WISEJ.NET 2.0 CONTROLS AND A NORTHWIND APPLICATION · MODULE 1 · NAVIGATIONBAR FROM THE FIRST ITEM TO A WORKING MENU

AI Coding Exercise

Use the course-grounded prompt, review the generated patch and record verification evidence.

LearnWisej

AI Coding Exercise

Implement module 1 as a bounded patch, then verify the result against the lab acceptance criteria.

Before prompting

Provide your current practice page and the included example files named below. Keep a copy of the prompt, response, reviewed diff and verification evidence. This exercise works with a coding assistant of your choice.

Ready-to-use prompt

```
Help me learn Wisej.NET 2.0 controls in C#. Explain the properties and event path before implementing a change.
Module 1: NavigationBar From the First Item to a Working Menu
Use the original files included in this course: code/NorthwindCourse/NorthwindPage.cs. Ask for their contents if unavailable.
Compatibility: .NET Framework 4.8, Wisej-2 2.0.54 and Wisej-2-NavigationBar 2.0.54. The web application host comes from the installed Wisej 2 template. CollapsibleSideButton is supplied course code; TagTextBox is the tagged editor; Switch is a CheckBox appearance.
Task: Build a standalone NavigationBar practice page that switches between Overview and Orders and survives a compact-mode round trip.
Acceptance criteria:
1. Create and run the empty Wisej 2.0 practice project with matching package versions.
2. Add a left-docked NavigationBar and a fill-docked content Panel.
3. Create two items once and route ItemClick using e.Item identity.
4. Toggle CompactView in both directions and repeat each selection five times.
5. Capture both views, docking behavior and handler-count evidence.
Initialize controls once, keep mutable data per Page, use decimal amounts and apply the documented inclusive date rules. Explain the exact files to change. Give one small patch and independent tests with expected results. Do not claim a command or browser check ran unless it did.
Evidence: A build log showing the pinned package versions; A two-item menu and an explained ItemClick handler; Evidence that changing item labels preserves routing; Compact and expanded screenshots; Repeated-selection results with no extra controls or handlers.
Review questions: Which code owns each item identity? Why should initialization not run on every click? How will this menu select a business view in the final project?
```

Review and verify

- ✓ Create and run the empty Wisej 2.0 practice project with matching package versions.

- ✓ Add a left-docked NavigationBar and a fill-docked content Panel.
- ✓ Create two items once and route ItemClick using e.Item identity.
- ✓ Toggle CompactView in both directions and repeat each selection five times.
- ✓ Capture both views, docking behavior and handler-count evidence.

Challenge the generated result

- ✓ Which code owns each item identity?
- ✓ Why should initialization not run on every click?
- ✓ How will this menu select a business view in the final project?

Record your decision

For every proposed file, record accept, revise or reject with a reason. Record exact build output and observed browser results. A plausible answer is not evidence that the application ran.

Included course examples

`code/NorthwindCourse/NorthwindPage.cs`

These original files are included in the code download. Build the exercise before comparing with the completed example.