

# Codeer.LowCode.Blazor



# 自己紹介

## ■ 氏名

- ・ 和田 采乃 (わだ あやの)

## ■ 趣味

- ・ 爬虫類飼育

## ■ 経歴

- ・ 2021-2024 IT企業 コンサルタント
- ・ 2024-現在 Codeer エンジニア

## ■ 担当業務

- ・ Codeer.LowCode.Blazorに関する業務  
開発、製品強化、展示会 など



# Codeer.LowCode.Blazor紹介

高品質なシステムを  
・**安く**  
・**早く**  
作成することができます。

しかも、完成したシステムには  
・**ローコード機能**  
・**AI機能**  
がついています。

# ローコードアプリを ~~使いませんか？~~ 作りませんか？

ライブラリで  
組み込み

※ライブラリでのローコード組み込みは他にはありません  
特許出願中 (特願2025-006741)

ローコード



Codeer.LowCode.Blazor

生成AI



業界  
ロジック

御社独自の  
ノウハウ

開発技術  
.NET

資産

Codeer.LowCode.Blazorを使えば  
あっという間に『**御社オリジナルのローコードアプリ**』を作成できます！

特許出願中 (特願2025-006741)

# ローコードアプリを 使いませんか？ 作りませんか？

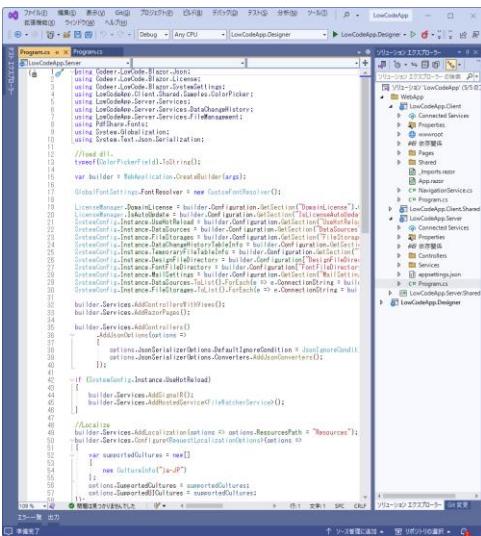


## 実行エンジン型の ローコードエンジン ライブラリ

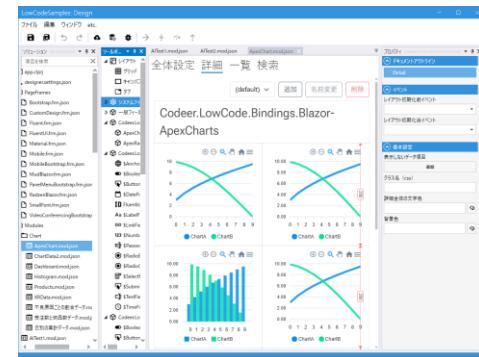
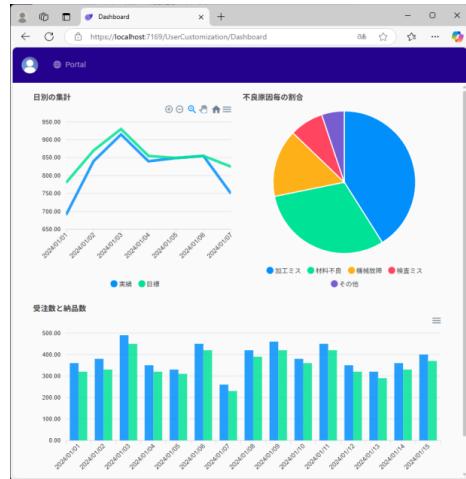


# OSS

OSSのライブラリとも  
自由に組み合わせることができる

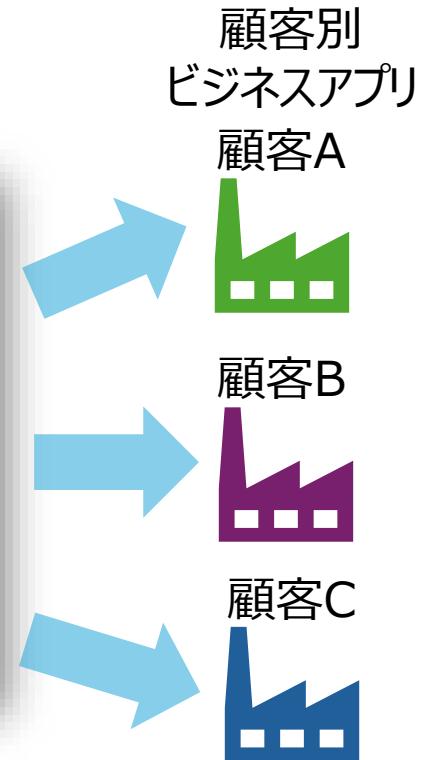
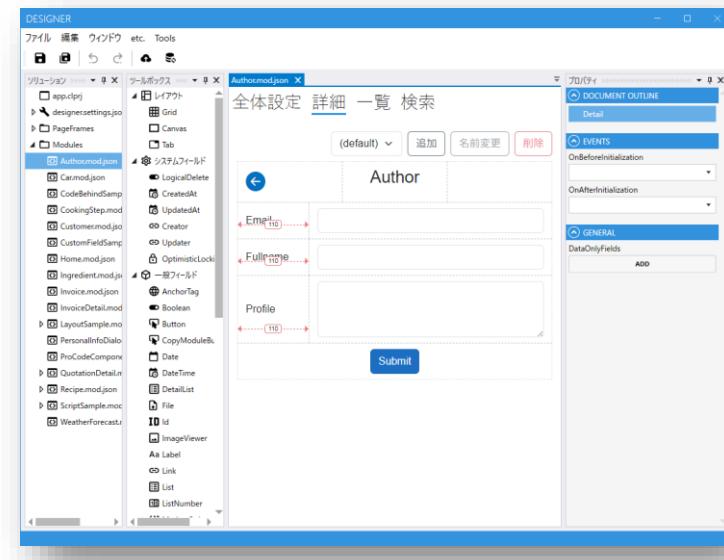
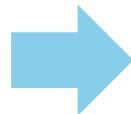
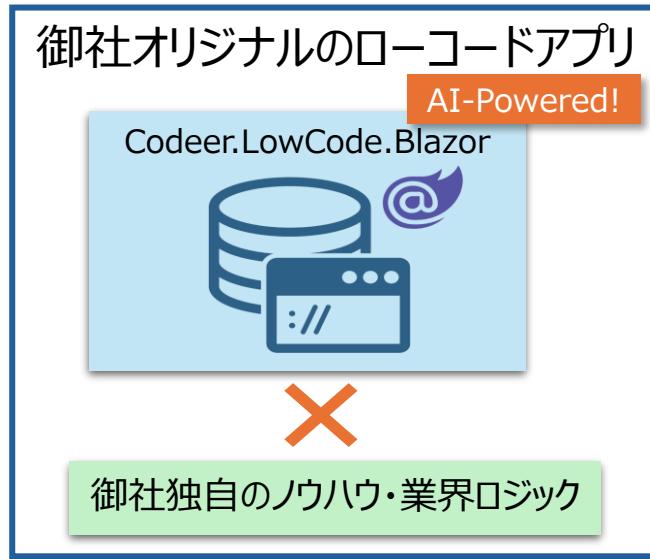


フルスクラッチがベースなので無制限にカスタマイズ



各分野に特化したオリジナルの  
ローコードアプリを作成！

# 御社のアプリにローコード機能とAI機能の組み込みが可能です！



ローコード機能とAI機能を組み込むことで  
高品質で高速な開発が可能

ベースとなるアプリから  
顧客別にカスタマイズして展開することが可能

## エクス様で新製品開発中



株式会社エクス様は、1994年の創業以来、生産管理システム「Factory-ONE 電脳工場」シリーズの開発を中心に、製造業向けの生産管理ソリューションを提供し、業界をリードしてきました。同社のソリューションは、多くの製造現場で生産体制の効率化を支えています。今回、さらに柔軟なシステムカスタマイズを可能にするため、「Codeer.LowCode.Blazor」を導入し、エンドユーザーによる内製化の推進と DX 加速を目指します。

# オンザリンクス様でパッケージへの組み込み

## 業界初！ローコードWMS



ONZALINX

集計



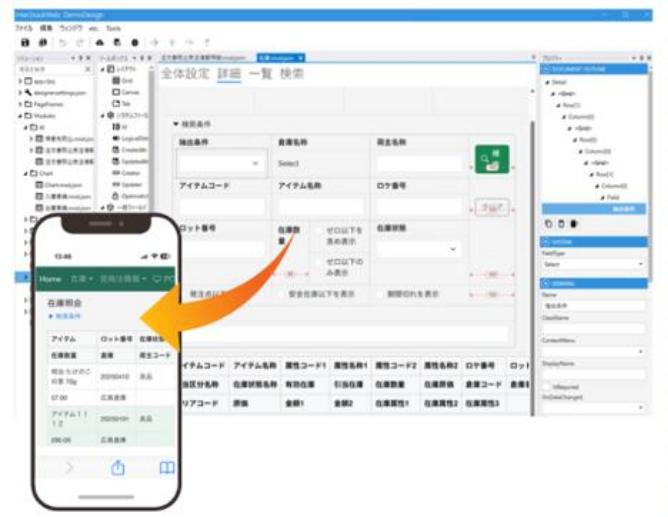
通常の開発手法ではローコード機能の設計/実装は数年規模・高額の投資を要する必要があります。しかし  
**Codeer.LowCode.Blazor** を利用すれば  
圧倒的に短い期間・低コストで実現できるのが非常に魅力的でした！

The screenshot shows a '納品書' (Delivery Note) screen with a green header. Below it, there are two smaller windows: '納品書 (専用)' (Delivery Note (Special)) and '納品書 (専用)' (Delivery Note (Special)). An orange arrow points from the top right of the main window towards the bottom right of the slide, indicating a mobile view.

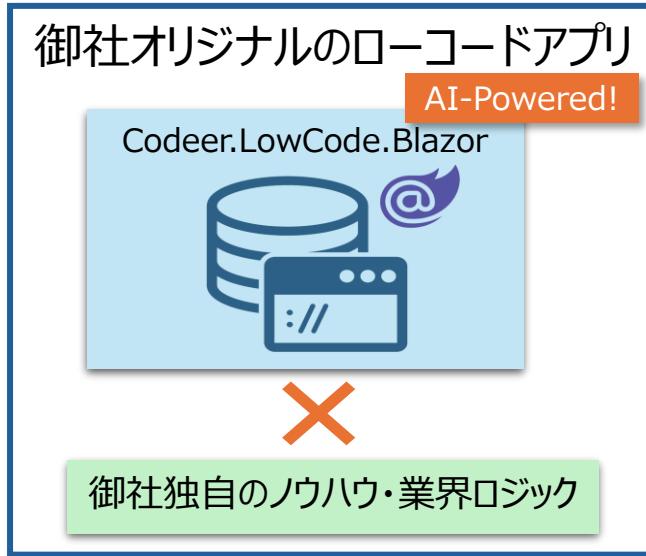
帳票



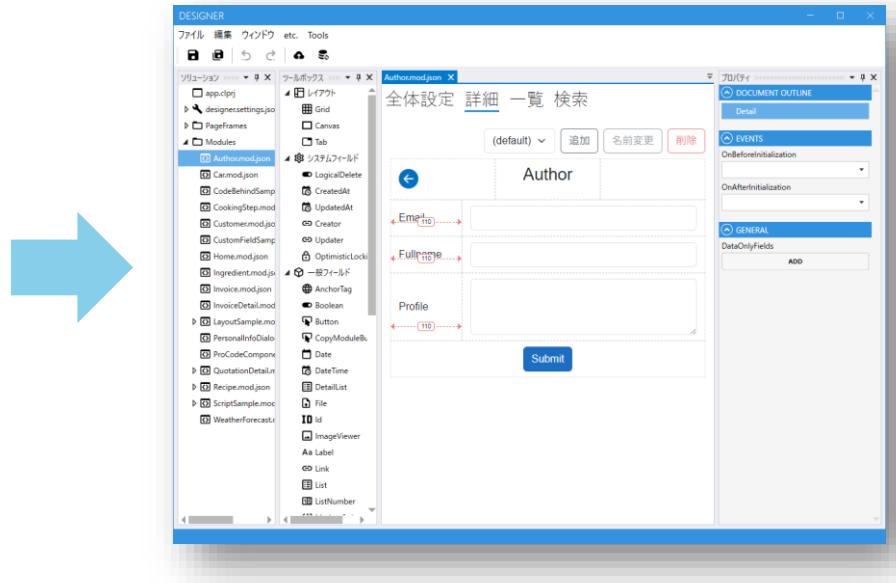
モバイル



# 低コストで高品質なアプリを開発できます！



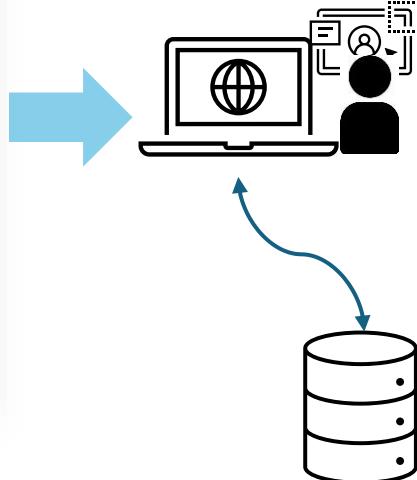
あつという間に開発完了



開発後はローコードアプリとして  
非技術職の方でもカスタマイズ可能  
メンテナンスも簡単

Blazorで開発フレームワークを最新化！

自社用  
ビジネスアプリ

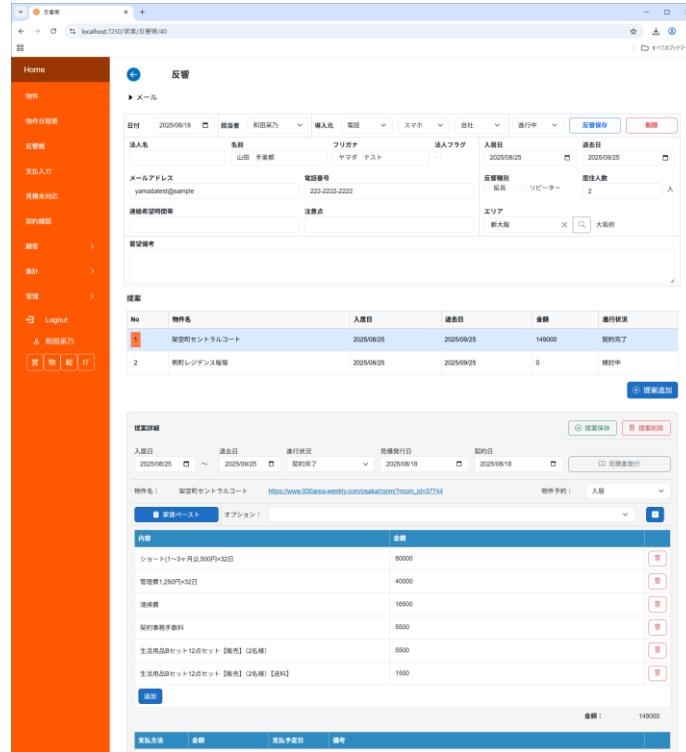


## BraTTo様 マンスリーマンション管理



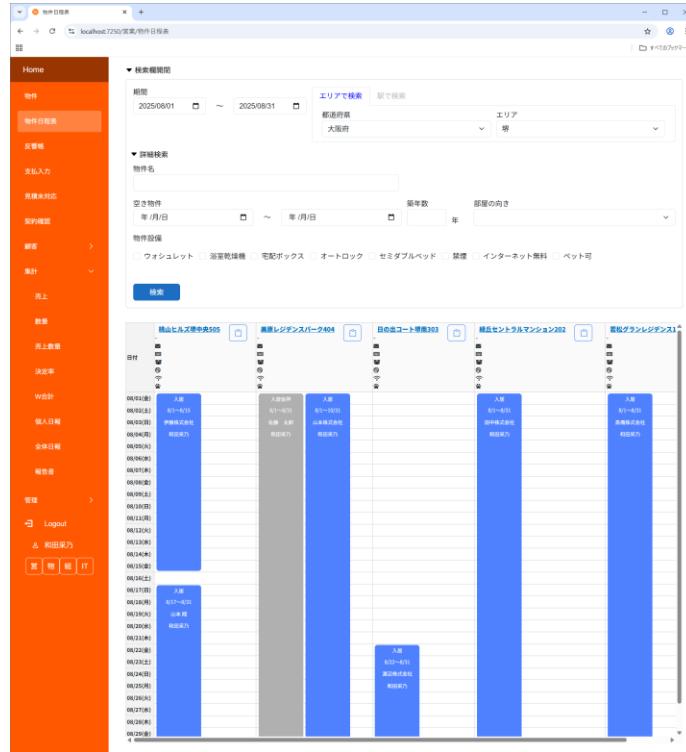
BraTTo様の案件はコンペ形式で、Codeer.LowCode.Blazor を活用したご提案が、  
**圧倒的なコストパフォーマンス**（低価格・短納期・高機能）により採用されました。  
2位につけていた企業がベトナムでのオフショア開発を提案していた中、弊社の提示価格は  
その半額以下。コスト競争力でも大きく差をつける結果となりました。  
また、通常のローコードツールと異なり、必要に応じてフルスクラッチでの開発も可能なた  
め、機能面でも一切妥協はありません。

## BraTTo様 マンスリーマンション管理



This screenshot shows the '反響' (Response) screen. It includes a search bar at the top and a main form area with fields for name, address, and contact information. Below the form is a table of responses with columns for No., 物件名 (Property Name), 入退日 (Check-in/Check-out Date), 調査日 (Investigation Date), 金額 (Amount), and 執行状況 (Execution Status). A detailed view of a specific response is shown below the table.

複雑な画面でもローコードで作成可能



This screenshot shows the '物件日報表' (Property Daily Report) screen. It features a large grid where each row represents a property and each column represents a date. The grid is color-coded by status, with blue representing active properties and grey representing inactive ones. The left sidebar contains navigation links for various reports and management functions.

特種なコンポーネントは  
カスタマイズで作成

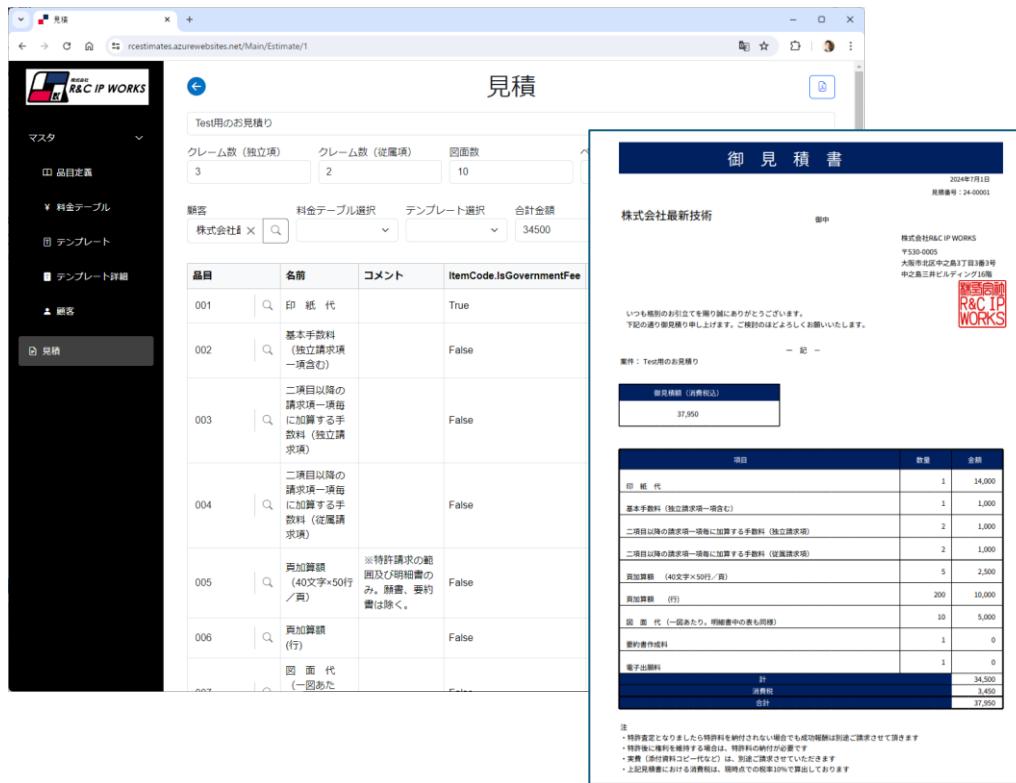


This screenshot shows the '物件日報' (Property Daily Report) screen. It displays a detailed summary table with various metrics and formulas. The table includes columns for '前月総賃料の売上(%)' (Previous month's total rental revenue ()), '新規入居者数' (New tenants), '賃料収取率' (Collection rate), and '合計決定期' (Total regular period). The bottom section of the table shows daily summaries for each day of the month.

SQLと組み合わせて様々な  
集計機能を作成

## 社内システム開発事例

# R&C IP WORKS様 特許見積システム



# 複雑な特許の見積書発行システム 要件定義も含め2日で完了

## コスト削減

項目	内容	コスト削減
高速なプロトタイピングを使った要件定義	仕様不具合による手戻り工数削減	20%削減
画面/機能作成	バックエンド処理まで含めた工数削減	50~80%削減
共通化による品質向上	不具合修正工数削減	50~80%削減
自動テスト	開発時の隠れたテスト工数削減 回帰テストの工数削減 作業の負荷分散	50~80%削減
メンテナンス	調整工数の削減	50~80%削減

ちなみに・・・

最近、弊社が受注したコンペ形式のSI案件において、  
**Codeer.LowCode.Blazor** を活用したご提案が、  
**圧倒的なコストパフォーマンス（低価格・短納期・高機能）**  
により採用されました。

2位につけていた企業がベトナムでのオフショア開発を提案  
していた中、

**弊社の提示価格はその半額以下。**コスト競争力でも大きく差  
をつける結果となりました。

また、通常のローコードツールと異なり、  
**必要に応じてフルスクラッチでの開発も可能なため、**  
**機能面でも一切妥協はありません。**

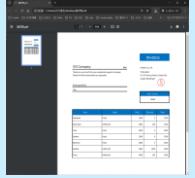


# Codeer.LowCode.Blazorの特徴



## 充実の機能

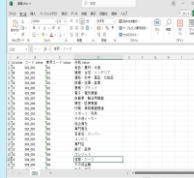
帳票出力



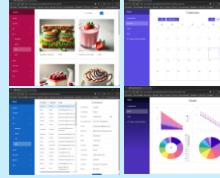
バーコードリーダー連携



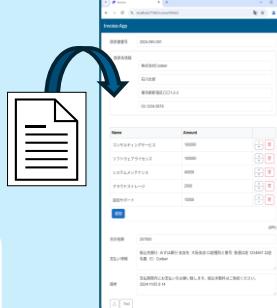
Excel連携



リッチなUI



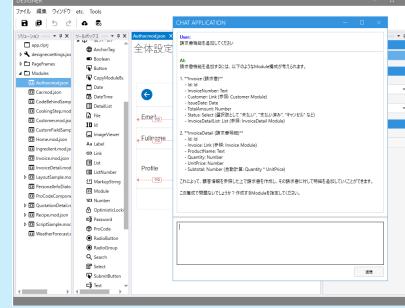
AIによるデータ取り込み



## 生成AI

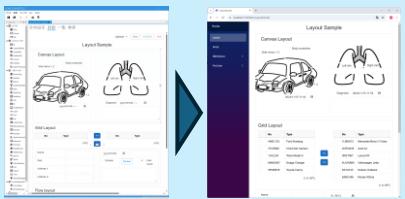


AIによる開発サポート

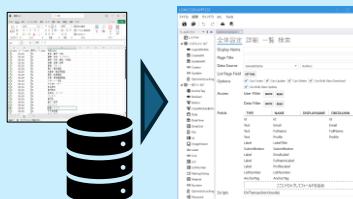


Codeer.LowCode.Blazor

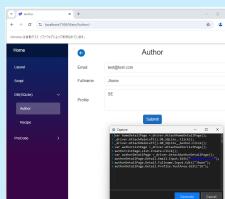
簡単に画面や機能を作成



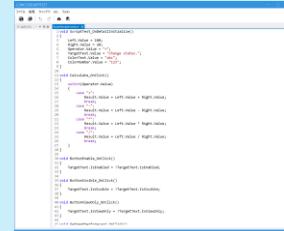
ExcelやDBの定義をもとに作成



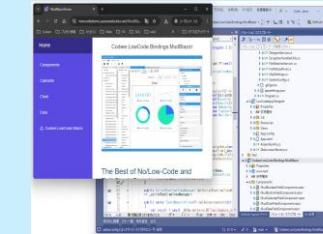
自動テスト



スクリプトで処理をカスタマイズ



プロコードとの自由な連携



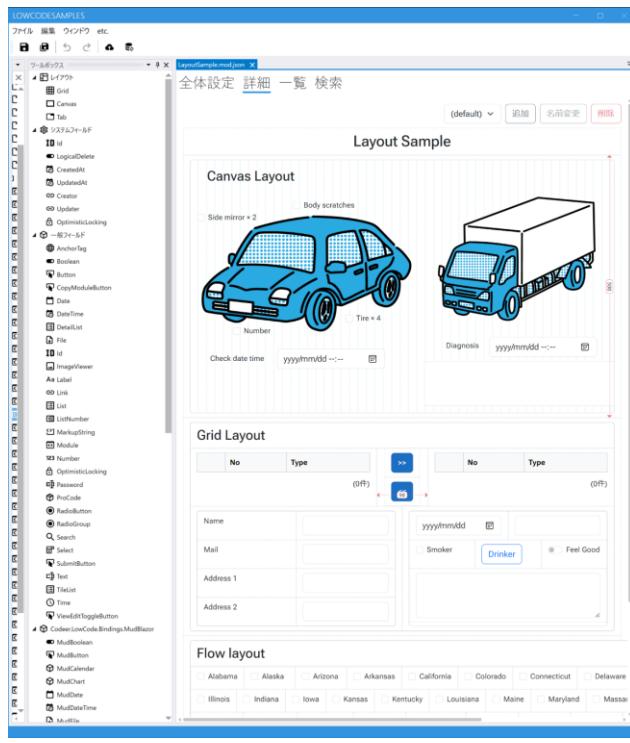
## 開発の効率化

## カスタマイズ



Codeer.LowCode.Blazor

# -レイアウト-



デザイン

This screenshot shows the Layout Sample website. It features three main sections:

- Canvas Layout:** Displays a car and a truck with labels for "Side mirror x 2", "Body scratches", "Number", and "Tire x 4". Below it is a date input field and a "Diagnosis" button.
- Grid Layout:** Two tables side-by-side. The first table has columns "No" and "Type" with rows for 4ABC123 (Ford Mustang), 7XYZ890 (Chevrolet Camaro), 1JKL234 (Tesla Model S), 5MN0567 (Dodge Charger), and 3PQR678 (Toyota Camry). The second table has columns "No" and "Type" with rows for 1LMN012 (Mercedes-Benz C-Class), 2OPQ345 (Audi A4), 3RS1567 (Lexus RX), 4UVW890 (Volkswagen Jetta), and 5XY2123 (Subaru Outback).
- Flow layout:** A horizontal list of US states.

サンプルサイトURL : [LayoutSample](https://lowcodemo.azurewebsites.net/Bootstrap/LayoutSample)

CanvasLayout、GridLayout、FlowLayoutの組み合  
わせで自由に画面を作成可能です。

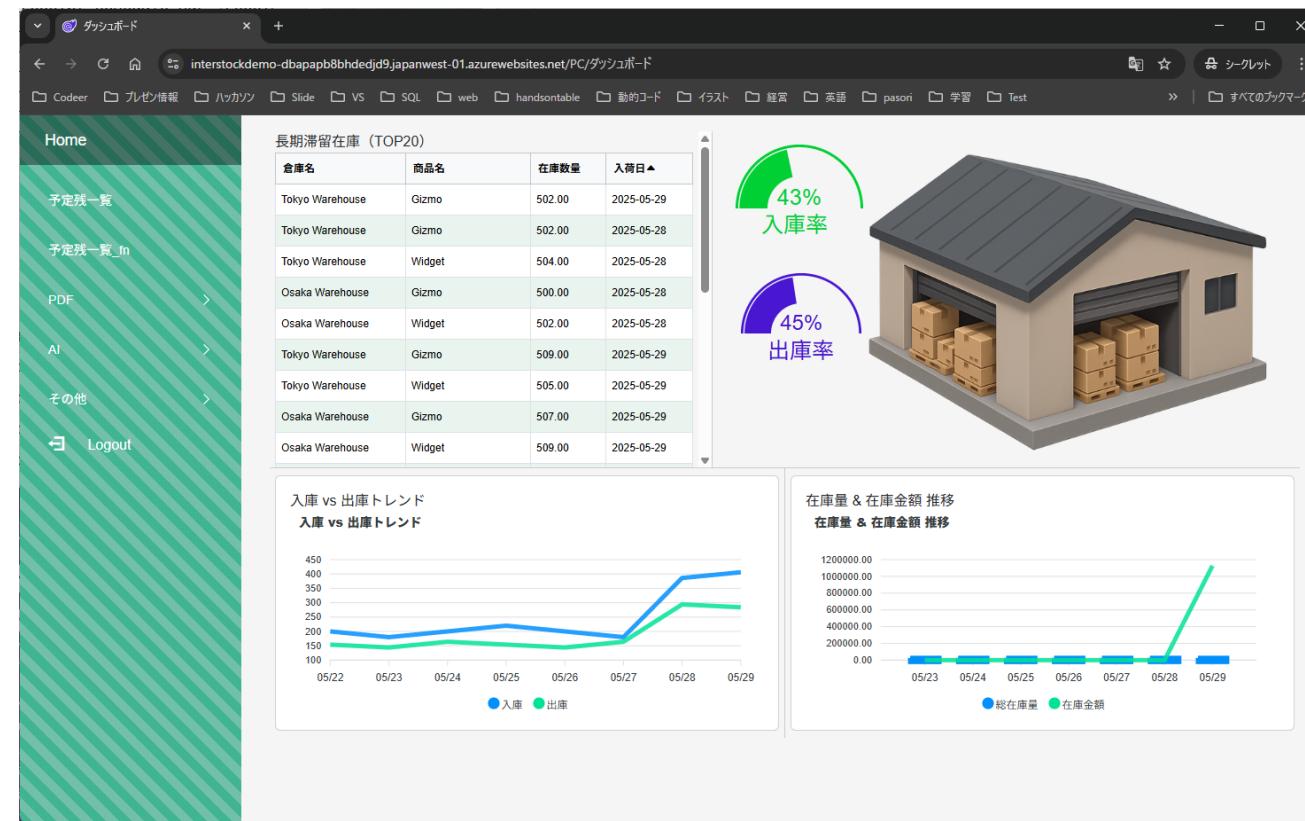
各UI部品の運動もノーコードや僅かなスクリプトで作成可  
能です。

サイドバー、ヘッダ、フッターの表示も設定可能です。

This screenshot shows a contact management application. It includes:

- Contact List:** A table with columns "FirstName", "LastName", "Email", "Address", "City", "State", "Country", "Phone", and "Notes". It lists contacts like Joseph Huber, Beth Miller, Linda Wilson, Steven Lewis, Jeffery Tucker, Mark Jones, etc.
- Food Grid:** A grid of images for "Vegetable Sandwich", "Strawberry Smoothie", "Raspberry Cheesecake", "Mocha", "Lemon Tart", and "Iced Tea".
- Form Fields:** A sidebar with various form fields for "Layout", "Controls", "DB", "List", "Multiflow", "Detail", "Title", "Split", "Script", "PreCode", "Multiblazor", "Radzen Blazor", "IgniteUI", "CSS", "Mobile", and "AI".

# -ダッシュボード-



オンザリンクス様のINTER-STOCK  
倉庫状況を一目で把握



デンソーウェーブ様のORIN3と組み合わせて  
PLCのデバイス情報をリアルタイムに表示

# -DB連携-

設定だけでテーブルと連携  
SQL直指定も可能！

The screenshot displays a web application interface. On the left, a sidebar lists various UI components such as Home, Layout, Controls, DB, Author, List, Script, ProCode, MudBlazor, Radzen Blazor, IgniteUI, CSS, Mobile, and AI. The main content area shows a 'Recipe' form with fields for Name, Serving Size, Description, and Author. Below the form is a table of ingredients with columns for Ingredientname, Quantity, Measurement, Preparation, and Note. To the right, there is a separate 'Dashboard' window showing three charts: a line chart for daily counts, a pie chart for reasons of failure, and a bar chart for order quantity and delivery quantity over time.

一般的なCRUDはもちろんJOINや1Nの関係の表現も可能です。

TableだけでなくViewやSQLにも関連付けることができる所以簡単に集計機能やグラフでの表示もできます。変更履歴も簡単に残せるようにしています。

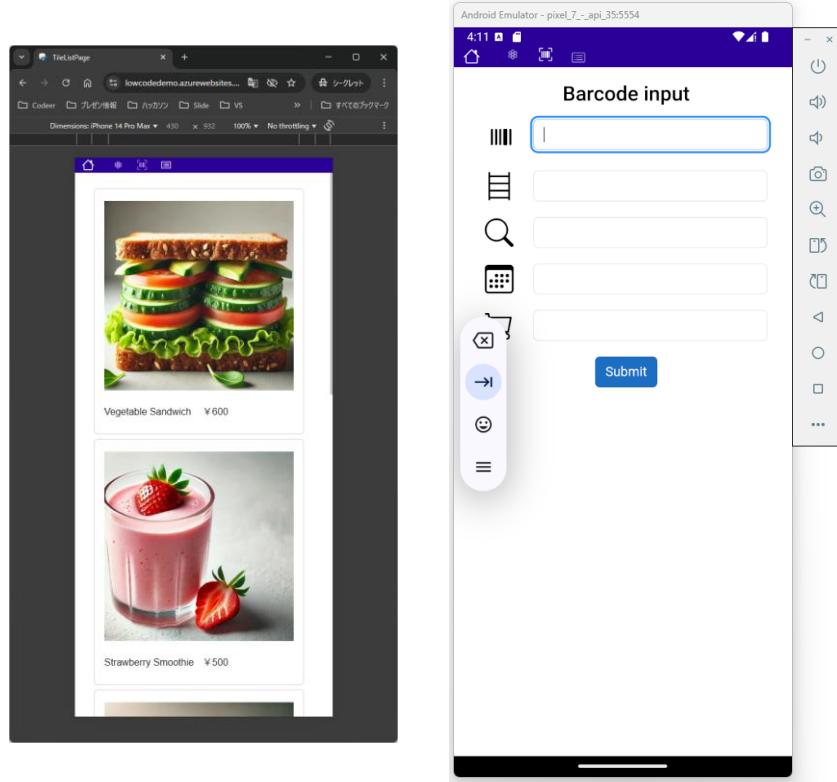
その他検索/論理削除/楽観ロック/作成更新情報など一般的にDBの操作で必要になるものは取り揃えていて多くの既存システムのDBと連携できます。

御社の既存のDBとも連携可能です。

例) SQL Server, Oracle, MySQL, PostgreSQL, SQLite

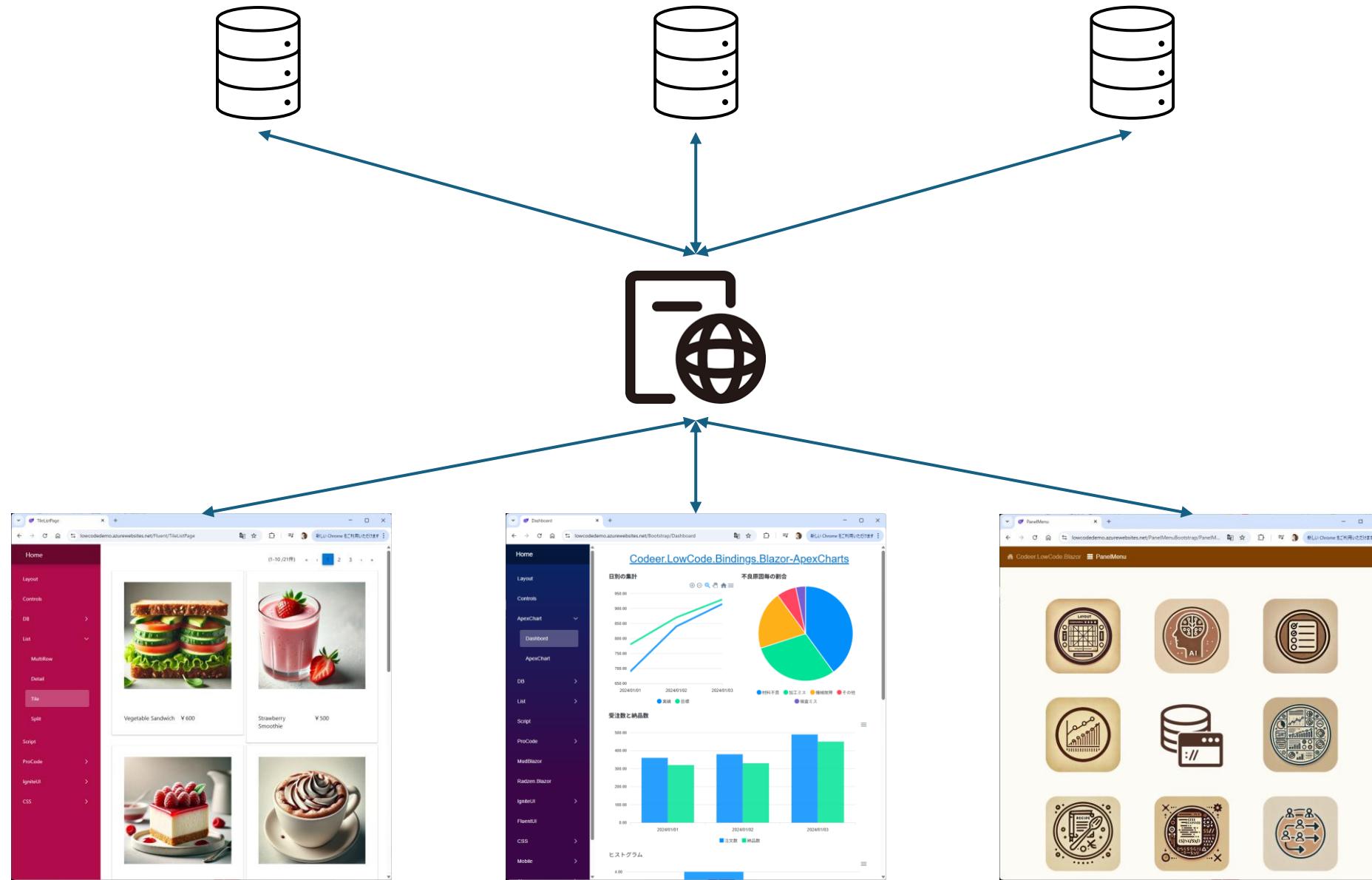
# -モバイル/ネイティブアプリ対応-

Webアプリなのでモバイルでも利用可能  
さらにネイティブアプリも作成可能です！

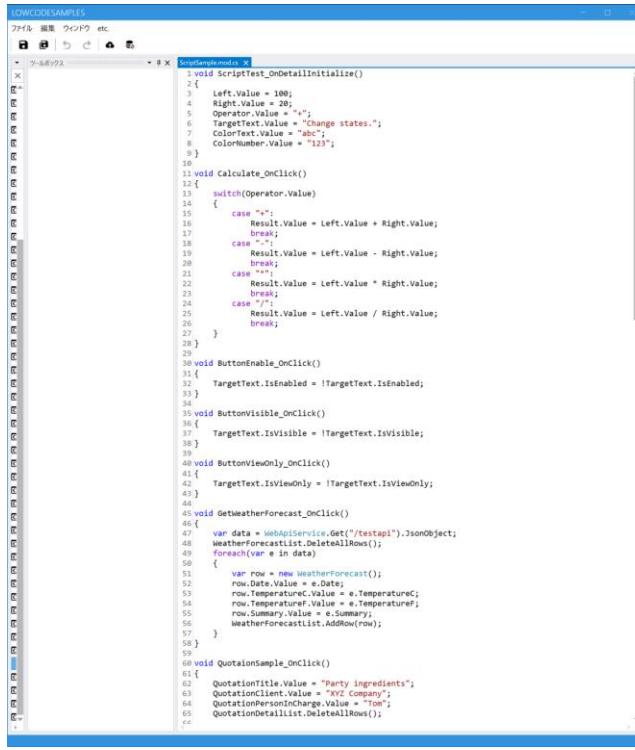


Webアプリなのでお手軽にスマートフォンでもご利用いただけます。  
さらにMAUIを使うことでネイティブアプリも作れます。  
iOS、Android、MAC、WindowsのネイティブAPIを利用できます。  
WindowsにおいてはWinForms、WPFも利用可能です。

# -1つのWebアプリで幅広い利用-

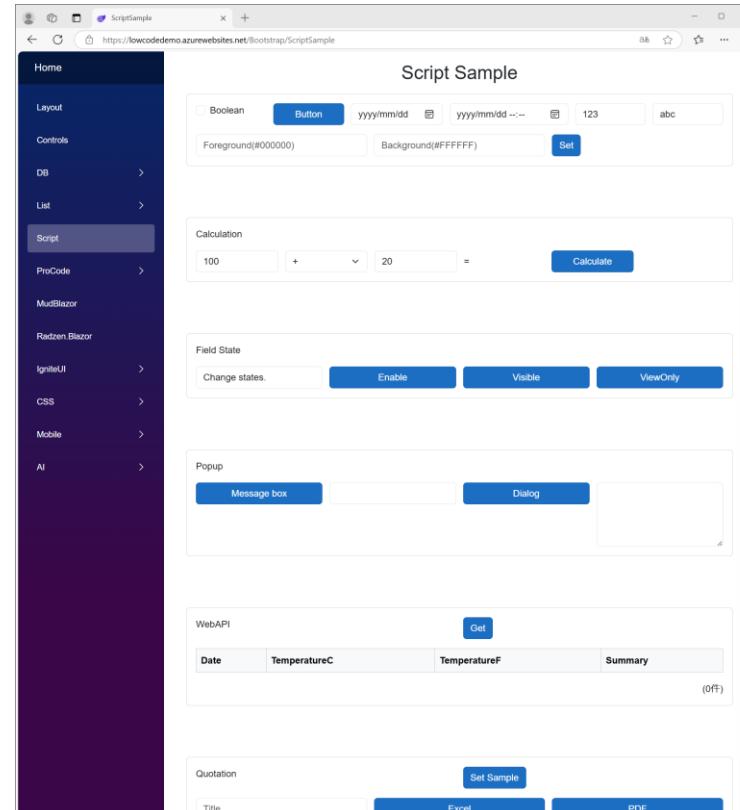


# -スクリプト-



```
LOW-CODE SAMPLES
ファイル 編集 ワンクリック etc.
LOW-CODE SAMPLES
ScriptSample.cs
1 void ScriptTest_OnDetailInitialize()
2 {
3     Left.Value = 100;
4     Right.Value = 20;
5     Operator.Value = "+";
6     TargetText.Value = "Change states.";
7     ColorText.Value = "abc";
8     ColorNumber.Value = "123";
9 }
10
11 void Calculate_OnClick()
12 {
13     switch(Operator.Value)
14     {
15         case "+":
16             Result.Value = Left.Value + Right.Value;
17             break;
18         case "-":
19             Result.Value = Left.Value - Right.Value;
20             break;
21         case "*":
22             Result.Value = Left.Value * Right.Value;
23             break;
24         case "/":
25             Result.Value = Left.Value / Right.Value;
26             break;
27     }
28 }
29
30 void ButtonEnable_OnClick()
31 {
32     TargetText.IsEnabled = !TargetText.IsEnabled;
33 }
34
35 void ButtonVisible_OnClick()
36 {
37     TargetText.IsVisible = !TargetText.IsVisible;
38 }
39
40 void ButtonViewOnly_OnClick()
41 {
42     TargetText.IsViewOnly = !TargetText.IsViewOnly;
43 }
44
45 void GetWeatherForecast_OnClick()
46 {
47     var data = WebAPIService.Get("/testapi").JsonObject;
48     WeatherForecastList.DeleteAllRows();
49     foreach(var e in data)
50     {
51         var row = new WeatherForecast();
52         row.Date.Value = e.Date;
53         row.TemperatureC.Value = e.TemperatureC;
54         row.TemperatureF.Value = e.TemperatureF;
55         row.Summary.Value = e.Summary;
56         WeatherForecastList.AddRow(row);
57     }
58 }
59
60 void QuotationSample_OnClick()
61 {
62     QuotationTitle.Value = "Party Ingredients";
63     QuotationClient.Value = "XYZ Company";
64     QuotationPersonInCharge.Value = "Tom";
65     QuotationDetailList.DeleteAllRows();
66 }
```

デザイナ



The screenshot shows a web application titled "Script Sample" with the URL <https://lowcodedemo.azurewebsites.net/bootstrap/ScriptSample>. The application interface includes:

- Layout:** A sidebar navigation menu with items like Home, Layout, Controls, DB, List, Script (selected), ProCode, MudBlazor, Radzen Blazor, IgniteUI, CSS, Mobile, and AI.
- Controls:** A section showing configuration for a "Button" control, including Foreground (#000000) and Background (#FFFFFF).
- Script:** A "Calculation" section with input fields for 100, +, 20, and a "Calculate" button.
- Field State:** A section for "Change states" with buttons for Enable, Visible, and ViewOnly.
- Popup:** A "Message box" and a "Dialog" section.
- WebAPI:** A "Get" button with tabs for Date, TemperatureC, TemperatureF, and Summary, showing a value of 0°F.
- Quotation:** A section with a "Set Sample" button, "Title" input, and "Excel" and "PDF" buttons.

サンプルサイトURL : [ScriptSample](https://lowcodedemo.azurewebsites.net/bootstrap/ScriptSample)

C#とほぼ同じ構文で記述できます。

僅かな実装で機能を実現できるようにAPI設計をしています。

コード補完も効くので簡単に実装できます。  
デバッグも可能です。

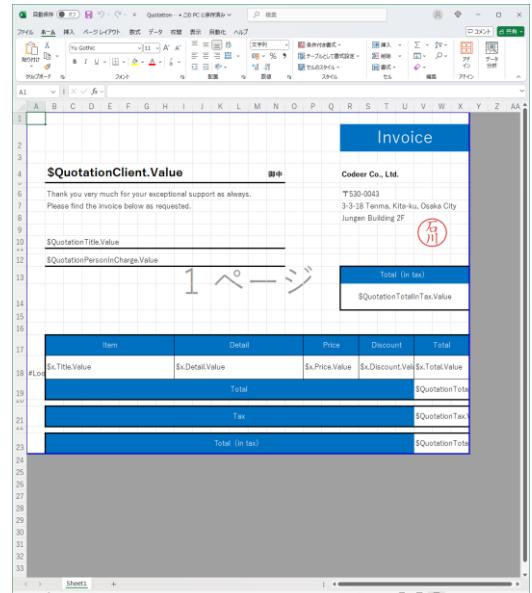
カスタマイズができるプロコードで実装した機能を呼び出すこともできます。

例) 一般的な演算処理、画面制御、  
WebAPIの実行、Excel編集/PDF作成など

# -便利機能-

## 帳票

テンプレートをExcelで作成してそれを書き換えることで帳票にも対応できます。PDFへの変換も可能です。



テンプレートのExcel

A screenshot of a web application demonstrating invoice generation. It shows a 'Script' section where a user has written code to generate an invoice. Below it, there are two preview windows: one showing the generated invoice PDF and another showing the generated Excel file. The invoice details include a company name 'XYZ Company', a date, and various items with their prices and discounts.

サンプルサイトURL : [Invoice](#)

## AI連携

生成AIと連携することで、画像やPDFなどからデータを取り込んだり、開発の効率化を促すことができます。

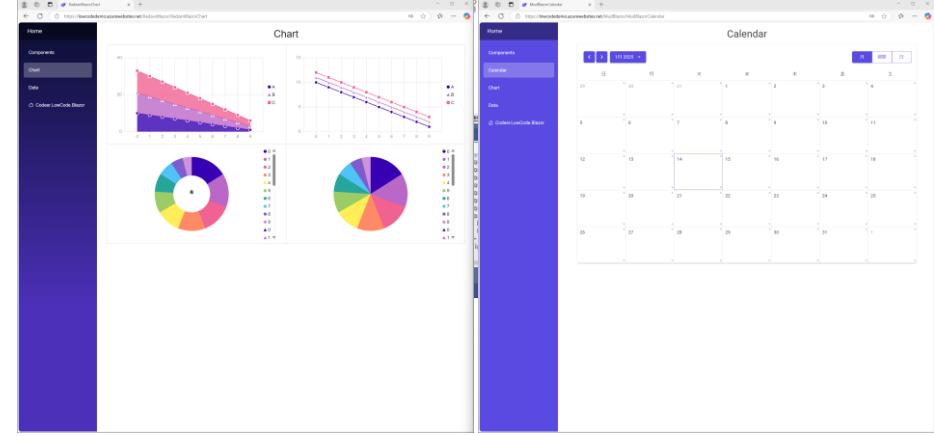
A screenshot of a developer interface showing AI integration. On the left, there's a 'DESIGNER' panel with a tree view of modules like 'AuthModule', '��', '��', '��', etc. In the center, there's a 'CHAT APPLICATION' window with a message about adding AI modules. On the right, there's a 'Name Card' component with a photo of a mountain and some text. A large blue arrow points from the 'DESIGNER' panel towards the 'Name Card' component.

A screenshot of a mobile application interface showing barcode integration. The screen is titled 'Check' and displays a form with fields for 'Name', 'Serving Size', 'Description', and 'Author'. To the right of the form is a barcode scanner icon. Below the form is a table with columns 'Ingredientname', 'Quantity', and 'Measurement'.

他にもバーコードリーダーと連携したり、Excelと連携して大量のデータを一括更新・ダウンロードすることなども可能です。

# -プロコードによる制限のない拡張-

```
using System;
using System.Threading.Tasks;
using System.Text.Json;
using Azure;
using Microsoft.Azure.DocumentDB;
using Microsoft.Azure.DocumentDB.Client;
using Microsoft.Azure.DocumentDB.Core;
using Microsoft.Azure.DocumentDB.Blob;
using Microsoft.Azure.DocumentDB.Blob.DesignLogic;
using Microsoft.Azure.DocumentDB.Blob.Document;
using Microsoft.Azure.DocumentDB.Blob.Resource;
using Microsoft.Azure.DocumentDB.Blob.Design;
using Microsoft.Azure.DocumentDB.Blob;
using Microsoft.Azure.DocumentDB.Prefetch;
using System.Collections.Generic;
using System.Linq;
namespace LowCodeApp.Server.Services.AI
{
    public static class TextAnalysisService
    {
        public static Task<FileToDataSync<ModuleData>> FileToDataSync(ModuleData moduleData, string? mode)
        {
            var test = await ExtractText(FileToDataSync<ModuleData>(moduleData), string.Empty);
            return await TextAnalysisSync(FileToDataSync<ModuleData>(moduleData), moduleData, test);
        }
        public static async Task<FileToDataSync<ModuleData>> TextAnalysisSync(ModuleData moduleData, string? mode)
        {
            var json = await DocumentAnalysisClient(<DesignLogic>.GetDesignLogic().Modules, moduleData);
            return await CreateModule(<DesignLogic>.GetDesignLogic().Modules, moduleData, json);
            new FileToDataSync<ModuleData>(<DesignLogic>.GetDesignLogic().Modules, moduleData);
            new ChatClient(<DesignLogic>.GetDesignLogic().Modules);
            new ChatClient(<DesignLogic>.GetDesignLogic().Modules);
        }
        static async Task<string> FindSimilarities(string[] candidates, string target)
        {
            var config = SystemConfig.Instance.AppSettings;
            var azureClient = new AzureClientBuilder()
                .new Uri(config.OAuth.Uri)
                .new ClientSecret(config.OAuth.ClientKey);
            var chatClient = azureClient.getChatClient(config.OAuth.Model);
            var completion = await chatClient.CompleteChatAsync(
                target,
                new System.Threading.CancellationToken());
            return completion;
        }
        // Please select and return the most likely match from the provided options.
        // If no match is found, return null.
        // The response will be interpreted by a program, so absolutely include no additional information.
    }
}
```



サンプルサイトURL：  
[RadzenBlazorChart](#)

サンプルサイトURL：  
[MudBlazorCalendar](#)



標準SQLでアクセス



<https://arc.cdata.com/jp>

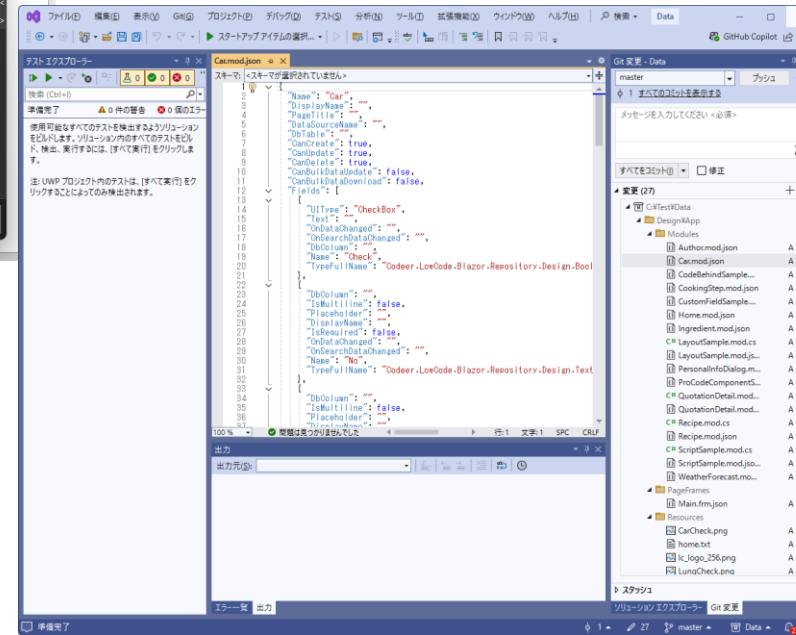
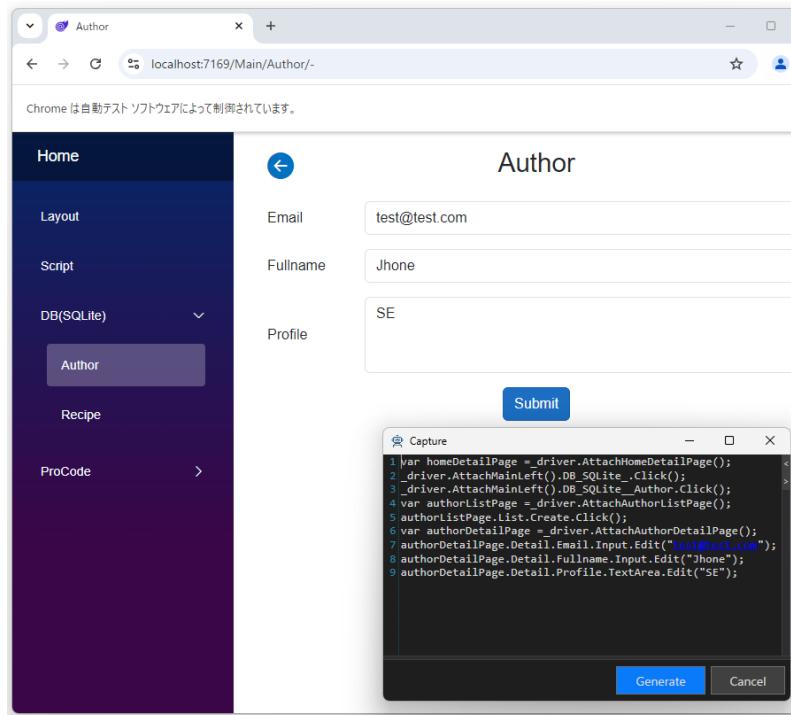
通常のロードコード製品と違い  
カスタマイズに制限がないので  
「できないことがあつたらどうしよう」  
という不安がない

ライブラリによる提供ですので  
アプリソースコードは自由に変更/拡張できます。

これにより、3<sup>rd</sup>パーティ製の外部システムとの連携が可能です。

- ・OSS、3rdPartyのUIコンポーネント
  - ・生成AI
  - ・CData社のデータコネクタ
  - ・etc

# -自動テスト、バージョン管理-



ローコードの部分もバージョン管理やテストが必要です。

設定データはテキストベースのデータですので、gitなどを利用してバージョン管理が簡単に実現できます。

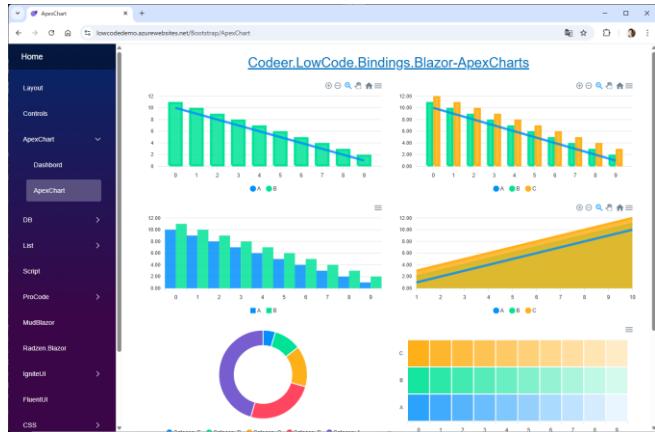
またテストに関しても弊社のTestAssistantProも併せてご利用いただくことで簡単に自動化できます。

# OSSとの連携

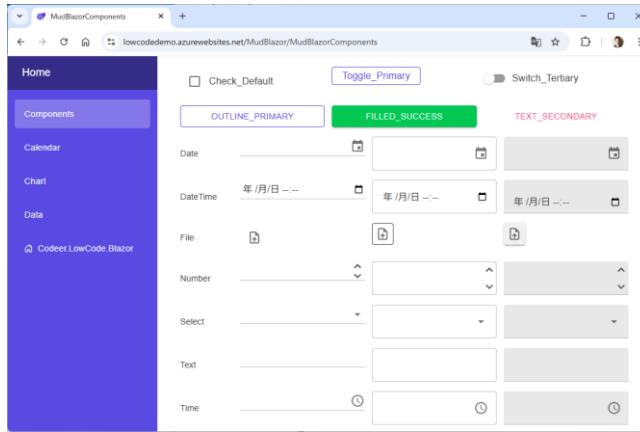
# -BlazorのOSSのコンポーネントと連携-

MITライセンスで公開中

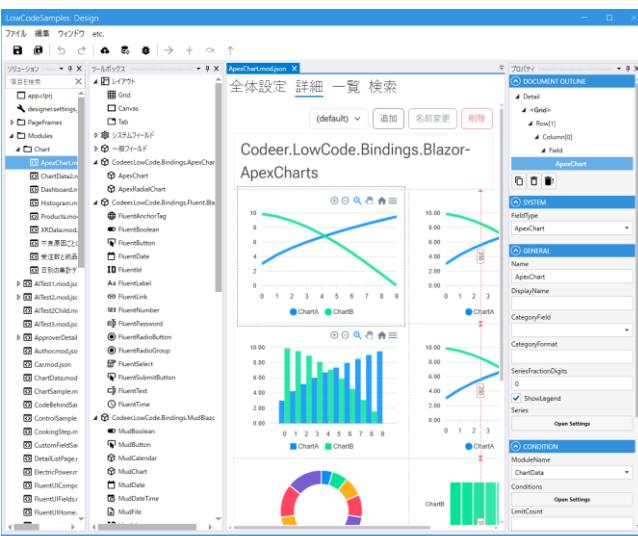
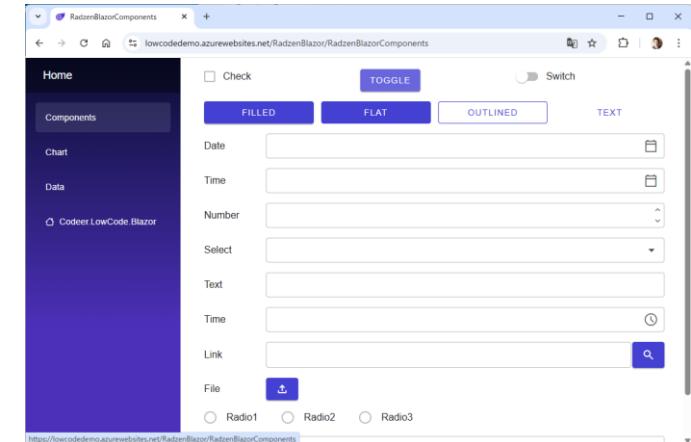
Codeer.LowCode.Bindings.  
Blazor-ApexCharts



Codeer.LowCode.Bindings.  
MudBlazor



Codeer.LowCode.Bindings.  
Radzen.Blazor



Codeer.LowCode.Bindings.Fluent.Blazor  
も対応中

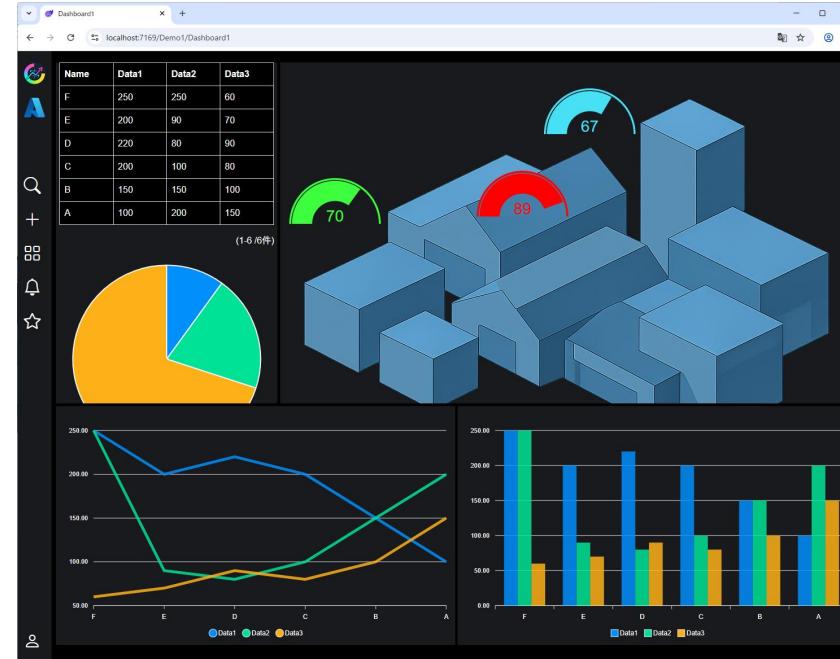
デザイナで画面に配置可能

# -組み込み機器の分野でも-

Codeer.LowCode.Bindings.ORiN3



ORiN3 とは、デンソーウェーブ様が開発しているFA向けの標準ネットワークインターフェース仕様「ORiN」の第3世代バージョンです。PLC、センサー等の様々な機器を統一的なAPIで扱えるミドルウェアとして開発されました。



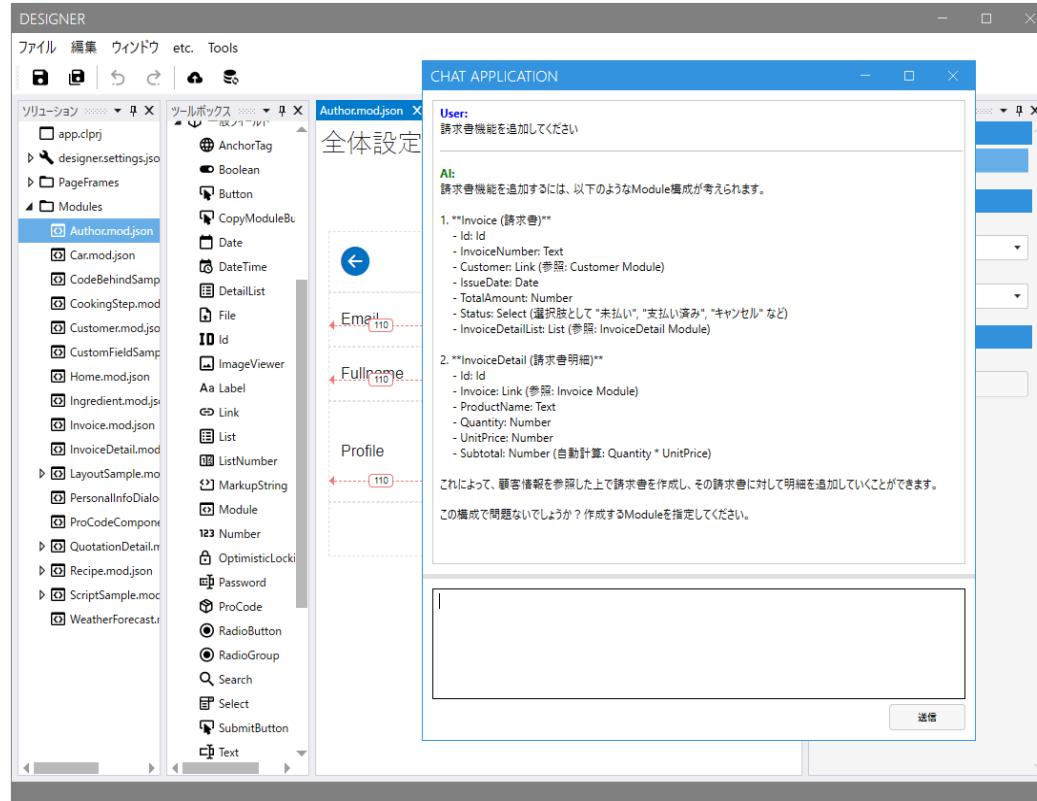
デンソーウェーブの社員さん  
とMITライセンスで開発中



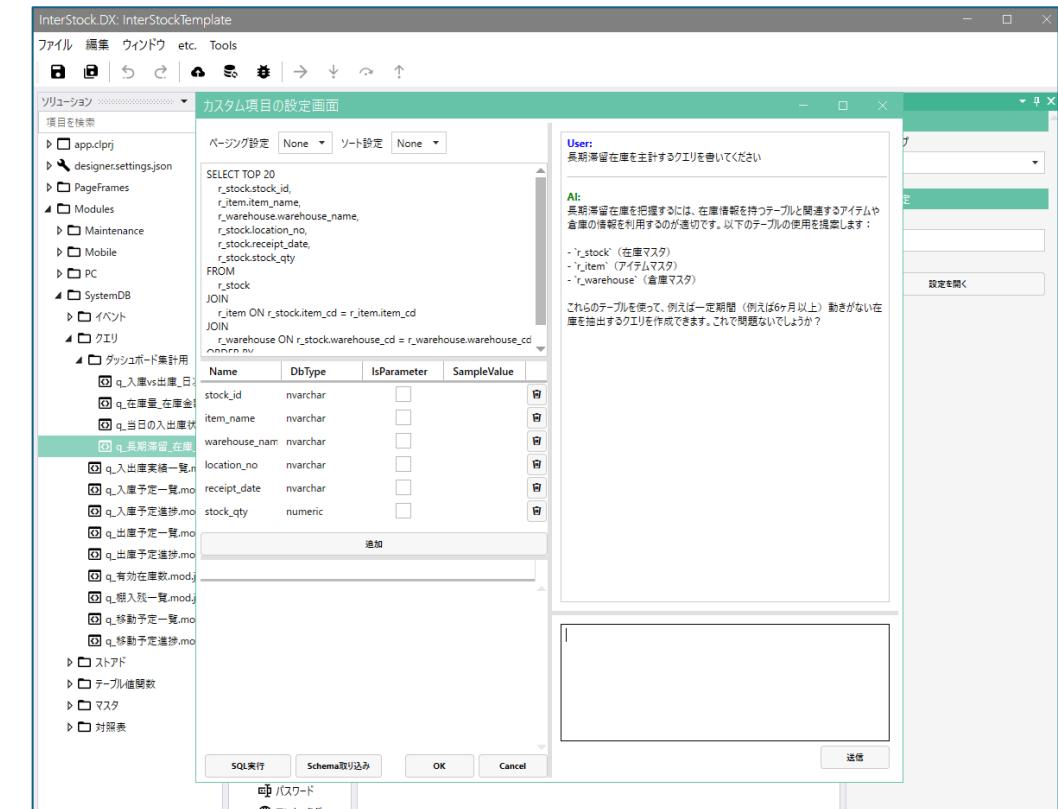
# AIを使って開発

# -AIで作成効率UP-

## モジュールを作成



## 集計のクエリを作成



コードは様々なメタデータを保持しているのでAIと相性が良いです

詳細はデモでご説明します！

# 成長錄

# プログラミング初心者から案件メイン担当者になるまで

これからエンジニアとして頑張る方、教える立場の方の参考に少しでもなれたらうれしいです

成長につながった3つの要素

**Codeer.LowCode.Blazor**

**AI**

**ペアプログラミング**

入社～1週間

1～2週間

デモ作成

案件

**■ 初めてのBlazor**

請求書画面を作成  
→1週間かけてようやく動いた  
  
Blazorの基礎について学ぶ

**■ ローコード開発開始**

Codeer.LowCode.Blazor  
を使って請求書画面を作成  
→1日で完成し、感動

**■ 提案に向けてデモ作成**

とにかくたくさん製品を使って開  
発し、製品の理解を深める  
→できることが増え、自信がつく

**■ 開発をメインで担当**

スピード感をもって開発  
改善もすぐ実施  
→実際に使われる機能を実現

**■ 展示会参加**

製品の特長を学ぶ

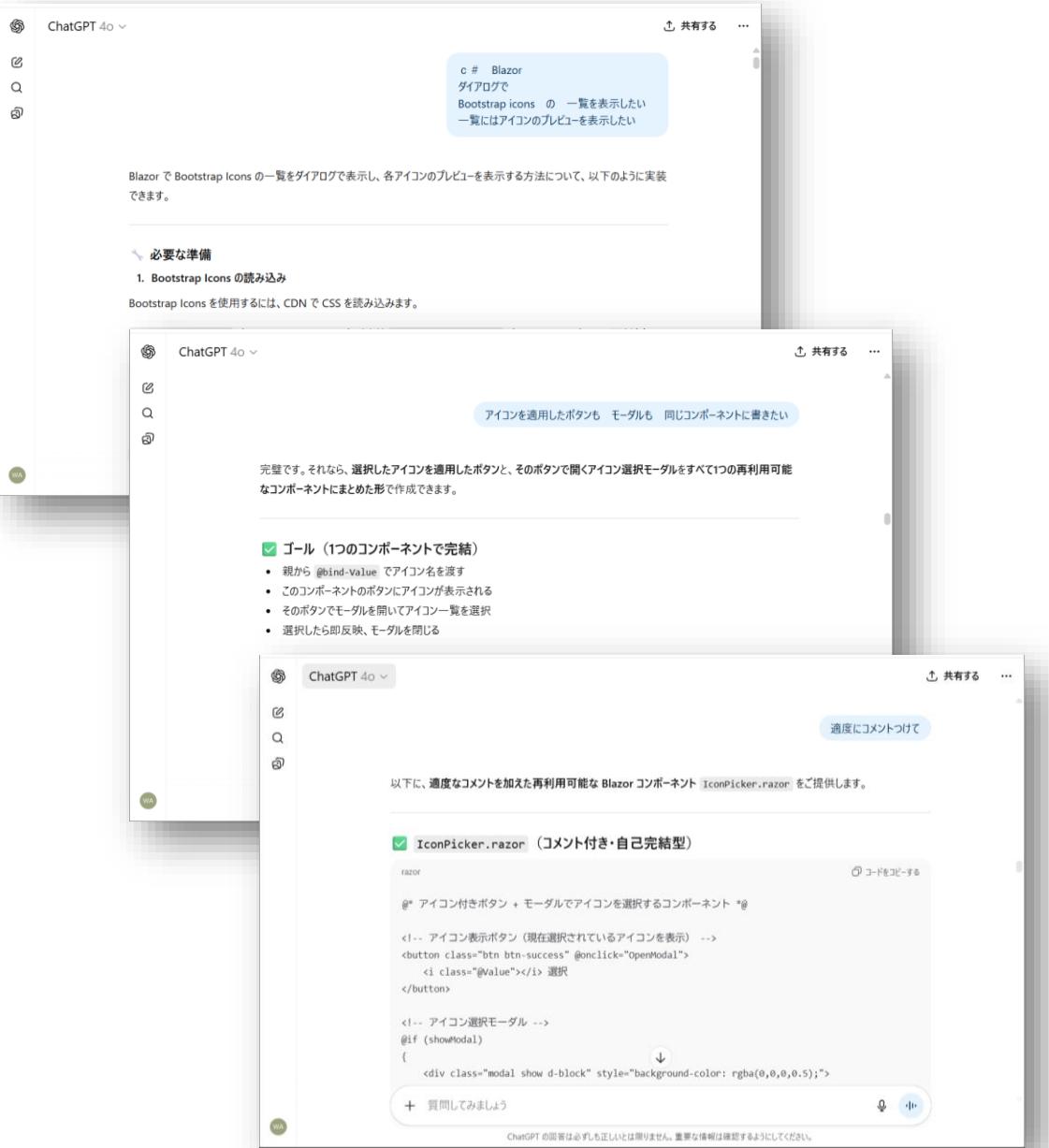
**■ アウトプット**

開発での気づきをもとにマニュア  
ル作成やブラッシュアップを行う

**■ ローコード開発研修**

製品の基本的な使い方をハン  
ズオンで実施

使う→伝える を何度も繰り返すことで  
短期間で案件のメインを担当できるようになった



## ■ 使用しているAI

- ChatGPT(Plus)

## ■ 活用方法

- プログラミングのサポート

- ① やりたいことをざっくり伝える
- ② 動く状態にする
- ③ 徐々に仕様を固めていく
- ④ ソースにコメント書いてもらう
- ⑤ ソースを理解してリファクタリング

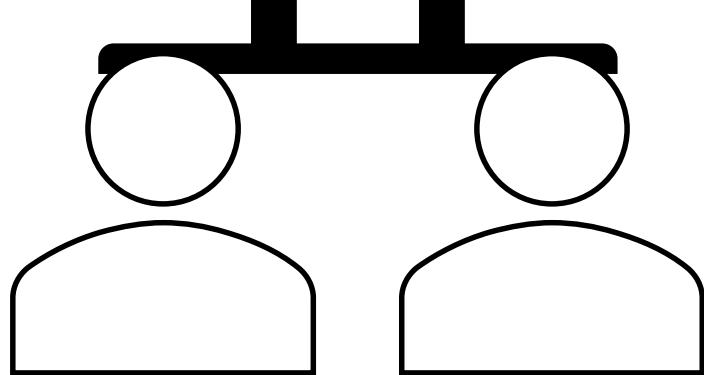
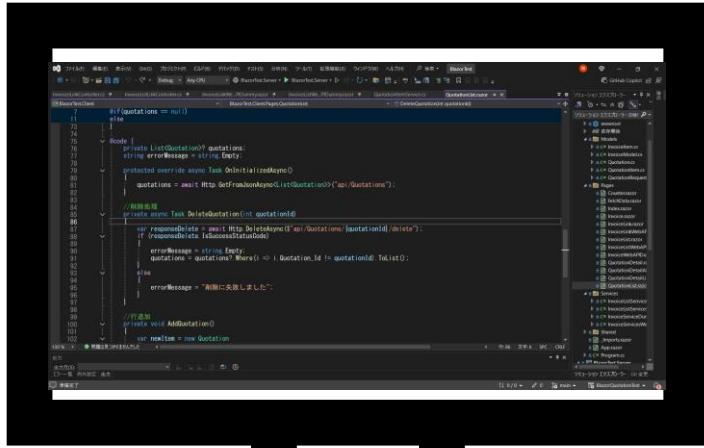
- エラー解消のサポート

エラーメッセージを送ってエラー解消のヒントをもらう

- Git、Azureに関する質問

競合解決、デプロイ作業について教えてもらう

この1年で一番会話した相手  
初歩的な質問もしやすく、開発がスムーズに



## ■ いつ

- ・ プロコードでうまくいかない時

## ■ 方法

- ・ 実現したいことを紙に書きだす
- ・ ヒントをもらう
- ・ 実際にコード書いてもらう

## ■ 効果

- ・ 考え方を学ぶ
- ・ うまくいかない時にどこを見るのか学ぶ  
開発者ツール、呼び出し履歴、ウォッチなど

「見る学習」はかなり効果的

## ■ テンプレートダウンロード（トライアル版）

<https://marketplace.visualstudio.com/items?itemName=Codeer.LowCodeBlazor>

## ■ マニュアル

<https://github.com/Codeer-Software/Codeer.LowCode.Blazor.Manual/blob/main/JP/README.md>

## ■ ライセンス情報

上記マニュアルの「ライセンス情報」をご確認ください

## ■ 紹介や操作方法の動画

<https://www.youtube.com/@codeer373>

## ■ サンプルサイト

<https://lowcodedemo.azurewebsites.net/>



■ 採用情報

<https://www.codeer.co.jp/Recruit>

■ ホームページ

<https://www.codeer.co.jp/>

■ お問い合わせ

<https://www.codeer.co.jp/Contact>





# THANK YOU !



Codeer.LowCode.Blazor