WebShop24

Luca Riebel, Stefan Schäfer, Oskar Bartsch,
Sebastian Albert, Philip Sagawe

Projekt Vision

„Wir verbinden Menschen, um lokale und unkomplizierte Lösungen zu fördern.“

Projekt Ziele

Einfachheit

Eine benutzerfreundliche Plattform für Käufer und Verkäufer.

Gemeinschaft

Stärkung lokaler Netzwerke durch transparente und vertrauenswürdige Transaktionen.

Innovation

Nutzung moderner Technologien für eine bessere Nutzererfahrung.

Ebay nur Cooler

Architektur Entscheidungen

Architekturstil

Service Based Architektur

Monolithisches Backend mit logischer Partitionierung

Entwurfsmuster

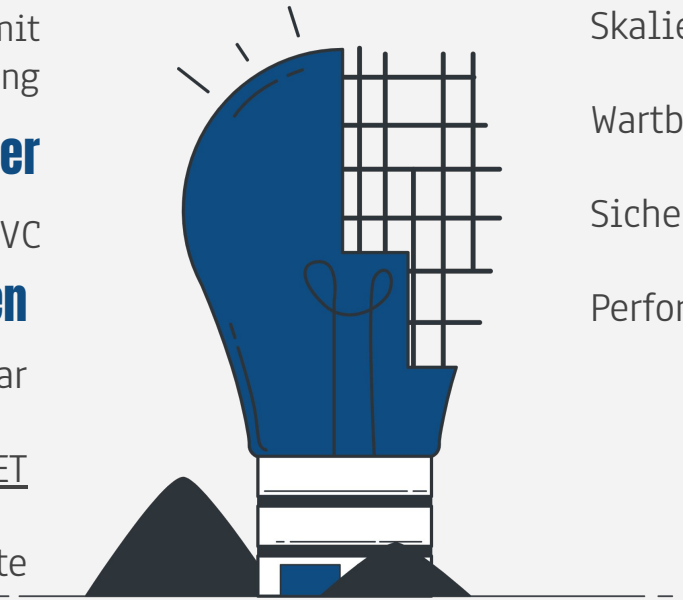
MVC

Technologien

Angular

ASP.NET

SQLite



Begründung

Skalierbarkeit

Wartbarkeit

Sicherheit

Performance

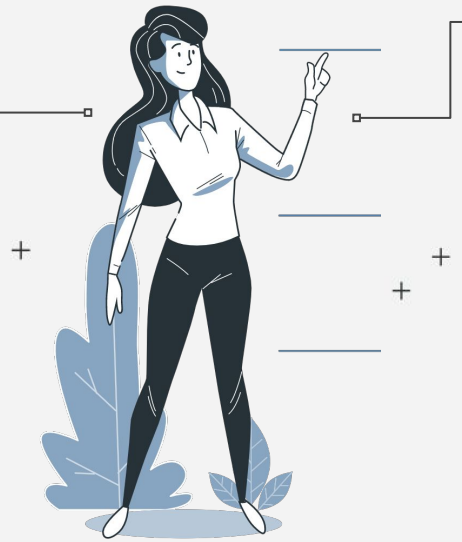
Design Patterns

Die hier gezeigten Design Patterns orientieren sich an den SOLID-Prinzipien für sauberen und wartbaren Code.

S - Single Responsibility

Controller (ASP.NET)
→ Jede Klasse (ArticleController, AuthController) hat genau eine Aufgabe: Verarbeitung einer bestimmten Art von HTTP-Anfragen.

Angular Services
→ ArticleService, AuthService– jede kümmert sich nur um eine Sache.



D - Dependency Inversion

ASP.NET Dependency Injection
→ Controller hängt nicht von konkretem AuctionService, sondern von IAuctionService ab.

Angular DI
→ Services werden über das Dependency-Injection-System injiziert, nicht fest im Code erzeugt.

Technologie Umsetzung

Technologie: Angular
Funktion: Dynamische Benutzeroberfläche, unterstützt Echtzeit-Updates.

Frontend

Backend

Technologie: ASP.NET
Funktion: Serverseitige Geschäftslogik mit klaren APIs.

Technologie: SQLite
Erweiterung: Caching mit Redis für Performance-Optimierung.

Datenbank

Entwurfsmuster (MVC)

Model (SQLite)

Beinhaltet die Daten der Anwendung.

View (Angular)

Verantwortlich für die Darstellung der Daten und die Interaktion mit dem Benutzer.

Controller (ASP.NET)

Steuert die Logik, verarbeitet Benutzereingaben und gibt Anweisungen an Model und View.

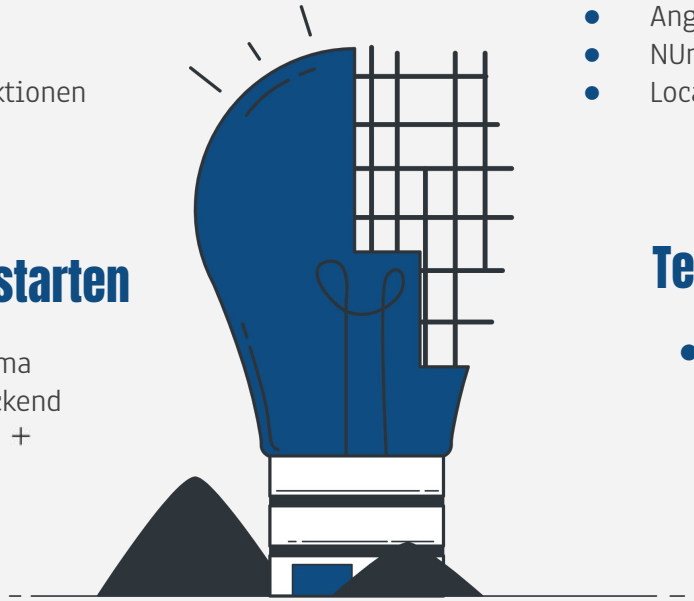
Qualitätssicherung

Test Ziele

- Frühe Fehlererkennung
- Absicherung zentraler Funktionen
- API Funktionalität

Testtools und Umgebung

- Angular CLI, Karma
- NUnit
- Locale Rechner



Testarten

- Unit Tests: Jasmine/Karma frontend und NUinit Backend
- API Tests: Swagger

Testabdeckung

- 80% - 85%

CI/CD Setup

- GitHub Actions
- 2 Jobs
  - Build and Test
  - Deploy
- Bei Pushes und Pull Requests auf “main”

Management Herausforderungen

Technische Herausforderungen

Probleme bei der Nutzung von Tools oder Software für das Projektmanagement.

Unzureichende technische Fähigkeiten einzelner Mitglieder.

Feedback und Iteration

Fehlendes oder ineffektives Feedback zwischen den Teammitgliedern.

Probleme beim Umgang mit Kritik oder Verbesserungsvorschlägen.

Kommunikationsprobleme

Unklare Kommunikation von Aufgaben, Zielen oder Erwartungen.

Unterschiedliche Kommunikationsstile oder mangelnde Erreichbarkeit.

Missverständnisse über Deadlines oder Prioritäten.

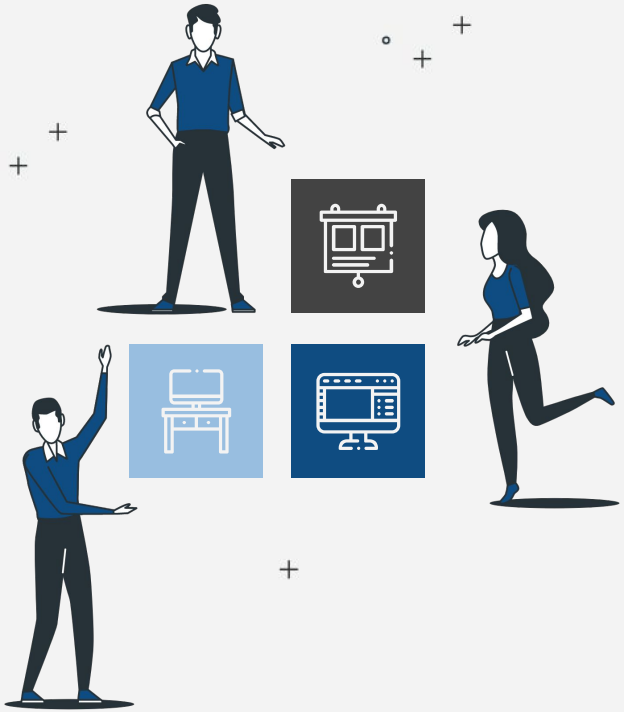
Management Strategien

Klare Planung

Mit dem Projekt -management Tool Jira

Klare Kommunikation

Offene Diskussionen fördern und Kommunikationskanäle festlegen



Regelmäßige Meetings

Kurze, regelmäßige Updates einplanen

Moderation

Einen Teamleiter, der Konflikte löst und die Aufgabenverteilung überwacht.

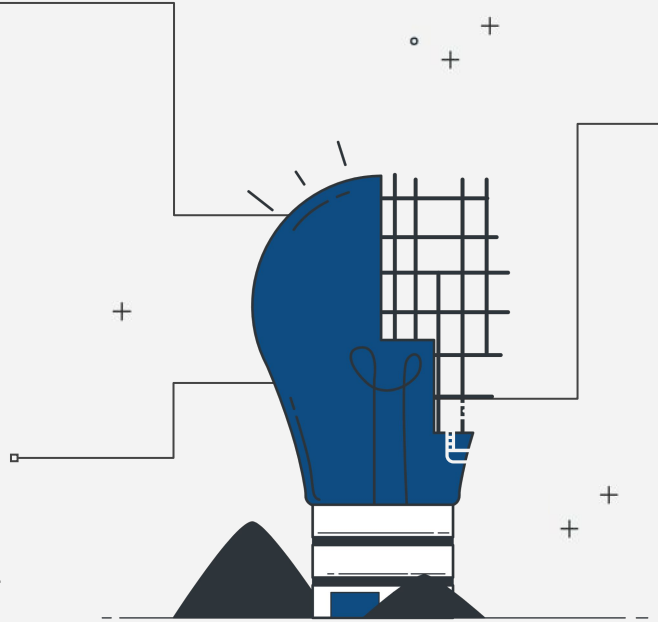
Learnings

Communication is key

Regelmäßige Meeting

Respektvoller Umgang

Lieber einmal zuviel Fragen



Design von Anfang beachten

Design wurde während der entwicklung etwas vernachlässigt

Dies hat für mehr Aufwand im Nachgang gesorgt

Alle Teammitglieder müssen im Boot sein

“abgehängte” Teammitglieder wirken sich negativ auf Teamdynamik und Produktivität aus

Live Demo