ISS Assignment 5

As we all know, the assignment is about transforming our WinUI app into a web app.

Yosua set up the project so that you can start working on it. Please note that this assignment is mostly a copy-paste work so do not rewrite things from scratch. Apart from the UI which will be in html and bootstrap (css library) you really should **NOT** rewrite code.

Key points for when you're working:

- Move your service and repository into the Class Library project
- Register your services and repositories using ASP.NET Core's built-in Dependency Injection:
  Add your interfaces and implementations to the DI container in Program.cs using
  services.AddScoped<>, AddSingleton<>, or AddTransient<> as appropriate.
  This allows your controllers to automatically receive service instances via constructor injection.
  Do not manually instantiate services or repositories in controllers or views — always rely on DI.
  An example of that would be
  https://github.com/cristicretu/UBB-SE-2025-MarketMessi/blob/main/MarketMinds.Web/Program.cs
  Lines 55-96.
  You're telling ASP.NET Core:
  "Whenever someone asks for IAuctionProductService, give them a new AuctionProductsService."

  We used to do this:
  IService service = new Service(new Repository());
  You're manually:
  1. Creating a repository
  2. Passing it to the service
  3. Maybe passing that service to a controller

  You own the construction logic.

  The Goal with DI: Let ASP.NET Core do that for you
  You just say:
  "Hey ASP.NET, I need an IService. You figure out how to build it — and give it to me ready to go."
  ASP.NET Core keeps track of all the pieces (services, repositories) you register. When it needs to create something (like a controller), it automatically builds and connects all its dependencies for you. You just ask for what you need, and it provides a fully ready object—no manual setup required.
- All the functionalities need to be guarded from unauthorised users (Robert's job)
- Be consistent on the UI. Use default styles, and if you want to do fancy stuff talk to the people who implement other stuff to see if they agree. The UI should be uniform across the entire application, not one red text page and one blue text page.