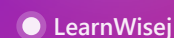


Designing the Application Shell

Design the outer structure of a real app — header, navigation, content area and status bar — and make it resize cleanly.



Lesson goal

Move from single screens to a simple **application shell** — the shared outer layout (header, navigation, content area, status bar) that stays on screen while the content changes.

1. What an application shell is

An **application shell** is the shared outer structure of a real app. The shell stays in place; only the content area swaps as the user moves around. A simple shell has four parts:

Shell part	Purpose	Beginner example
Header	App name, current user, and breadcrumb.	ServiceDesk Application · Signed in as: Support Agent
Left navigation	The pages the user can open.	Dashboard, Tickets, Customers, Settings
Content area	Displays the selected page content.	Dashboard cards or a Tickets list
Status area	Shows recent action or system feedback.	03:26 PM - Application shell loaded.

2. Page vs Window/Form1

Wisej lets you build the shell as a **Page** or as a **Window/Form1**. They create different user experiences.

Choice	What it feels like	When to use it
Page	A web page that fills the browser frame.	Best for a modern training app or dashboard shell.
Window/Form1	A desktop-like window inside the browser that can move, minimize, maximize and close.	Good for desktop-style tools, popups, or when you intentionally want window behavior.



Important UI habit

If you build the shell as a Window/Form1, **remove or disable the minimize button** unless you also build a taskbar or a clear way to restore it. A minimized browser window can look like the app disappeared, which confuses beginners.

3. Anchoring and docking keep the layout usable

Anchoring tells a control which sides of its parent to stay attached to. **Docking** makes a control fill a side or the remaining space. Together they keep the shell usable when the browser is resized.

Control area	Recommended layout	Why
Header panel	Dock = Top (or Anchor Top + Left + Right)	The header should stretch across the top.
Left navigation panel	Dock = Left (or Anchor Top + Bottom + Left)	Navigation stays on the left and keeps its height.
Content panel	Dock = Fill (or Anchor all four sides)	The current page uses the remaining space.
Status label/panel	Dock = Bottom (or Anchor Bottom + Left + Right)	Status messages stay visible at the bottom.
Navigation buttons	Anchor Top + Left + Right inside the nav panel	Buttons keep a clean, even width.

4. Navigate from one place

Don't scatter page-changing logic across every button. Instead, each button calls **one shared navigation method**. That keeps the app easy to debug and change later.

```

private void btnDashboard_Click(object sender, EventArgs e)
{
    NavigateTo("Dashboard");
}

private void btnTickets_Click(object sender, EventArgs e)
{
    NavigateTo("Tickets");
}

private void NavigateTo(string pageName)
{
    pnlContent.Controls.Clear();
    if (pageName == "Dashboard")
        pnlContent.Controls.Add(new DashboardView() { Dock = DockStyle.Fill });
    else if (pageName == "Tickets")
        pnlContent.Controls.Add(new TicketsView() { Dock = DockStyle.Fill });

    lblBreadcrumb.Text = "Home / " + pageName;
    lblStatus.Text = DateTime.Now.ToString("hh:mm:ss tt") + " - Opened " + pageName + "
page.";
}

```

One method clears the content panel, loads the right view, updates the **breadcrumb** ([Home](#) / [Tickets](#)) and writes a line to the **status** area.

5. Permissions as a design concept

Permissions don't have to start as a full login system. They can begin as a simple **role** idea: a Support Agent can *view* Settings but not save changes. Later this becomes real authentication and authorization.

Role	Dashboard	Tickets	Customers	Settings
Support Agent	View	Create/close	View	View only
Manager	View	Create/close	Create/edit	Save settings



Completion check

You can identify the header, navigation, content and status areas; use anchoring or docking instead of fixed positions; explain why Page and Window/Form1 differ; and route every nav button through one shared navigation method.