

# Lab 4: ASP.NET Page Lifecycle and Web Form Integration

## Overview

This lab focuses on understanding the ASP.NET Page Lifecycle, the debugging process within the Visual Studio environment, and the transition from static HTML prototypes to dynamic Web Forms. Students will implement core functionalities for the Pine Valley Furniture Company web application.

---

## Problem 1: ASP.NET Page Lifecycle Demonstration

Develop an ASP.NET page to demonstrate the various events raised and handled during page processing once the page object has been created (e.g., `Page_Init`, `Page_Load`, `Control Events`, `Page_PreRender`).

## Requirements

- Develop the solution in Visual Studio and host it on your assigned web space.
- In the space provided below, draw the sequence of the page processing lifecycle.
- Provide a detailed description of each stage in the lifecycle based on class discussions and presentations.

## Problem 2: Debugging and Breakpoints

With reference to Problem 1, perform an experiment using the Visual Studio debugger. Learn how line-by-line code execution (stepping) is realized.

## Task

Write the specific benefits of using breakpoints and step-through debugging in the development of complex web applications below:

### Problem 3: From Prototypes to Web Forms

Refer to Lab 1, where paper prototyping and HTML interfaces were developed for the *Pine Valley Furniture Company*. Convert those plain HTML interfaces into functional ASP.NET Web Forms.

#### Functional Requirements

The application must incorporate the following modules:

- i) **New Customer Registration:** Input forms for user details.
- ii) **Searching of Products:** A search interface for furniture items.
- iii) **Products Selection and Order Placement:** Ability to pick items and checkout.
- iv) **Product Catalog Update:** Administrative functionality to modify the catalog.
- v) **Payment Provision:** A mock payment gateway interface.

### Problem 4: Implementing Search Mechanism

Implement the search functionality for furniture products identified in Problem 3(ii).

#### Logic Requirements

When a visitor enters a search string into an input control (e.g., `TextBox`) and clicks the search button, the application must filter the product list and display all relevant items.

#### Deliverable

Write the C# code for the Button Click Handler below:

## Problem 5: Testing and Validation

Develop and execute a set of test cases to validate the search functionality implemented in Problem 4.

### Test Case Table

Write the details of your test cases (Input, Expected Result, Actual Result, Status) in the table below:

Test Case ID	Input/Condition	Expected Outcome	Actual Result
TC-01	Empty Search String	Show all products	
TC-02	Valid Product Name	List matching items	
TC-03	Special Characters	Handle gracefully	
TC-04	Non-existent Item	"No results found" msg	