

State of the Art Webapps mit Blazor Gamification Content selber Hosten

Johannes Baischer

Paris Lodron Universität Salzburg

17. Juni 2025, Salzburg

Inhalt

- 1 Einleitung
- 2 Blazor Framework
- 3 Server Side
- 4 Hosting und Infrastruktur
- 5 Client Side
- 6 Cross Platform Development
- 7 Gamification mit H5P

Einleitung

- Ziel: Interaktive H5P-Lerninhalte
- Fokus: Gamification, eigene Hosting-Lösung, plattformübergreifend
- Neues Framework mit C# statt JavaScript
- Full Stack Development richtig verstehen

Projektüberblick

Einleitung

Blazor Framework

Server Side

Hosting und Infrastruktur

Client Side

Cross Platform Development

Gamification mit H5P

Literatur

- Entwicklung einer Webanwendung mit Microsoft Blazor WebAssembly
- Rollenbasierte Authentifizierung, sichere Nutzerdatenverwaltung
- PostgreSQL-Datenbank, Hosting auf eigenem Linux-Server
- Integration von H5P-Inhalten für gamifizierte Lernmodule
- Cross-Plattform: Browser, Windows-App (.NET MAUI), Android, iOS

Blazor Framework

Was ist Blazor?

- Microsofts Antwort auf JavaScript-basierte Webframeworks
- Entwicklung von Webanwendungen in C#
- Razor-Syntax: HTML und C# in einer Datei
- Nutzen der leistungsstarken .NET-Plattform
- Zwei Varianten: Blazor Server und Blazor WebAssembly

Blazor Server vs. WebAssembly

Blazor Server	Blazor WebAssembly
Herkömmliche Web-App Logik läuft am Server SignalR für UI-Updates Geringe Client-Anforderungen	Basiert auf speziellem Bytecode Logik läuft im Browser WebAssembly-Standard Schnellere Reaktionszeit, weniger Serverlast

Einleitung

Blazor
Framework

Server Side

Hosting und
Infrastruktur

Client Side

Cross
Platform
Development

Gamification
mit H5P

Literatur

- **Blazor WebAssembly**
- Als Server reicht Mini-PC
- Möglichkeit für Offline-Nutzung
- .NET MAUI für plattformübergreifende, native Apps

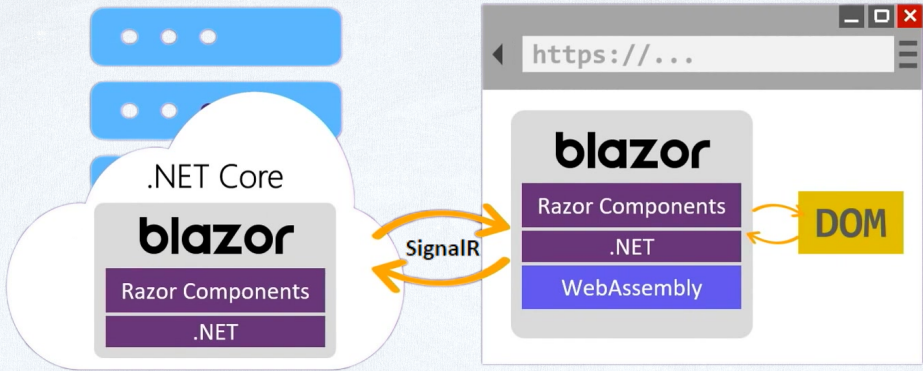


Abbildung: SignalR Architektur [8]

Projekt erstellen

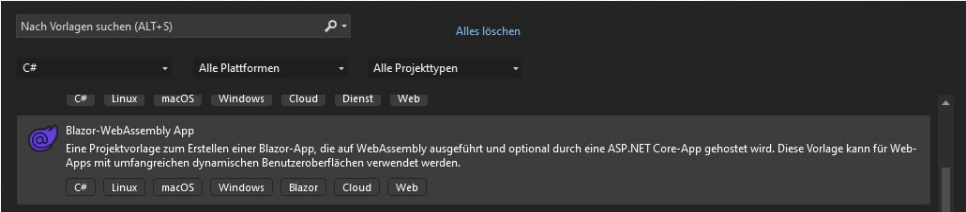


Abbildung: Erstellen der WebAssembly-App

Projekt erstellen

Blazor-WebAssembly App C# Linux macOS Windows Blazor Cloud Web

Framework ⓘ

.NET 7.0 (Keine Unterstützung) ▼

Authentifizierungstyp ⓘ

Keine ▼

☒ Für HTTPS konfigurieren ⓘ

☒ ASP.NET Core gehostet ⓘ

☒ Progressive Webanwendung ⓘ

☐ Keine Anweisungender obersten Ebene verwenden ⓘ

Abbildung: Konfiguration der Hosting-Methode

Projektvorlage

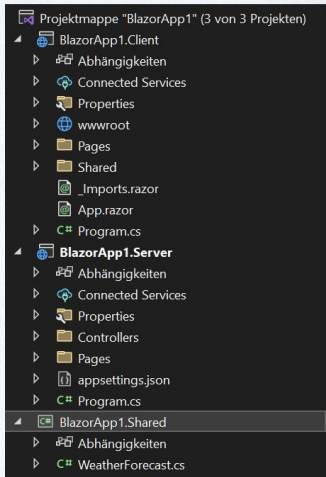


Abbildung: Blazor WebAssembly Projektvorlage

Server Side

- Initialisierung und Konfiguration (Program.cs)
- Authentifizierung und Autorisierung (ASP.NET Identity)
- API-Schnittstellen für Datenbankzugriffe
- E-Mail-Verifizierung und Token-Verschlüsselung

SignalR und API

- SignalR für Echtzeit-Kommunikation (Blazor Server)
- WebAssembly ist durch Browser limitiert
- WebAssembly nutzt klassische API-Schnittstellen
- Typensichere Controller mit ASP.NET Core

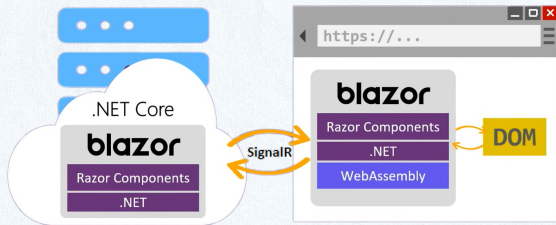


Abbildung: SignalR Architektur [2]

- Jeweils für jedes Projekt (Server, Client, MAUI) ein Program.cs
- Services registrieren (z.B. Authentifizierung, Datenbank)
- Middleware für Routing und API-Endpoints
- Beispiel: Registrierung des Identity Frameworks

Authentifizierung und Rollen

- ASP.NET Identity Framework
- Basierend auf „Claims“
- Rollen: System, Student, Teacher
- Eigene Policies (z.B. E-Mail-Verifizierung)
- Claims in (JWT)-Token speichern für API-Anfragen

- Autorisierung über Policies
- Beispiel: `[Authorize(Policy = „EmailVerified“)]`
- Rollenbasierte Autorisierung
- Beispiel: `[Authorize(Roles = „Teacher“)]`
- Zugriff auf API und UI steuern

Datenbankzugriff

Einleitung

Blazor
Framework

Server Side

Hosting und
Infrastruktur

Client Side

Cross
Platform
Development

Gamification
mit H5P

Literatur

- PostgreSQL mit Entity Framework Core
- Typensicherer Zugriff auf Daten
- BlazorWasm darf keine direkte Datenbankverbindung haben

Weitere Server Aufgaben

- E-Mail-Verifizierung und Passwort-Reset
- AES-Verschlüsselung (logisch, Schlüssel wäre sonst im Client)

Hosting und Infrastruktur

Hosting-Setup

- Linux-Server (z.B. Xubuntu auf HP-Elitedesk)
- Deployment mit .NET Core



Abbildung: HP Elitedesk Mini-PC [1]

Hosting-Setup

Einleitung

Blazor
Framework

Server Side

Hosting und
Infrastruktur

Client Side

Cross
Platform
Development

Gamification
mit H5P

Literatur

- systemd-Service für automatischen Neustart
- Reverse Proxy mit Apache2
- Domain und **HTTPS mit DuckDNS und Let's Encrypt**
- PostgreSQL-Datenbank auf demselben Server

Domain Setup

- DuckDNS für dynamische DNS
- Let's Encrypt für kostenlose SSL-Zertifikate
- Automatisches Zertifikat-Management mit Certbot
- **edubai.duckdns.org** *EDU + BAI* nicht *e-Dubai*

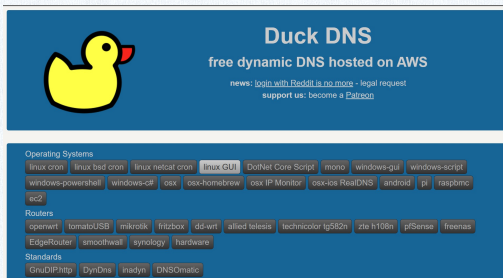


Abbildung: DuckDNS Website

PostgreSQL Datenbank

- Relationale Datenbank mit objektorientierten Erweiterungen
- Verwaltung mit pgAdmin
- Entity Framework für typensicheren Zugriff
- Beispiel-Tabelle: `user_credentials`

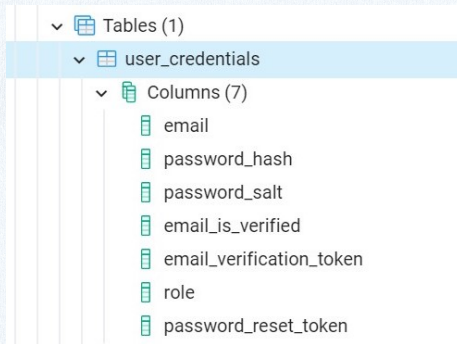


Abbildung: pgAdmin Menü

```
1 public partial class UserCredential
2 {
3     public string Email { get; set; } = null!;
4     public string PasswordHash { get; set; } = null!;
5     public string PasswordSalt { get; set; } = null!;
6     /// <summary>
7     /// true if email verification process was succesful
8     /// </summary>
9     public bool EmailIsVerified { get; set; }
10    /// <summary>
11    /// token to be sent via email in case of verification or password reset
12    /// </summary>
13    public string? EmailVerificationToken { get; set; }
14    public string Role { get; set; } = null!;
15    /// <summary>
16    /// temporary token for password reset request
17    /// </summary>
18    public string? PasswordResetToken { get; set; }
19 }
```

Client Side

Blazor WebAssembly Client

- Ausführung im Browser via WebAssembly
- Razor-Komponenten: HTML + C#
- MVC/MVVM-Pattern für Strukturierung
- Dependency Injection für Dienste (= Laufzeitauflösung von Abhängigkeiten)

WebAssembly

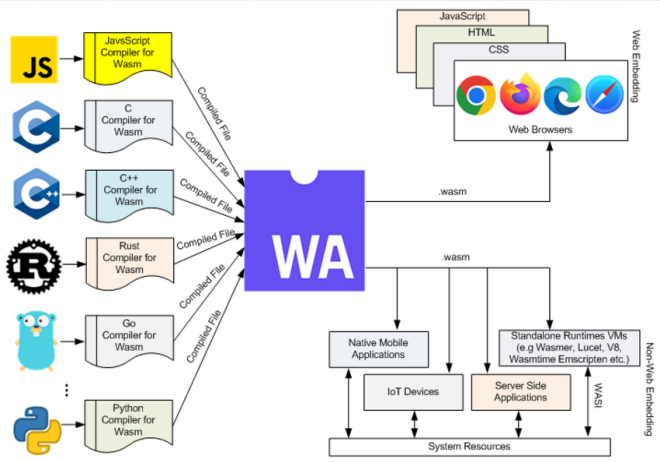


Abbildung: WebAssembly Flowchart [7]

Progressive Web App (PWA)

- Offline-Funktionalität
- Installation auf dem Gerät möglich

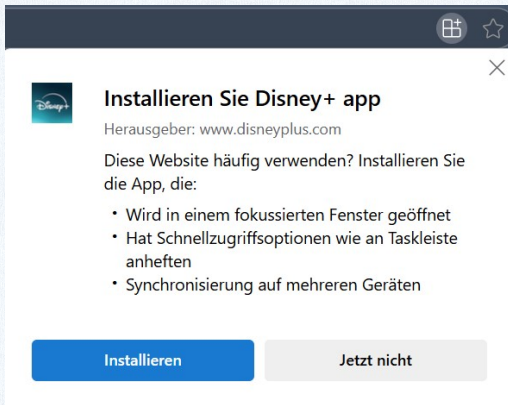


Abbildung: WebAssembly PWA Installation

Blazor-Syntax

```
1 <AuthorizeView>
2   <Authorized>
3     @if (LearningApps == null)
4     {
5       <p>@Localizer[Localization.loading]</p>
6     }
7     else
8     {
9       <div class="learningapps__all__apps">
10        @foreach (LearningAppData la in LearningApps!)
11        {
12          <LearningAppCard Data=la NavigationManager=NavigationManager />
13        }
14      </div>
15    }
16  </Authorized>
17 </AuthorizeView>
```

JS Interop und File Management

- Für alle Funktionalitäten, die nicht direkt in Blazor verfügbar sind
- **Support für alle großen JS-Bibliotheken!**
- In unserem Fall: H5P-Integration

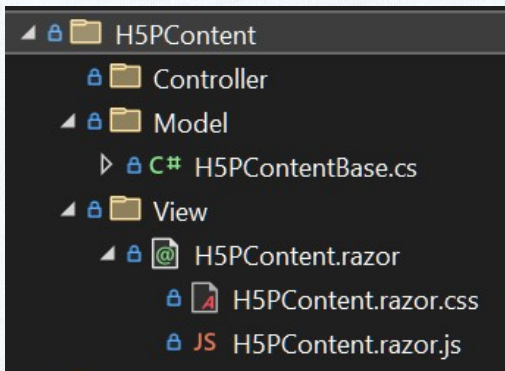


Abbildung: H5P Modul Struktur

Mehrsprachigkeit und UI

- Ressourcen-Dateien für Lokalisierung
- Responsive Design mit CSS Media Queries
- Benutzeroberfläche: Login, Registrierung, Lernapps

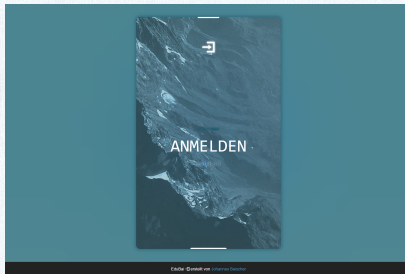


Abbildung: Login-Seite

.resx-Dateien

State of the
Art Webapps
mit Blazor
Gamification
Content
selber
Hosten

Johannes
Baischer

Einleitung

Blazor
Framework

Server Side

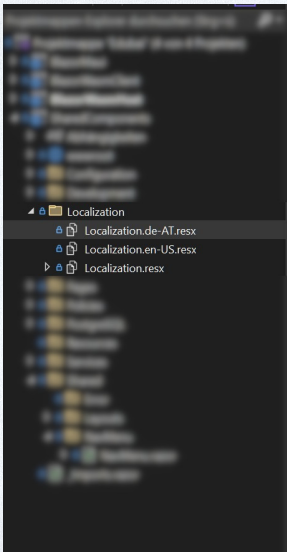
Hosting und
Infrastruktur

Client Side

Cross
Platform
Development

Gamification
mit H5P

Literatur



Name	Wert
confirm	Bestätigen
create_account	Account Erstellen
created_by	erstellt von
EduBai	EduBai
email	Email
featured	Empfohlen
forgot_password	Passwort vergessen
forgot_password_text	Fordere einen Link zum Zurücksetzen deines Passworts an deine
give_feedback_here	Gib hier Feedback!
have_you_tried	Schon probiert?
Johannes_Baischer	Johannes Baischer
learning_apps	Learning Apps
loading	Wird geladen...
login	Login
password	Passwort
password_reset_email_sent	Dein Link zum Zurücksetzen deines Passworts wurde an deine
please_verify_your_email	Bitte verifiziere deine Email-Adresse!
please_verify_your_email_text	Um alles sicher zu halten und sicherzustellen, dass dein Account
reset	Zurücksetzen
reset_password_text	Lege unterhalb dein neues Passwort fest.
sign_in	Anmelden
sign_up	Registrieren
student	Schüler:in
teacher	Lehrer:in
welcome	Willkommen
you_are_not_logged_in	Du bist nicht angemeldet!
*	

Cross Platform Development

Razor Class Library (RCL)

- Gemeinsame Codebasis für Web, Server und native Apps
- Wiederverwendbare Komponenten und Logik
- Projektreferenzierung in BlazorWasmHost, BlazorWasmClient, BlazorMaui

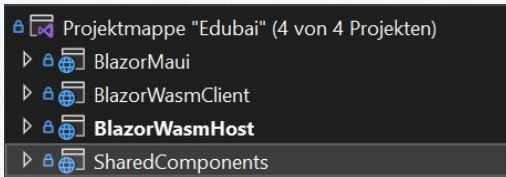


Abbildung: Projektmappe mit RCL

Blazor MAUI

- .NET MAUI Blazor Hybrid für native Apps (Windows, Android, iOS)
- WebView2 für Ausführung der Blazor-App
- Zugriff auf native APIs (z.B. LocalStorage) möglich -> kein Server nötig

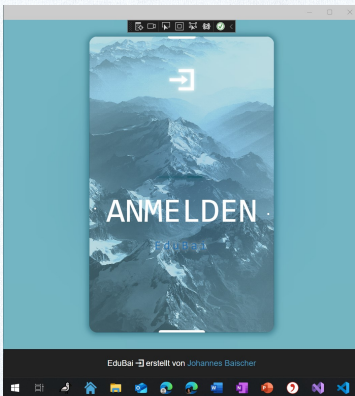


Abbildung: EduBai Windows App

Plattformabhängiger Code

- Unterschiedliche Implementierungen für Web und native Apps
- Beispiel: Datenbankzugriff, Local Storage
- Plattform-Info-Service via Dependency Injection

Gamification mit H5P

Gamification-Konzept

- Motivation und Engagement durch spielerische Elemente
- H5P: Open-Source-Standard für interaktive Lerninhalte
- Integration in EduBai für selbstverwaltete Lernmodule

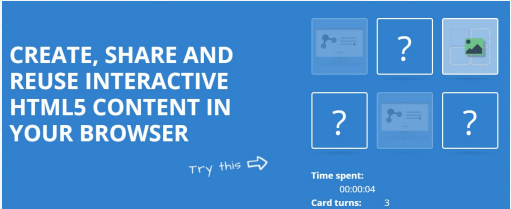


Abbildung: H5P Website Beispiel

H5P-Content in EduBai

- Erstellung mit Lumi H5P Editor
- Bereitstellung mit h5p-standalone JavaScript-Bibliothek
- Einbindung per JS-Interop in Blazor-Komponenten

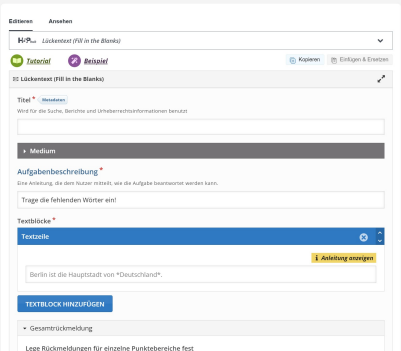


Abbildung: Lumi Editor



Abbildung: EduBai Lernapps

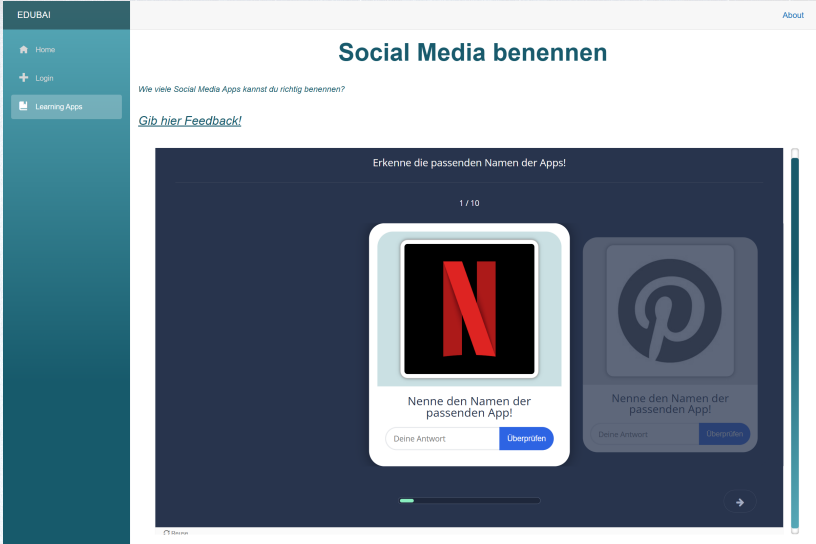


Abbildung: Beispiel App

[5], [4], [3], [6]

Inc. HP. *HP EliteDesk 800 G6 Desktop Mini PC*. <https://support.hp.com/at-de/product/details/hp-elitedesk-800-g5-desktop-mini-pc/27066639>. Accessed: 2025-06-12. 2025.

Inc Microsoft. *ASP.NET Core Blazor*. https://learn.microsoft.com/de-de/aspnet/core/blazor/index/_static/blazor-server.png?view=aspnetcore-9.0. Accessed: 2025-06-09. 2024.

Inc Microsoft. *ASP.NET Core Blazor*. https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website. Accessed: 2025-06-07. 2025.

Inc Microsoft. *ASP.NET documentation*. https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website. Accessed: 2025-06-07. 2025.

Inc Microsoft. *Build your first web app with ASP.NET Core using Blazor*. <https://dotnet.microsoft.com/en-us/learn/aspnet/blazor-tutorial/intro>. Accessed: 2025-06-07. 2025.

Inc Microsoft. *Introduction to Razor Pages in ASP.NET Core*. https://learn.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website&tabs=visual-studio. Accessed: 2025-06-07. 2025.

Partha Ray. *Webassembly Data Flow Architecture*. https://www.researchgate.net/figure/WebAssembly-data-flow-architecture_fig2_373229823. Accessed: 2025-06-15. 2023.

SergeyZ. *Blazor: Server and WebAssembly at the same time in one application*.

Vielen Dank für Ihre Aufmerksamkeit!

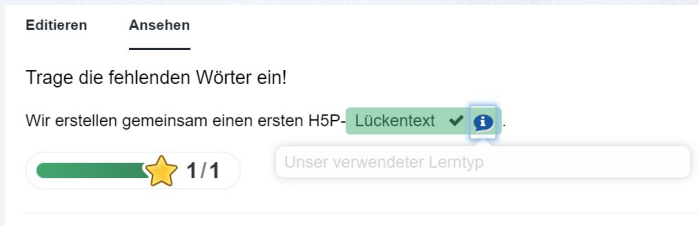


Abbildung: Lumi Viewer

<https://edubai.duckdns.org/>

<https://github.com/Johannes-Baischer/State-of-the-Art-Webapps-mit-Blazor>