

Full-Stack Blazor & ASP.NET Engineer • Custom Web Applications • Scalable Architecture

I design and build full-stack applications using Blazor WebAssembly, ASP.NET, EF Core, and SQL. My work prioritizes structural clarity, predictable performance, and maintainability. I specialize in transforming loosely defined ideas into robust, production-ready systems with clean APIs, solid authentication flows, and modular architecture.

Core Technical Capabilities

- Blazor WebAssembly (UI, components, routing, state)
- ASP.NET backend with layered architecture
- REST API design + integration
- Authentication & Authorization (Identity, JWT, roles)
- Entity Framework Core + relational DB modeling
- Admin tooling (dashboards, logs, moderation systems)
- Data analytics pages (charts, aggregations, statistics)
- UI/UX logic for complex workflows
- Deployment to Render, Azure, GitHub Pages

Case Study: Personal Diary Application

Date	Entries count	Symbols count	Average symbols per entry	Total mood of day
10/30/2025	10	2023	202.3	-1
10/31/2025	1	227	227	1
11/8/2025	3	375	125	1
11/10/2025	1	505	505	-1
11/12/2025	1	53	53	1
11/20/2025	1	347	347	1
11/21/2025	2	904	452	-1
Totals:	19	4434	233	

Load statistics onto chart

Legend: Entries count in day, Symbols count in day, Average symbols per Entry in day, Total mood point in day

Nov 2 2025 Nov 9 Nov 16

How are this is working, if i am...

11/21/2025 Collapse

7:20 PM (Collapse)

I can write There some diaries, and then read them... **Rich text formatting** is not included

And headings too

1. Well list consisting
2. organize
3. and unorganization
4. lots of unorganization

Even here's can be added

Now, there is still a lot to improve

7:21 PM (^▽^) Collapse

Time are too correctly showed, with timezones differences.

Normal B I U O A K E M S

Please, enter what you think currently

Send you can select mood

Problem

Users needed a centralized system for structured journaling with privacy controls, mood tracking, analytics, and administrative oversight.

Solution

Designed and implemented a full Blazor SaaS-style system:

- Authentication via ASP.NET Identity
- Rich text editor with format persistence
- Mood tracking integrated into entry metadata
- PDF export for entries and groups
- PIN protection for sensitive areas
- Admin panel with:
 - User management
 - Group controls
 - Moderation actions
- Statistics dashboard with charts, aggregates, and time-based trends
- Modular architecture prepared for scaling (services, repositories)

The screenshot displays four main views of the application:

- Login Screen:** Shows a form with "Username: Drugoi" and "Password:". It includes links for "Forgot your password?" and "Create account".
- Admin Dashboard:** A sidebar menu with "Home", "Manage account", "Logout", "Statistics", "Admin", "Report issue", "Developing", "PIN Start", "Other are the...", and "Create new diary group".
- Group Management:** A modal titled "Check pin for group" with a text input field containing "....." and a "by" button.
- User Management:** A table titled "Manage Users" with columns "Id", "User name", "Activated", "Role", and "Description". It lists four users: "MybrotherJoinMe" (activated, Admin, Blank), "IncredibleAccountfo" (activated, Default, 123), "TESTINGUNWORK" (not activated, Default, 123123123 is password), and "Admin" (activated, Admin, Blank). Each row has a "Actions with user" section containing buttons for "Deactivate account", "Reset password", "Delete user", "Deactivate account", "Reset password", "Give admin rights", "Delete user", "Activate account", "Reset password", "Give admin rights", and "Delete user".

Technical Notes

- C# 12 / .NET 9
- EF Core with migrations
- Clean separation between UI, Domain, and Infrastructure
- Custom middleware for activity logging
- Responsive UI via Bootstrap + custom components
- CI automation for testing and deployment

Outcome

A maintainable and extensible web application demonstrating full lifecycle engineering: UX, backend logic, data modeling, admin workflows, and analytics.

Links: [Github Repo](#), [Live demo](#), [Portfolio page](#)