

# Structurio

---

**Structurio** ist eine modulare Projektmanagement-Anwendung, entwickelt mit .NET (WPF) und Python. Sie ermöglicht eine umfassende Verwaltung von Projekten mit Funktionen zur Aufgabenplanung (Kanban), Zeit- und Kostenverfolgung, Dateioorganisation sowie Projektkonfiguration – inklusive PDF-Vorschau und grafischer Auswertungen.

## Features

- **Login- und Registrierungssystem**
- **Projektverwaltung:** Projekte erstellen, bearbeiten, löschen
- **Kanban-Board** mit Drag-and-Drop und Inline-Bearbeitung
- **Zeiterfassung** mit Check-In/Out und manueller Bearbeitung
- **Kostenverwaltung** inkl. PDF-Rechnungsupload und Kreisdiagramm
- **Dateiverwaltung** mit PDF-Vorschau und Suchfunktion
- **Projekteinstellungen** mit Farbauswahl
- **Diagrammübersicht** für Ausgaben
- **Cloud-Datenbank (Neon-Tech)**
- **Swagger-Dokumentation & REST-API**
- **Unit-Tests** im Frontend & Backend
- **Logging** aller Benutzeraktionen (Serilog)

## Architektur

- **Frontend:** WPF (C#, .NET 8), MVVM-nah
- **Backend:** Python (Flask + Connexion), RESTful API
- **Datenbank:** PostgreSQL via Neon
- **PDF-Viewer:** PdfiumViewer
- **Diagramme:** LiveChartsCore (Kreisdiagramm)
- **Logging:** Serilog (Konsole & Datei)
- **Dokumentation:** Swagger UI & Doxygen

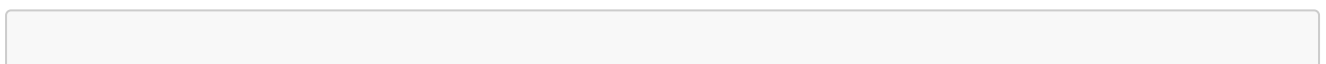
## Setup

### Voraussetzungen

- [.NET 8 SDK](#)
- [Python 3.11](#)
- Visual Studio (für WPF)
- Visual Studio Code (für Backend)

### Installation

#### 1. Frontend



```
dotnet add package LiveChartsCore.SkiaSharpView.WPF --version 2.0.0-rc5.4
```

## 2. Backend

```
pip install -r requirements.txt
python start_server.py
# Server läuft unter http://localhost:8080
```

## Testen

- Unit-Tests im Frontend (MSTest)
- Backend-Tests über Swagger / Unit-Tests

## Rollen & Rechte

- **Projektleiter:** Projektleiter kann mehr sachen im Projekt machen als ein Normaler User.
- **User:** Alle User können in einem Projet arbeiten.

## Dokumentation

- [Betriebsanleitung \(User Guide\)](#)
- [Lastenheft \(Anforderungen\)](#)
- [Pflichtenheft \(Technische Umsetzung\)](#)
- [Projekttagbuch \(Entwicklungsverlauf\)](#)
- [Wichtige Infos \(für Lehrpersonen\)](#)

## Entwickler

Name	Fokusbereiche
<b>Valentin</b>	Projekte, Kanban, Logging, UniTest, Doxygen, Dateisystem, Projektverwaltung, (Frontend/Backend)
<b>Maximilian</b>	Timestamp, Kostenverwaltung, PDF-Viewer, Logging, UniTest, Doxygen, Diagramme (Frontend/Backend)

## Lizenz

- Alle Icons, Emojis und Testdaten wurden mit ChatGPT generiert.
- Keine externen Bildquellen verwendet.
- Fachquellen: StackOverflow, offizielle Framework-Dokumentationen, ChatGPT und Gemini.

## Achtung

- Falls der Build nicht funktioniert dann einfach die relativen Pfade in der App.xaml.cs anpassen. Hinweise dazu sind bereits im Kommentarbereich dort enthalten.
-