  
    clickable: true,  
  },  
  loop: true,  
  centeredSlides: true,  
  autoplay: {  
    delay: 3500,  
    disableOnInteraction:  
false,  
  },  
});
```

```

    r uninterrupted operation
    s."
</span>
</div>

```

## ▼ Product Page



The title typing effect is managed by a JavaScript function, allowing customization of the displayed word and adjustment of the typing animation timing by modifying the '`data-text`' attribute value for the element with id = `text`

```

document.addEventListener('DOMContentLoaded', function () {
  consoleText('text');

  // Function to create a typing animation effect
  function consoleText(id) {
    // Get references to the console container and set
    var target = document.getElementById(id);
    var word = target.getAttribute('data-text');

    // Check if the target element exists
    if (!target) {
      // Exit the function if the target element is r
      return;
    }
  }
}

```

```

// Function to simulate typing characters
function type() {
    // Check if there are more letters in the current word
    if (letterCount <= word.length) {
        // Update the target element with the current letter
        target.innerHTML = word.substring(0, letterCount);
        letterCount += 1;

        // Set a timeout to continue typing after a delay
        setTimeout(type, 75);
    } else {
        // Display the full word without disappearing
        target.innerHTML = word;

        // Hide the cursor
        con.style.display = 'none';
    }
}

var letterCount = 0;
var con = document.getElementById('console');
type();
}
});

```

 Greenstem Business Software Sdn Bhd (387389-H)

Home   Product   About Us   Gallery   Tutorial   Contact Us

## Highlight Features

Experience unparalleled efficiency with Greenstem Software – transforming the way we manage operations, boost productivity, and drive success.

The feature animation is governed by an observer, which adds the `active` class when the element is first visible on the screen. Additionally, the observer is set to reset whenever there is a change in content. This approach ensures that the animation is triggered specifically when each feature becomes visible, providing a dynamic and engaging user experience

```
.features.active {  
    transform: scale(1);  
}
```

```
.features {  
    color: white;  
    position: absolute;  
    transform: scale(0);  
    transition: all 1s ea  
se;  
}
```

```
<div class="features color2 feature3 left">Data Migratio  
n from Other System</div>  
<div class="features color1 feature4 right">Provide Suff  
icient <br> Information for Decision Making</div>
```

The color tag is utilized to define distinct colors, where "color1" corresponds to the background color `#29A87A`, and "color2" corresponds to `#6046FF`.

```
.features.color1 {  
    border-radius: 20px;  
    background: #29A87A;  
    box-shadow: 0px 4px 1  
00px 0px #DFDFDF;  
    font-size: 1vw;  
}
```

The left/right tag determines the transform origin of the animation.

```
.features.left {  
    transform-origin: rig  
ht bottom;  
}
```

```
.feature1 {  
    top: 15%;
```

The feature tag is employed to specify the size of the container and set the animation delay, contributing to the overall styling and animation effects of the feature.

```
left: 10%;  
width: 18vw;  
padding: 3vh 2.4vw;  
}
```

The screenshot shows a 'System Preview' window titled 'Invoice Entry'. It displays a form with various input fields and a large table of items. On the right side, there is a preview image of a product, which is a small cylindrical component. The code snippet above defines the styling for this preview area, specifically setting its position relative to the left, width, and padding.

The preview section is managed by the `productPreviewList()` function. When a button is clicked, this function is called with two parameters: the selected button and the ID of the chosen image. It adds the "active" class to both the button and the image, ensuring that the selected image is displayed and the button color is changed, creating a visual indicator for the user.

```
function productPreviewList(selectedButton, targetId) {  
    // Get references to all list-group items and produc  
    t preview images  
    var buttons = document.querySelectorAll('.list-group  
-item');  
    var imgList = document.querySelectorAll('.product-pr  
eview-img');  
  
    // Remove the 'active' class from all list-group ite  
ms and product preview images  
    buttons.forEach(function (button) {  
        button.classList.remove('active');
```

```

    });

    imgList.forEach(function (element) {
        element.classList.remove('active');
    });

    // Get the target image element based on the provided targetId
    var target = document.getElementById(targetId);

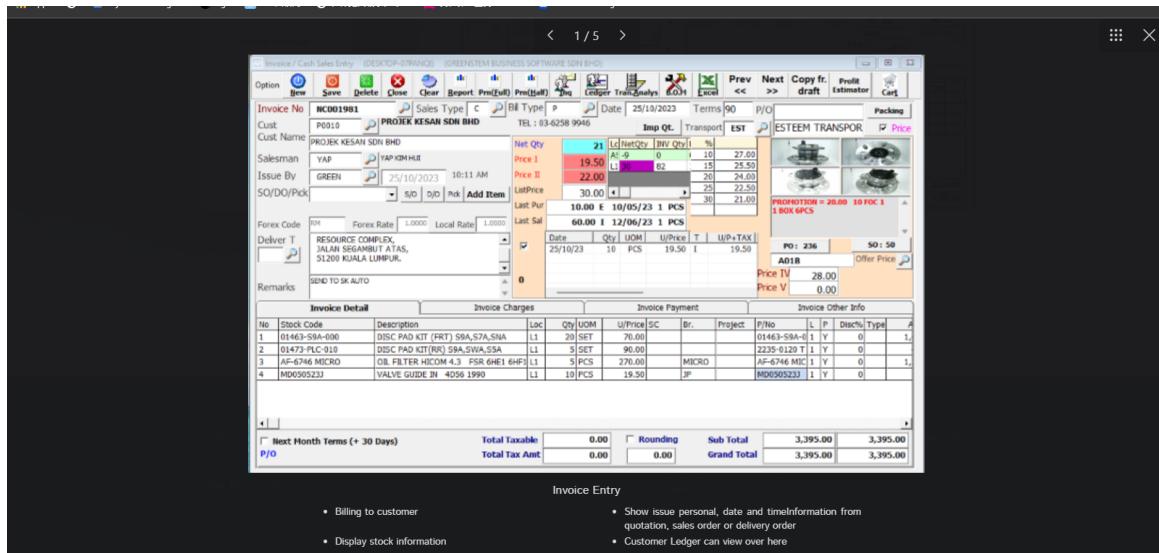
    // Check if the target element exists
    if (target) {
        // Add the 'active' class to the selected button and show the corresponding image
        selectedButton.classList.add('active');
        target.classList.add('active', 'animate__animate', 'animate__fadeIn');
    }
}

```

```

<li
    class="list-group-item"
    onclick="productPreviewList(this, 'SparePartSalesOrder')"
>
    <a>Sales Order</a>
</li>

```



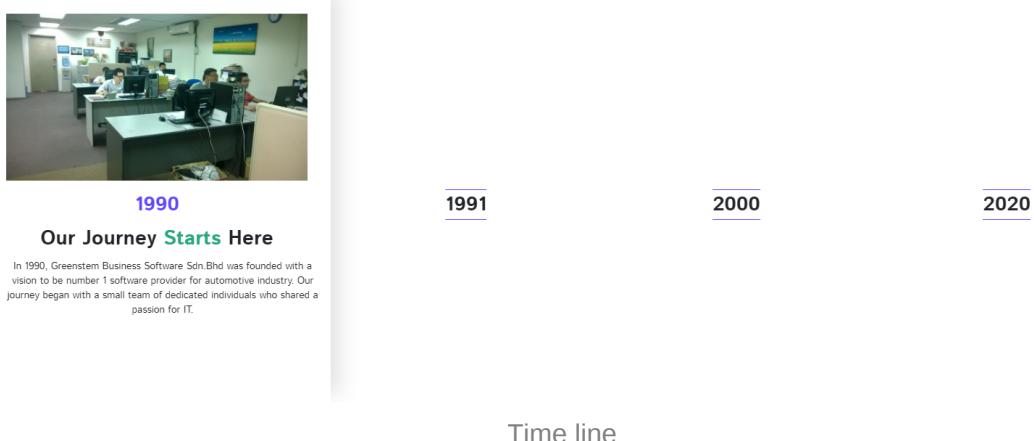
```

<div class="product-preview-img" id="SparePartSalesOrder">
    
                <li>Check stock information, price, picture , online quantity</li>
                <li>Sales history with date, customer name, quantity, unit price</li>
                <li>Issue sales order off line, send to office when internet available</li>
                <li>Print sales order in office to pick up</li>
                <li>Issue D/O or Invoice in office by extract data from sales order</li>
            </ul>">
    />
</div>

```

## ▼ About Us Page

## Historical Milestones



The timeline content expands when the user hovers over it, and this expansion is detected and managed by JavaScript to ensure that only one element is expanded at a time.

```
<div class="timeline-item">
  

  <h3 class="timeline-notation">1991</h3>

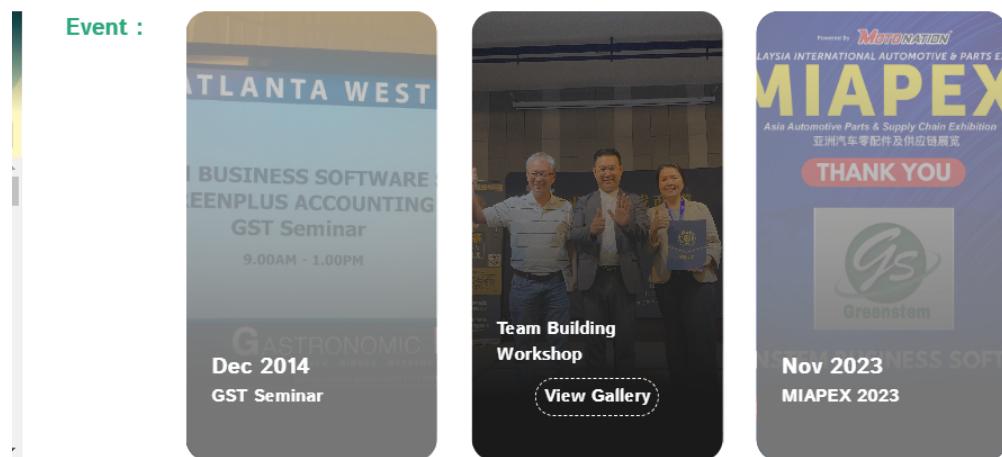
  <div class="timeline-content">
    <h3 class="timeline-content-title">Dos-Based System</h3>
    <p class="timeline-desc">
      We are poised for success as we embark on the journey to develop a cutting-edge accounting system based on the foundation of our decades of experience with DOS technology. Our dedicated team is committed to creating an innovative solution that will redefine
    </p>
  </div>
</div>
```

```
the way businesses  
    manage their finances.  
</p>  
</div>  
</div>
```

```
document.addEventListener('DOMContentLoaded', function  
() {  
    // Get all elements with the class 'timeline-item'  
    var timelineItems = document.querySelectorAll('.time  
line-item');  
  
    // Iterate through each timeline item  
    timelineItems.forEach(function (item) {  
        // Add a mouseover event listener to each timeli  
ne item  
        item.addEventListener('mouseover', function () {  
            // Add the 'hover' class to the current time  
line item  
            item.classList.add('hover');  
  
            // Iterate through all timeline items and re  
move the 'hover' class from others  
            timelineItems.forEach(function (otherItem) {  
                if (otherItem !== item) {  
                    otherItem.classList.remove('hover');  
                }  
            });  
        });  
    });  
});
```



The Facebook widget on the website utilizes the meta plugin tools, and additional settings can be referred to at <https://developers.facebook.com/docs/plugins/page-plugin/>.



The event container is built using Swiper.js and Fancybox. It allows users to select a category and view images by clicking the corresponding button.

```
<div class="swiper-slide event-slide">
  <div class="event-container team-building">
    <div class="event-title">
      <h3 class="event-date"></h3>
      <h3 class="event-name">Team Building Workshop</h3>
    </div>
    <button class="team-building-Btn">View Gallery</button>
```

```
on>
</div>
</div>
```

The background image is situated on the event-container with a distinctive title class name, and the button plays a role in the JavaScript settings. All content images are displayed using a JavaScript function, allowing for easy modification of the JS code to update the images.

```
document.addEventListener('DOMContentLoaded', function
() {

    function initializeFancybox(images) {
        return () => {
            new Fancybox(images, {
                Thumbs: { show.onStart: false },
                Carousel: { Navigation: false },
                Toolbar: {
                    display: {
                        left: [],
                        middle: ["prev", "infobar", "nex
t"],
                        right: ["thumbs", "close"],
                    },
                },
            });
        };
    }

    function setupGallery(buttonSelector, folderName, nu
mberOfImages) {
        var galleryButtons = document.querySelectorAll(b
uttonSelector);

        if (galleryButtons) {
            galleryButtons.forEach(function (button) {
                button.addEventListener('click', functio
n () {
```

```

        var imageObjects = [];

        for (var i = 1; i <= numberOfImages;
i++) {
            var imageObject = {
                src: "/pic/" + folderName +
"/" + i + ".jpg",
                type: "image",
                caption: folderName,
            };
            imageObjects.push(imageObject);
        }

        initializeFancybox(imageObjects)());
    });
}
}

setupGallery('.evm2023-btn', 'sept23_EVM-Asia23', 1
0);
setupGallery('.team-building-Btn', 'team_building',
7);
setupGallery('.miapex2023-btn', 'Nov23_Miapex-2023',
4);
setupGallery('.gstSeminar-btn', 'gst-seminar', 4);
});

```

The `setupGallery` function requires three parameters: the button's class name, folder name, and the number of images. It utilizes a for loop to automatically generate image objects for Fancybox settings. It is crucial to organize each event's images in separate folders and adhere to the specified naming format.



The gallery is constructed using Swiper.js, featuring a cover-flow effect and a zoom-in effect for the active image. To include additional images, simply add images with the class name "`gallery-image`" under the "`gallery-swiper`" container.

```
.gallery-swiper .swiper-slide-active .gallery-image {
    filter: grayscale(0);
    opacity: 1;
    width: 150%;
    transform: translateX(-15%);
    z-index: 1;
    position: relative;
}
```

```
<div class="gallery-image-container">
    <div class="swiper gallery-swiper">
        <div class="swiper-wrapper">
            <?php
                $galleryPath = "/pic/gallery/";
                $numberOfImages = 12;

                for ($i = 1; $i <= $numberOfImages
                    $imagePath = $galleryPath . "galle
                ?>
                    <div class="image-container swi
                    Register No Training Video Enhanced vers  
</a>
```

```
document.addEventListener("DOMContentLoaded", function  
() {  
    // Retrieve the videoPath from the URL query parameter  
    var urlParams = new URLSearchParams(window.location.search);  
    var videoPath = urlParams.get("videoPath");  
  
    if (videoPath) {  
        // Set the source of the video element based on the  
        // query parameter  
        var selectedVideo = document.getElementById("selected-video");  
        selectedVideo.src = ". ./video/" + videoPath;  
  
        // Load and play the video  
        selectedVideo.load();  
        selectedVideo.play();
```

```
    }  
});
```

## ▼ Contact Us Page

The image shows a contact form titled "Consultation for Free". It includes fields for Company Name, Your name, Phone Number, Email Address, and Message. There is also a checkbox for Privacy and Confidentiality Statement and a "Send" button. To the right of the form is a green decorative element with the text "Get in Touch Right Now!" and a subtext: "Reach us without delay to expand your understanding through additional inquiries." Below this is an illustration of a person sitting at a desk with a computer, surrounded by plants and a bird.

The form utilizes the PHP built-in mailer function to send an email once the user fills it out. All fields in the form are required, and upon submission, a status message for the email will be displayed. The variable `$email_acc` is stored in `link.php` for easy modification across all pages.

```
<?php  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $name = htmlspecialchars($_POST['name']);  
    $phoneNo = htmlspecialchars($_POST['phoneNo']);  
    $emailAddress = htmlspecialchars($_POST['emailAd  
dress']);  
    $messageContent = htmlspecialchars($_POST['mess  
age']);  
    $companyName = htmlspecialchars($_POST['compan  
yName']);  
  
    $to = $email_acc;  
    $subject = "Website Message";  
    $message =  
        "Company: " . $companyName . "\n" .  
        "Name: " . $name . "\n" .  
        "Phone Number: " . $phoneNo . "\n" .
```

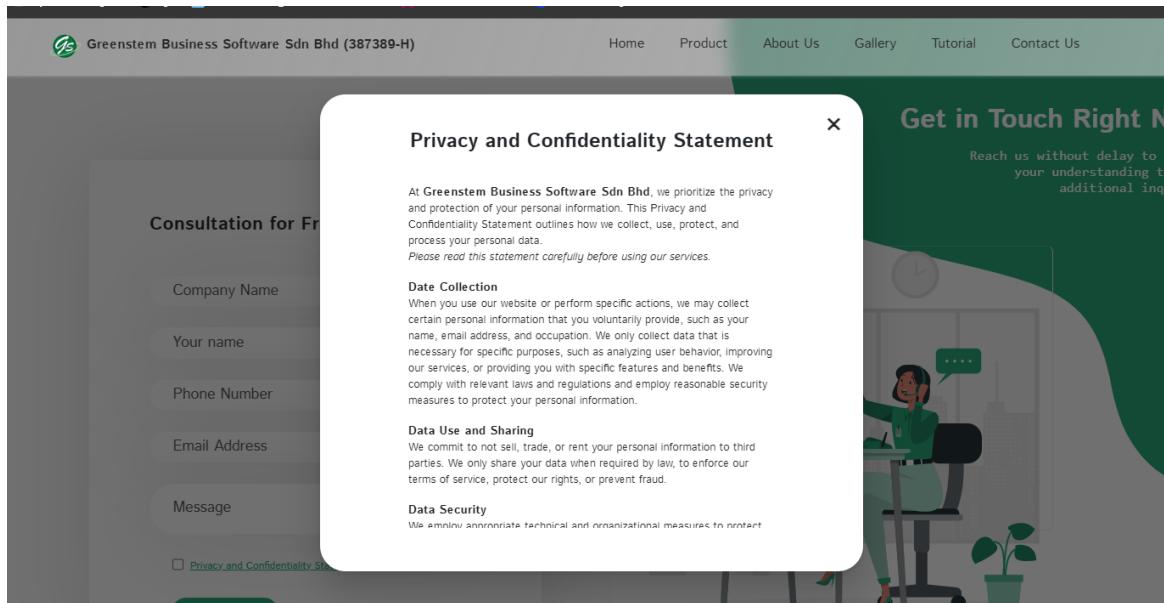
```

        "Email Address: " . $emailAddress . "\n\n" .
        "Message: " . $messageContent;
$headers = "From: admin@greenstem.com.my" . "\r
\n" .
        "Reply-To: " . $emailAddress;

        if (mail($to, $subject, $message, $headers)) {
            $statusMessage = 'Your message has been sent
successfully!';
        } else {
            $statusMessage = 'Oops! Something went wron
g. Please try again later.';
        }

        echo "<script>alert('$statusMessage');</script>";
    }
?>

```



The pop-up window is controlled by the `showPopup()` function. It determines the visibility of the content when the user clicks on the "Privacy and Confidentiality Statement" button. The window can be closed either by clicking the close button or outside the container.

```
function showPopUp(button) {
    var popUp = document.getElementById("popUpWindow");

    button.style.color = "#29a87a";
    button.style.textDecoration = "underline";

    popUp.classList.add("active");
}
```

```
function closePopUp() {
    var popUp = document.getElementById("popUpWindow");
    var closeButton = document.getElementById("popUpCloseBtn");

    if (event.target === popUp || event.target === closeButton) {
        popUp.classList.remove("active");
    }
}
```

```
document.addEventListener("DOMContentLoaded", function () {
    var popUp = document.getElementById("popUpWindow");

    if (popUp) {
        popUp.addEventListener("click", function () {
            closePopUp();
        });
    }
});
```

## F A Q

---

- 1. What Services Does Greenstem Offer? +
- 2. How Does Greenstem Prioritize Client Success? +
- 3. What is Greenstem's Approach to Innovation? +
- 4. What are Greenstem's Business Hours? +
- 5. How can I Contact Greenstem? +
- 6. How can I Obtain a Price Quote from Greenstem? +

The FAQ section is implemented using the same function as the homepage's "[why](#) [Select Us](#)" part, utilizing the `toggleAccordion()` function to control the visibility of content when clicking on the headers.

## ▼ Career Page

[Join Our Team](#)

\*Select Position

Email address

Name

\*Please attach a copy of your Resume or CV\*

Choose File No file chosen

[Send](#)

The job application form allows users to submit their applications with basic personal information such as name, email, and a resume/CV. It utilizes PHP's built-in mailer function to send an email to the admin account when users complete the form. The uploaded file is stored in the cPanel uploads folder for

record-keeping. The variable `$email_acc` is stored in `link.php` for easy modification across all pages.

```
<?php

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $position = $_POST['position'];
    $email = $_POST['careerEmail'];
    $name = $_POST['careerName'];

    // File upload handling
    $uploadDir = __DIR__ . '/uploads/';
    $uploadedFile = $_FILES['careerFile']['tmp_name'];
    $fileName = $_FILES['careerFile']['name'];

    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo '<script>alert("Invalid email address.");</scr
        exit; // Stop execution if validation fails
    }

    if (move_uploaded_file($uploadedFile, $uploadDir . $fil
        // Email content
        $to = $email_acc;
        $subject = 'Job Application Submission';
        $messageBody = "Position: $position <br>";
        $messageBody .= "Name: $name <br>";
        $messageBody .= "Email: $email <br><br>";

        // Attach the uploaded file to the email
        $fileContent = file_get_contents($uploadDir . $file
        $fileAttachment = chunk_split(base64_encode($fileCo

        $boundary = md5(time());
        $headers = "From: admin@greenstem.com.my\r\n";
        $headers .= "Reply-To: $email\r\n";
        $headers .= "MIME-Version: 1.0\r\n";
        $headers .= "Content-Type: multipart/mixed; boundar
        $message = "--$boundary\r\n";
```

```

$message .= "Content-Type: text/html; charset=\"utf
$message .= "Content-Transfer-Encoding: 7bit\r\n\r\n"
$message .= "$messageBody\r\n";
$message .= "--$boundary\r\n";
$message .= "Content-Type: application/pdf; name=\""
$message .= "Content-Transfer-Encoding: base64\r\n"
$message .= "Content-Disposition: attachment; filer
$message .= "$fileAttachment\r\n";
$message .= "--$boundary--";

// Send the email
if (mail($to, $subject, $message, $headers)) {
    echo '<script>alert("Application submitted succ
} else {
    echo '<script>alert("Application submission fai
}
} else {
    echo '<script>alert("File upload failed. Please cor
}
}

?>

```

The "Position" section is dynamically generated using a PHP array `$positions`. To update the list, simply modify the values in the array.

```

<select name="position" id="position" class="form-select" r
      <option selected disabled>*Select Position<
      <?php
          // Array of positions
          $positions = [
              "Software Sales Executive",
              "Web Programmer",
              "Apps Programmer",
              "UI/UX Designer",
              "Accounting System Support",
              "Technical Support"
          ];

```

```

        // Loop through the array and generate option
        foreach ($positions as $position) {
            echo '<option value="' . htmlspecialchars(
            )
        ?>
    </select>

```

The file upload section only accepts PDF format files with a maximum size of 10MB. A JavaScript function, `formValidation()`, is utilized to check the file size. If the size exceeds 10MB, it triggers an alert and prevents the submission process.

```

function formValidation() {
    const maxFileSize = 10 * 1024 * 1024 //10mb
    var fileInput = document.getElementById('careerFile');
    var fileSize = fileInput.files[0].size;

    if (fileSize > maxFileSize) {
        alert('File size exceeds the maximum allowed limit')
        return false; // Prevent form submission
    }

    return true;
}

```

## ▼ Additional JavaScript code

```

document.addEventListener('DOMContentLoaded', function
() {
    var txt = document.querySelector('.company-name');

    if (isPhone) {
        txt.innerHTML = "Greenstem Business Software <br
> Sdn Bhd (387389-H)";
    }
})

```

This JavaScript code adjusts the content of an HTML element with the class "company-name" during page load. If the variable `isPhone` is true, it sets a multi-

line company name to improve alignment on phone devices, likely enhancing the navigation bar's responsiveness.

```
document.addEventListener('DOMContentLoaded', function
() {
    var fb = document.getElementById('fbPage');

    if (fb && isPhone) {
        fb.setAttribute('data-width', 300);
    }
})
```

This JavaScript code modifies the Facebook page plugin on page load. If the variable `isPhone` is true, it sets the `data-width` attribute of the Facebook page element with the ID 'fbPage' to 300, potentially adjusting the width of the embedded Facebook page on phone devices for better alignment or responsiveness.

## Backup

To ensure the safety and preservation of your website's source code, regular backups are in place. The following backup methods are employed:

- **GitHub Version Control:**

The website's source code is maintained in a GitHub repository. Version control allows for tracking changes and ensures a secure backup.

<https://github.com/ThengFeng/greenstem.git>

- Local Server:

Additionally, a copy of the website files is stored locally for an offline backup, providing an extra layer of security.

\\GREEN-SRV\Greenstem\official website

## Special Thanks

- ChatGPT ( generate this documentation template )
- <https://caniuse.com/> (check the browser compatibility of different web technologies)
- <https://colorsui.com/> (generate color template)