

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**

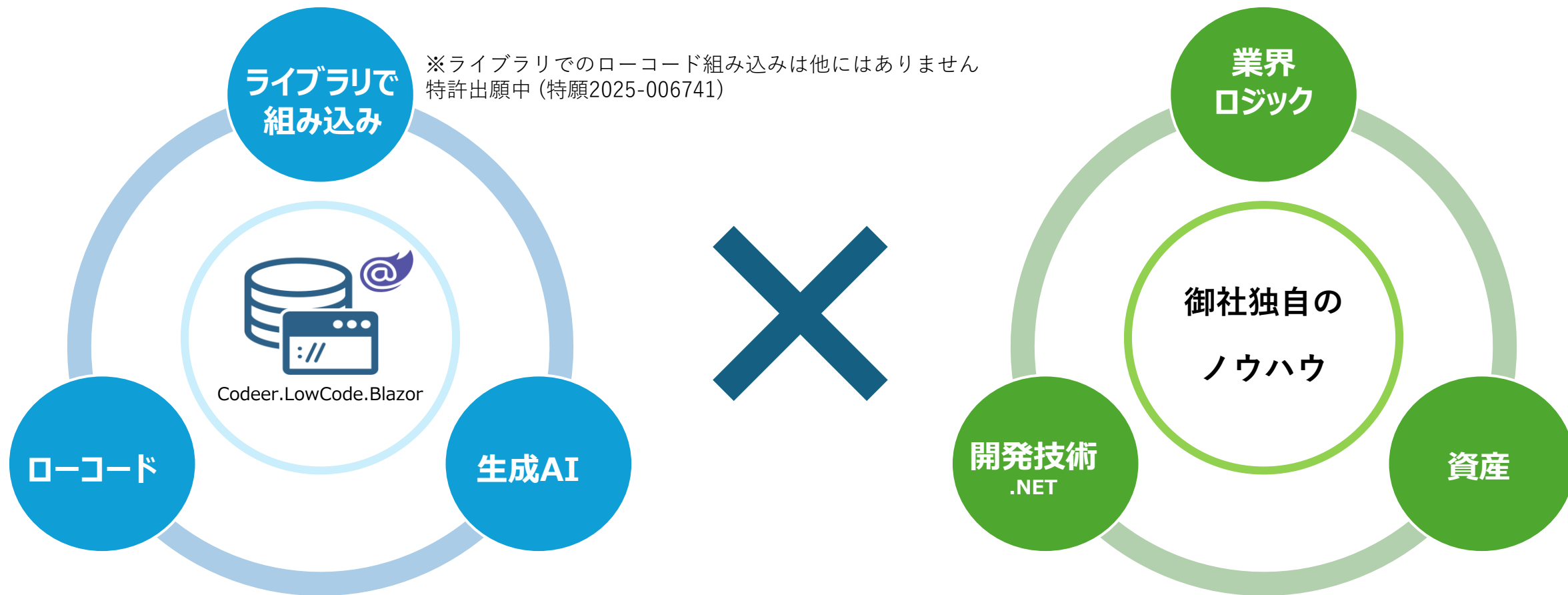
作成することができます。

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？~~ 作いませんか？



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)

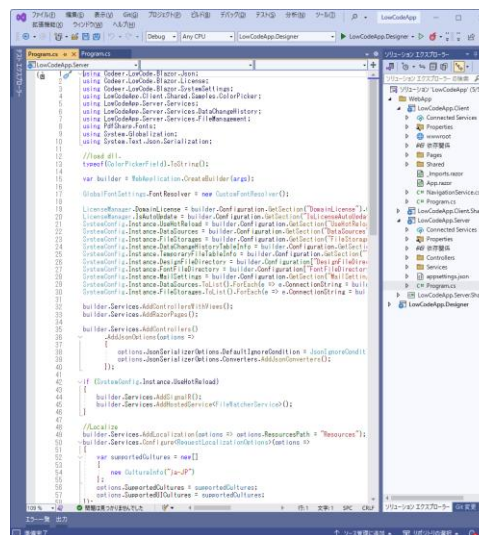
ローコードアプリを ~~使いませんか？~~ 作りませんか？



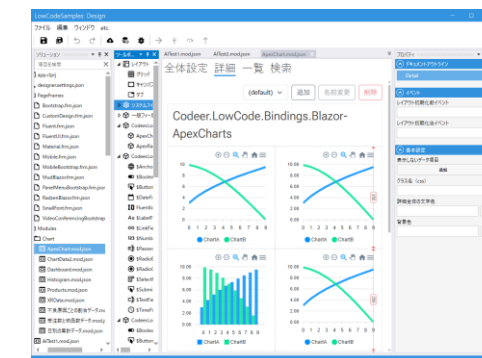
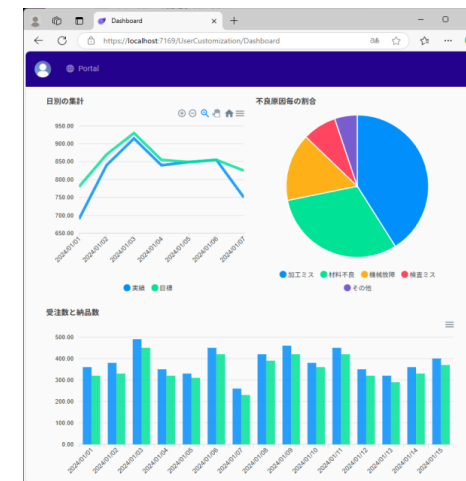
実行エンジン型の
ローコードエンジン
ライブラリ



OSSのライブラリとも
自由に組み合わせることができる

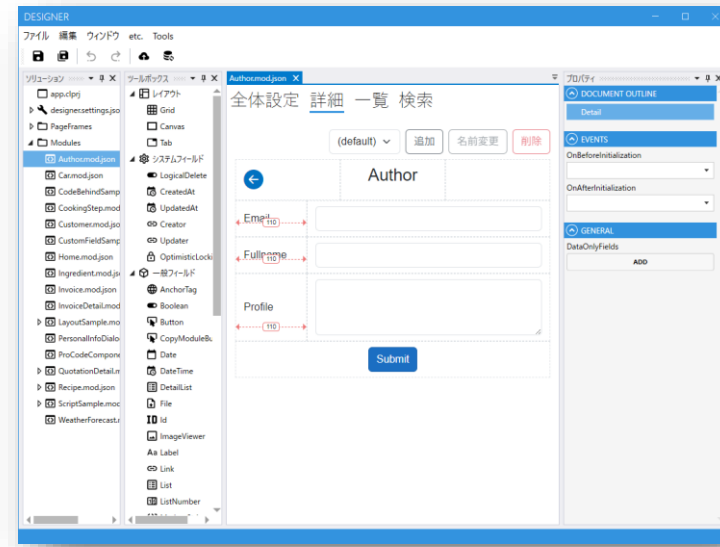
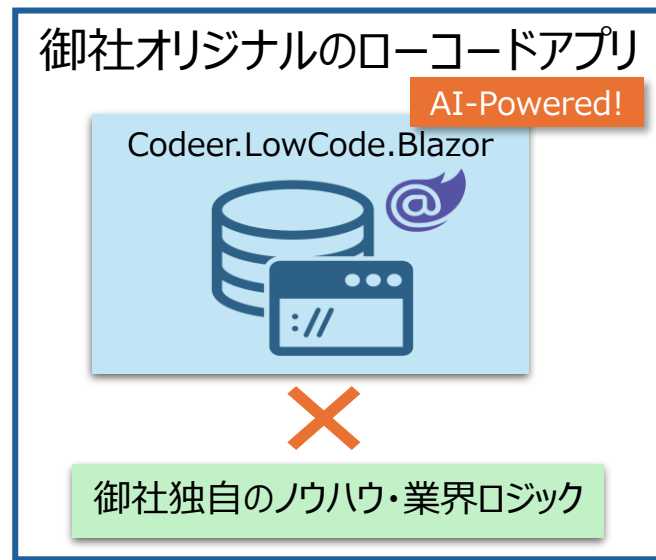


フルスクラッチがベースなので
無制限にカスタマイズ

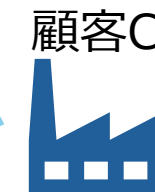
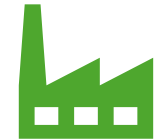


各分野に特化したオリジナルの
ローコードアプリを作成！

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！



顧客別
ビジネスアプリ
顧客A



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

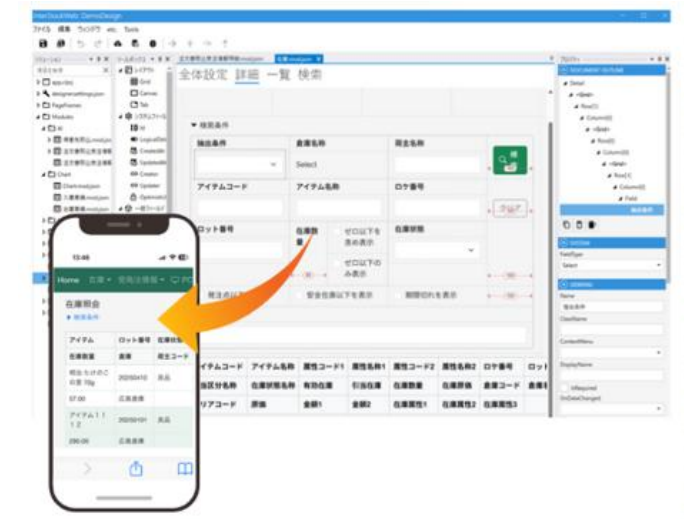
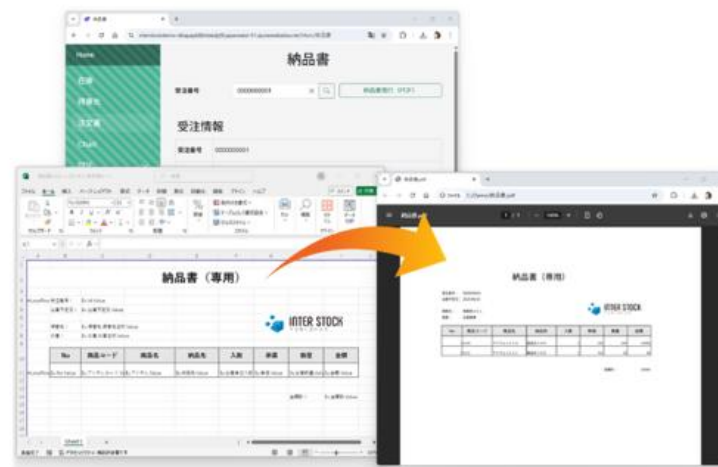
通常の開発手法ではローコード機能の設計/実装は数年規模・高額の投資を要する必要があります。しかし**Codeer.LowCode.Blazor**を利用すれば圧倒的に短い期間・低コストで実現できるのが非常に魅力的でした！



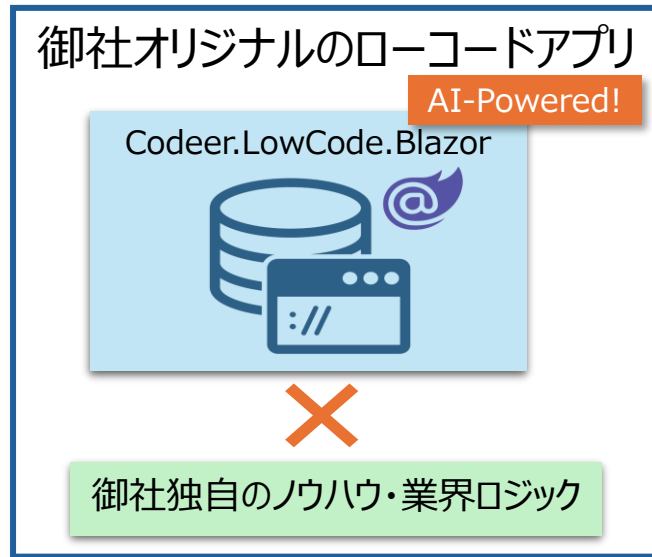
集計

帳票

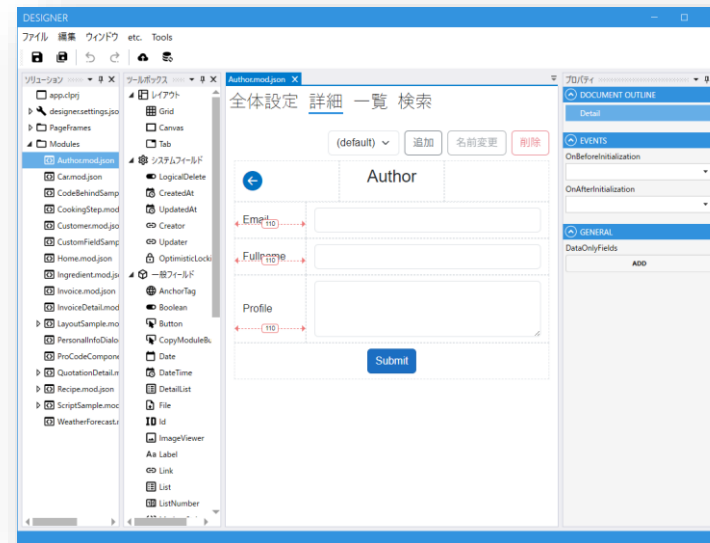
モバイル



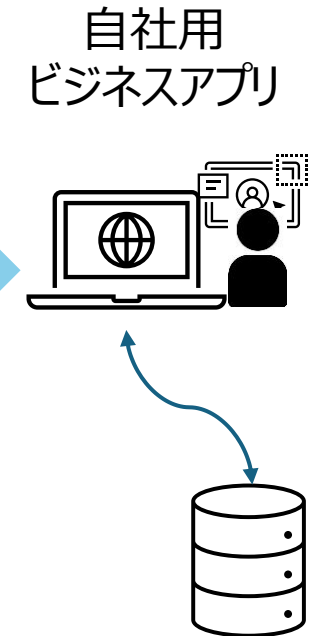
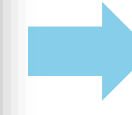
低コストで高品質なアプリを開発できます！



あっ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



既存の資産と簡単に連携
DB
(SQL Server, Oracle, MySQL,
PostgreSQL, SQLite)

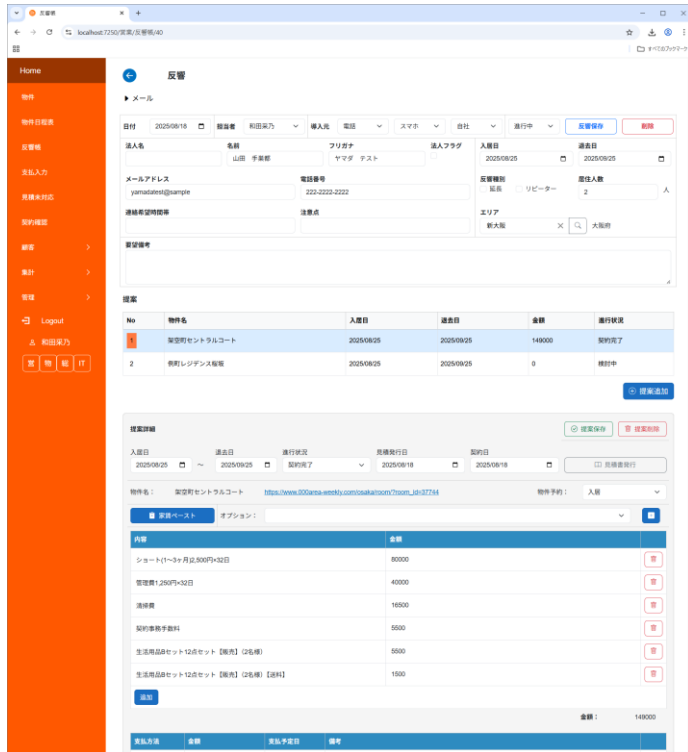
Blazorで開発フレームワークを最新化！

BraTTo様 マンスリーマンション管理

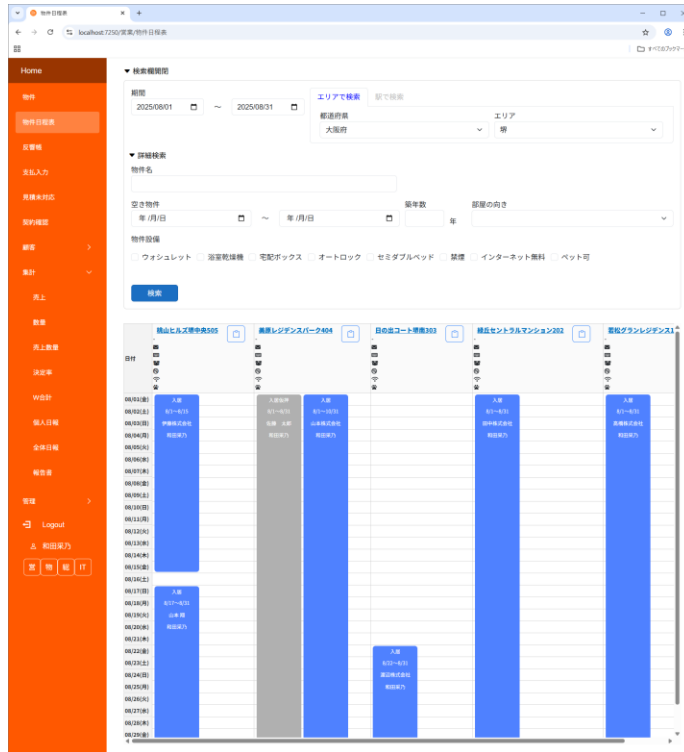


BraTTo様の案件はコンペ形式で、Codeer.LowCode.Blazor を活用したご提案が、**圧倒的なコストパフォーマンス**（低価格・短納期・高機能）により採用されました。2位につけていた企業がベトナムでのオフショア開発を提案していた中、弊社の提示価格はその半額以下。コスト競争力でも大きく差をつける結果となりました。また、通常のローコードツールと異なり、必要に応じてフルスクラッチでの開発も可能なため、機能面でも一切妥協はありません。

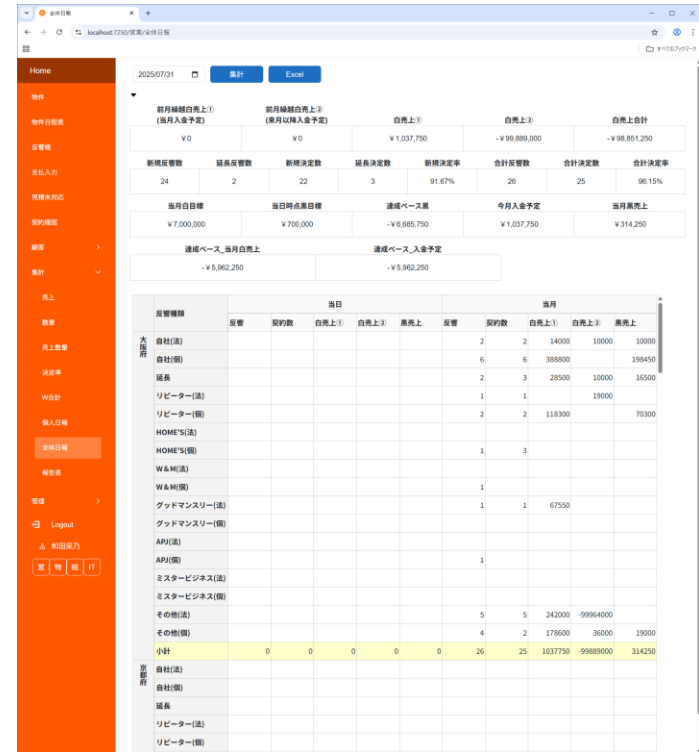
BraTTo様 マンスリーマンション管理



複雑な画面でもローコードで作成可能



特種なコンポーネントは
カスタマイズで作成



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

見積

Test用のお見積り

クレーム数 (独立項) 3 クレーム数 (従属項) 2 回数 10

顧客 株式会社 料金テーブル選択 テンプレート選択 合計金額 34500

品目	名前	コメント	ItemCode.IsGovernmentFee
001	印 紙 代		True
002	基本手数料 (独立請求項 一項目)		False
003	二項目以降の 請求項一項目 に追加する手 数料 (独立請 求項)		False
004	二項目以降の 請求項一項目 に追加する手 数料 (従属請 求項)		False
005	追加書類 (40文字×50行 /頁)	※特許請求の範 囲及び明細書の み。特許、要約 書は除く。	False
006	追加書類 (行)		False
007	図 面 代 (一回あた		False

御 見 積 書

2024年7月1日
見積番号：24-0001

株式会社最新技術 御中

株式会社R&C IP WORKS
〒330-0001
水戸市北区中之島3丁目3番1号
中之島三井ビルディング16階

いつも感謝のお礼でもありがとうございます。
下記の通り見積り申し上げます。ご検討のほどよろしくお願い申し上げます。

案件：Test用のお見積り

御見積額 (消費税込)

37,950

項目	数量	金額
印 紙 代	1	14,000
基本手数料 (独立請求項一項目)	1	1,000
二項目以降の請求項一項目に追加する手数料 (独立請求項)	2	1,000
二項目以降の請求項一項目に追加する手数料 (従属請求項)	2	1,000
追加書類 (40文字×50行/頁)	5	2,500
追加書類 (行)	200	10,000
印 紙 代 (一回あたり。明細書中の表も同様)	10	5,000
契約書作成料	1	0
電子出願料	1	0
計		34,500
消費税		3,450
合計		37,950

注

- ・特許査定となりましたら特許料を納付されない場合でも成功報酬は別途ご請求させていただきます
- ・特許後に権利を維持する場合は、特許料の納付が必要です
- ・消費税 (別件請求にて代金) は、別途ご請求させていただきます
- ・上記見積書における消費税は、現時点での税率10%で算出しております

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

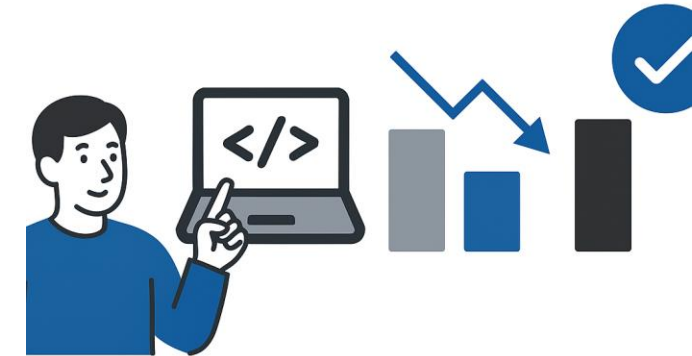
項目	内容	コスト削減
高速なプロトタイピング を使った要件定義	仕様不具合による手戻り工数削減	20%削減
画面/機能作成	バックエンド処理まで含めた工数削減	50～80%削減
共通化による品質向上	不具合修正工数削減	50～80%削減
自動テスト	開発時の隠れたテスト工数削減 回帰テストの工数削減 作業の負荷分散	50～80%削減
メンテナンス	調整工数の削減	50～80%削減

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。

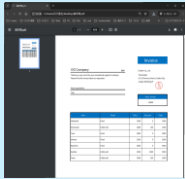


Codeer.LowCode.Blazorの特徴

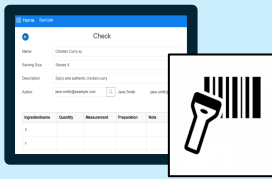


充実の機能

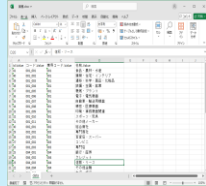
帳票出力



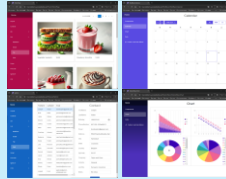
バーコードリーダー連携



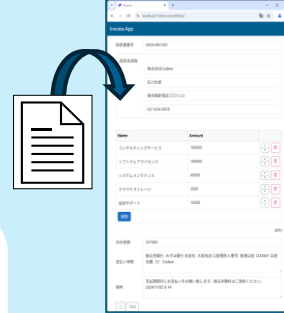
Excel連携



リッチなUI



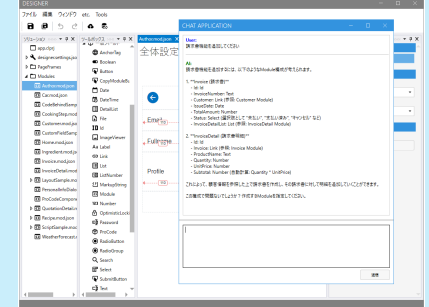
AIによるデータ取り込み



生成AI

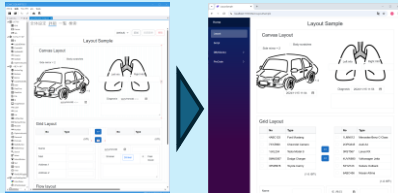


AIによる開発サポート

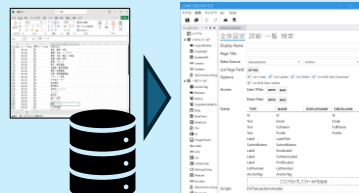


Codeer.LowCode.Blazor

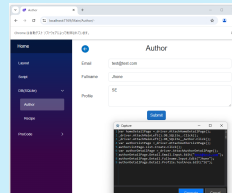
簡単に画面や機能を作成



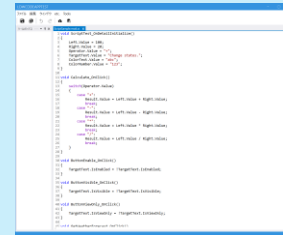
ExcelやDBの定義をもとに作成



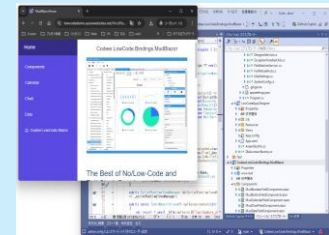
自動テスト



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

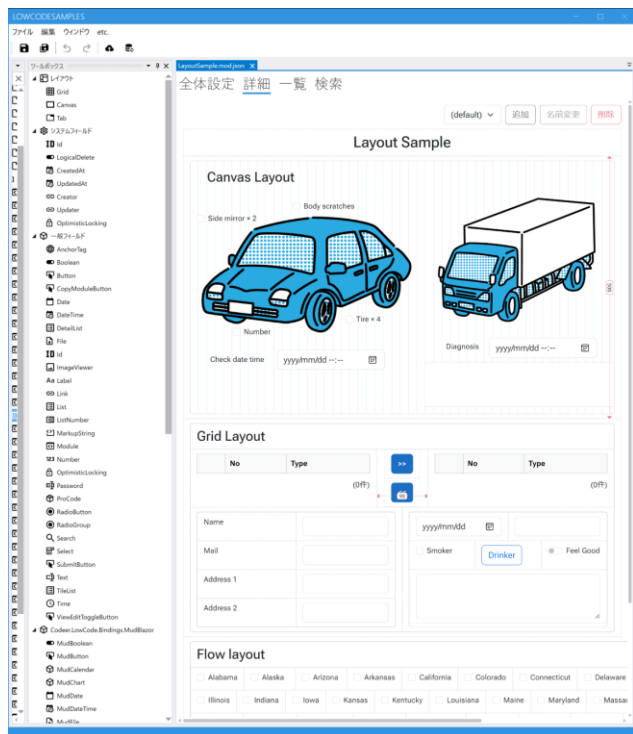


開発の効率化

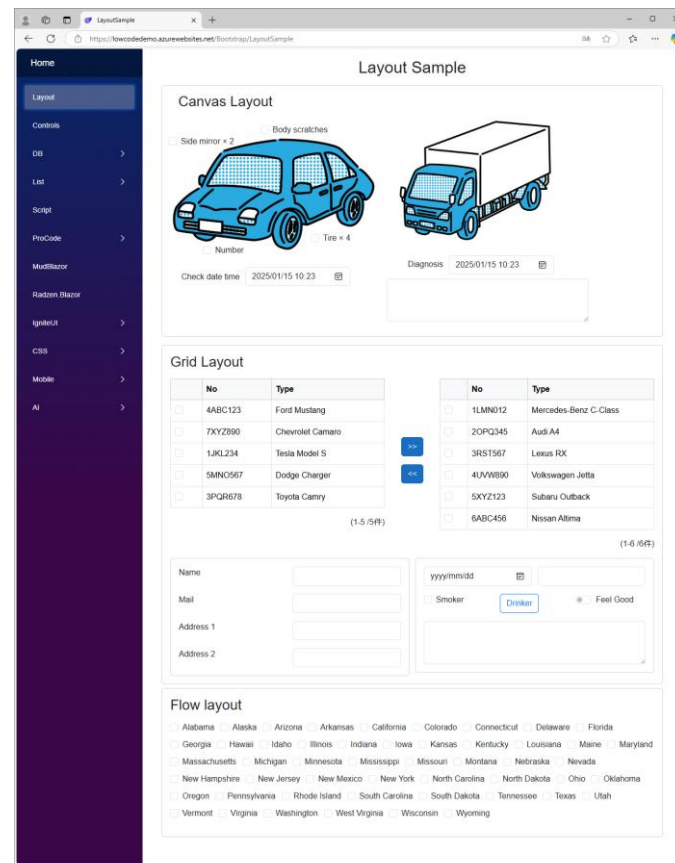
カスタマイズ



-レイアウト-



デザイン

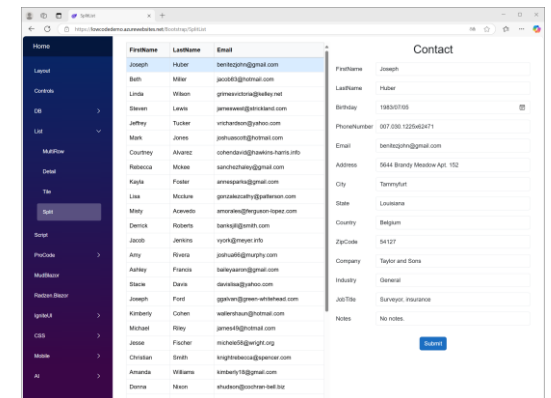
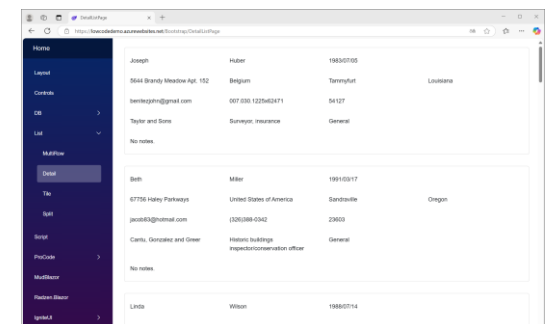
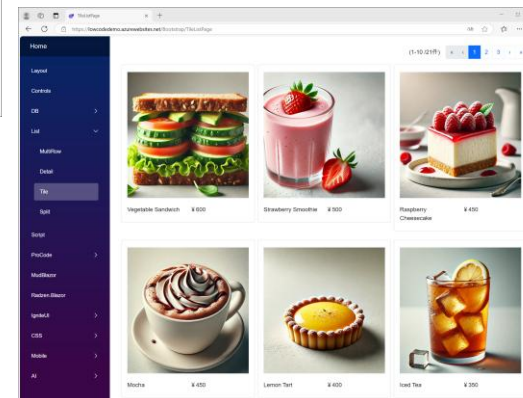
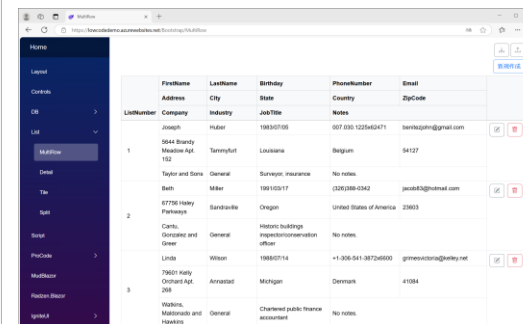


サンプルサイトURL : [LayoutSample](https://www.demio.azurewebsites.net/Bootstrap/LayoutSample)

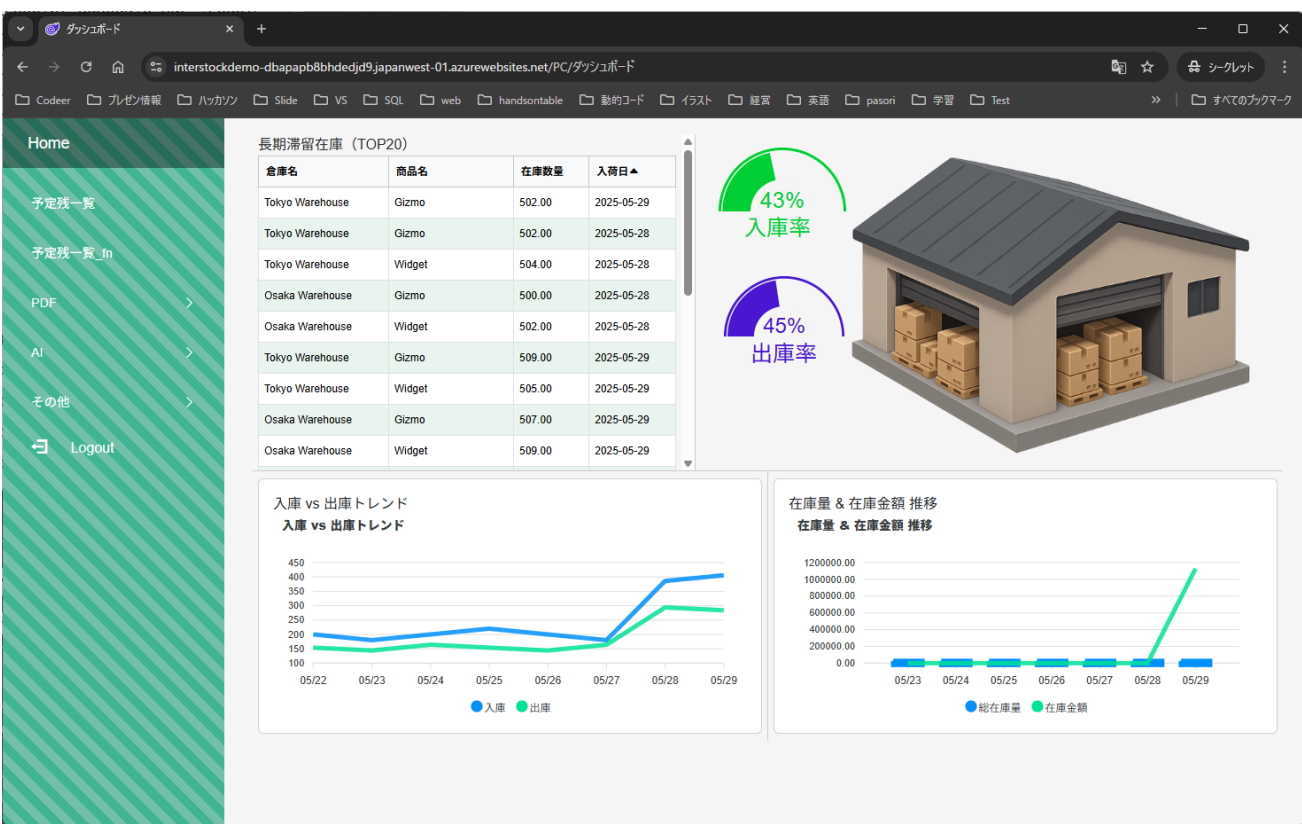
CanvasLayout、GridLayout、FlowLayoutの組み合わせで自由に画面を作成可能です。

各UI部品の連動もノーコードや僅かなスクリプトで作成可能です。

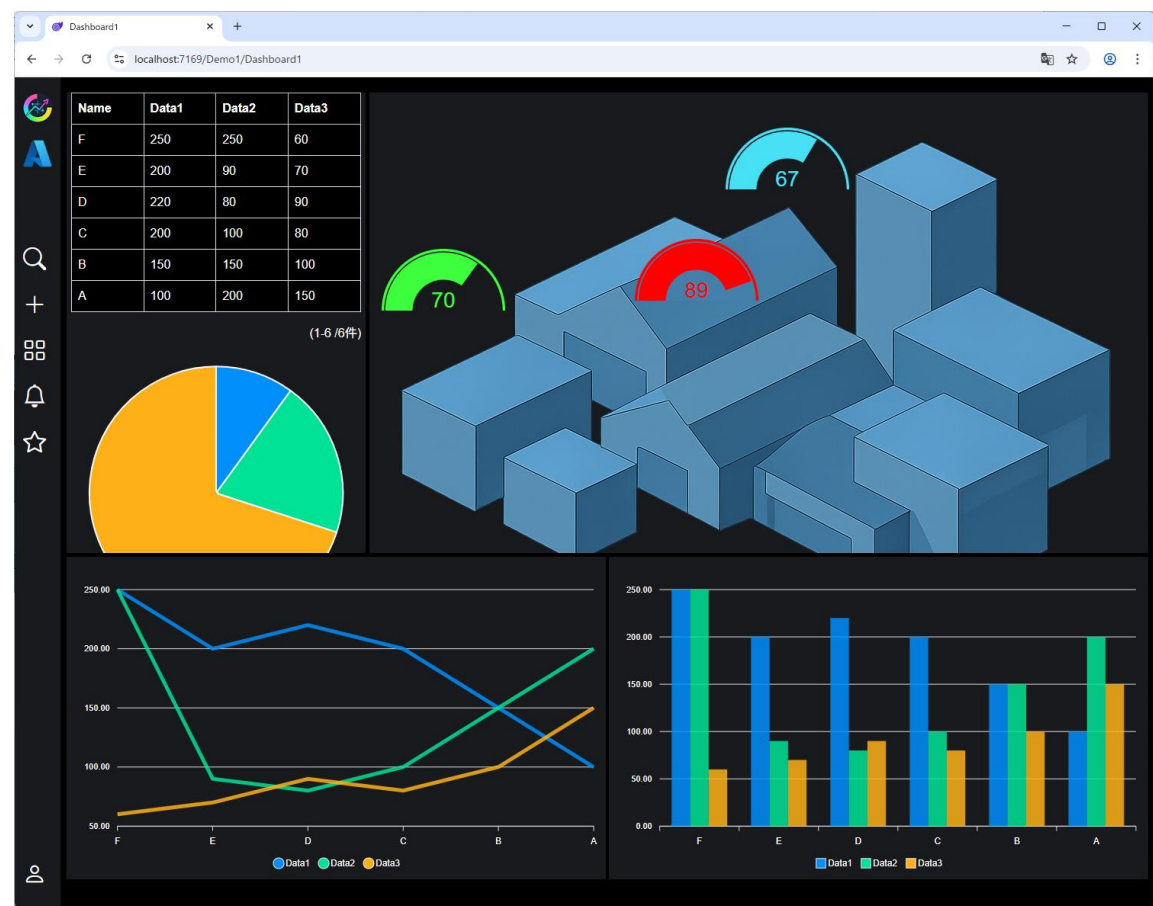
サイドバー、ヘッダ、フッターの表示も設定可能です。



-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！

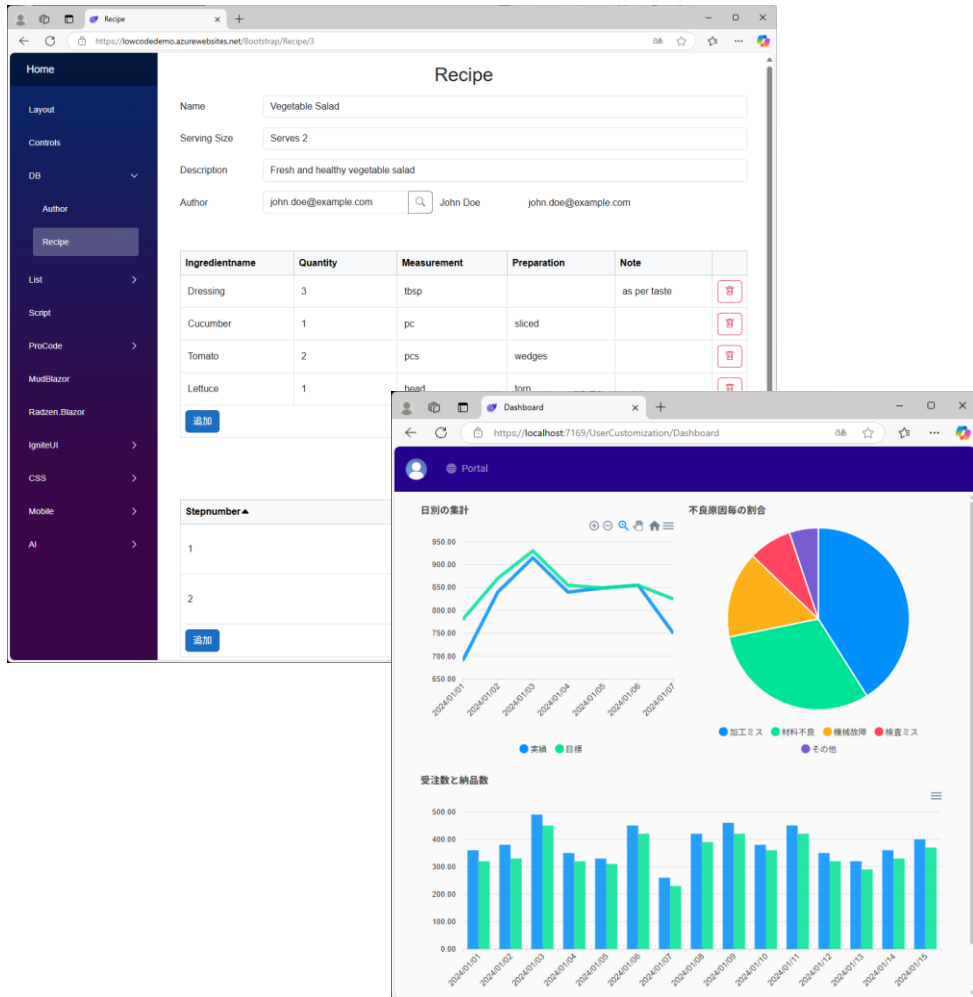
一般的なCRUDはもちろんJOINや1Nの関係の表現も可能です。

TableだけでなくViewやSQLにも関連付けることができるので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。

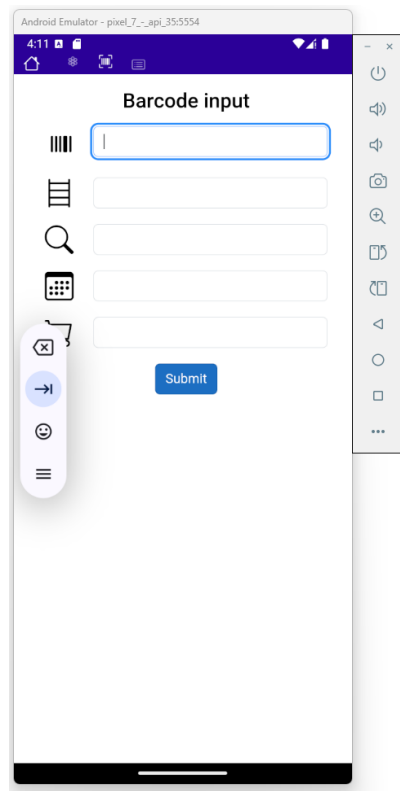
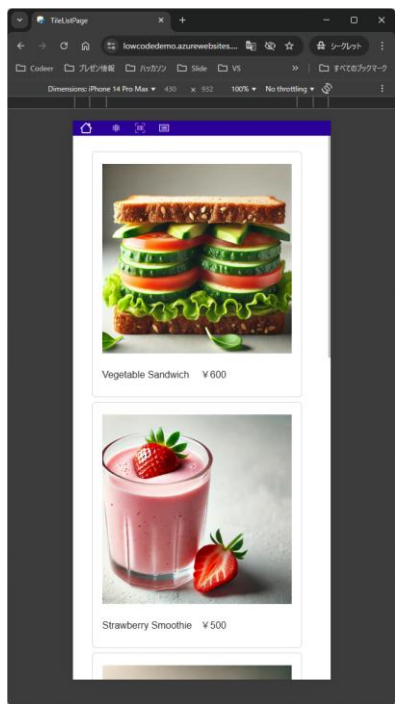
御社の既存のDBとも連携可能です。

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



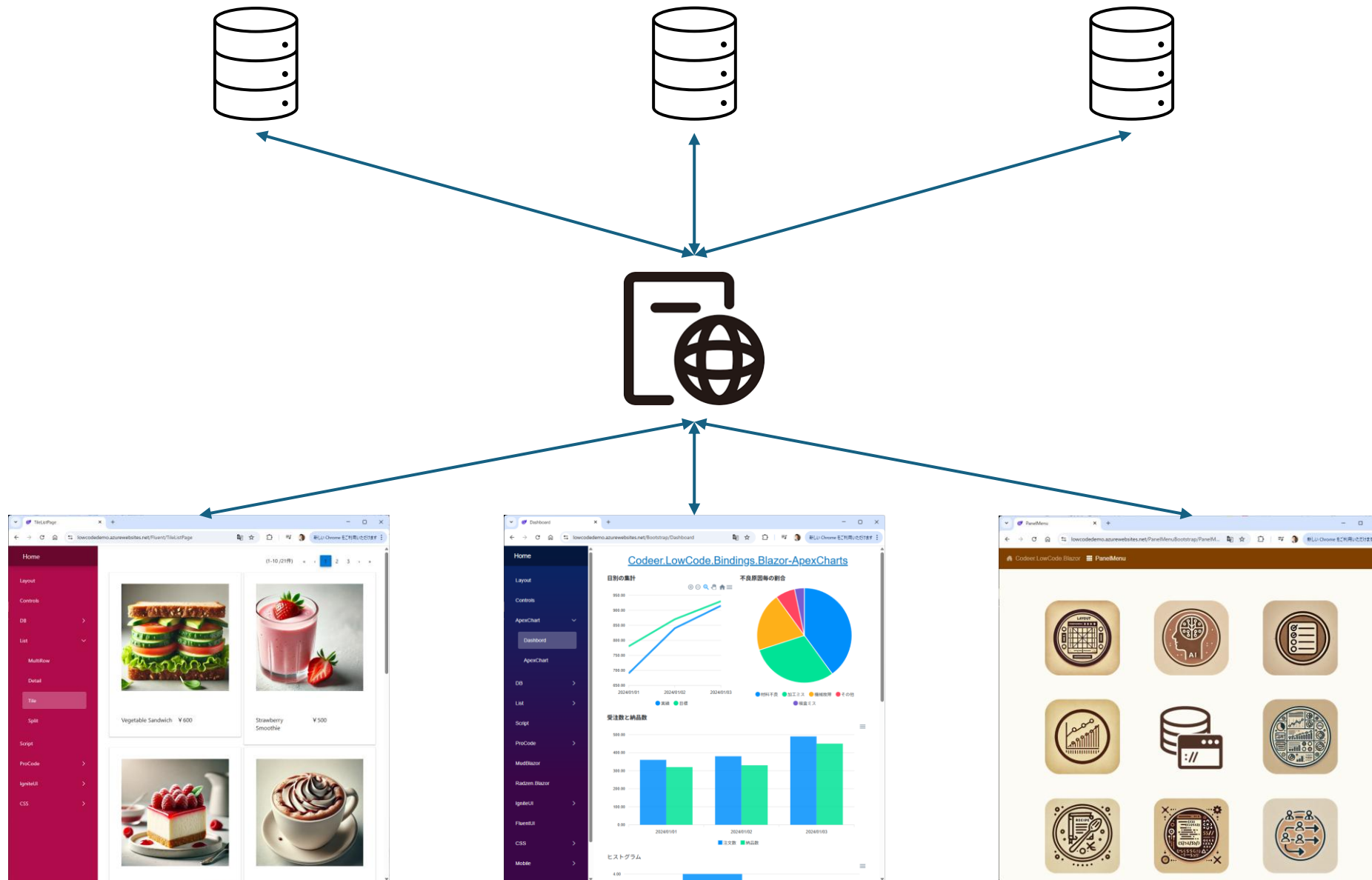
-モバイル/ネイティブアプリ対応-

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、アンドロイド、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-

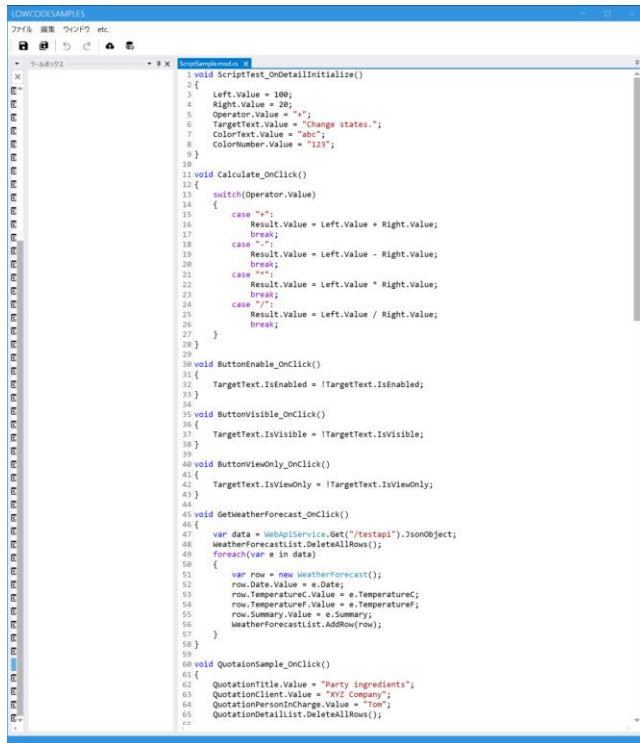


複数のDBと連携

複数機能を無理なく作成

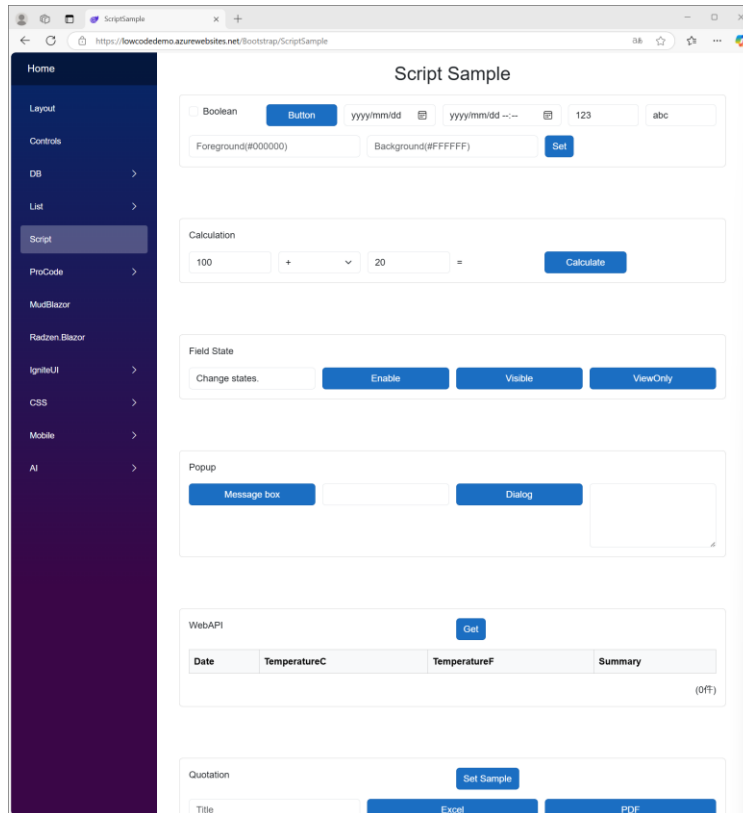
権限による画面の分岐

-スクリプト-



```
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 = WebApiClient.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



サンプルサイトURL : [ScriptSample](https://nowcodedemo.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。

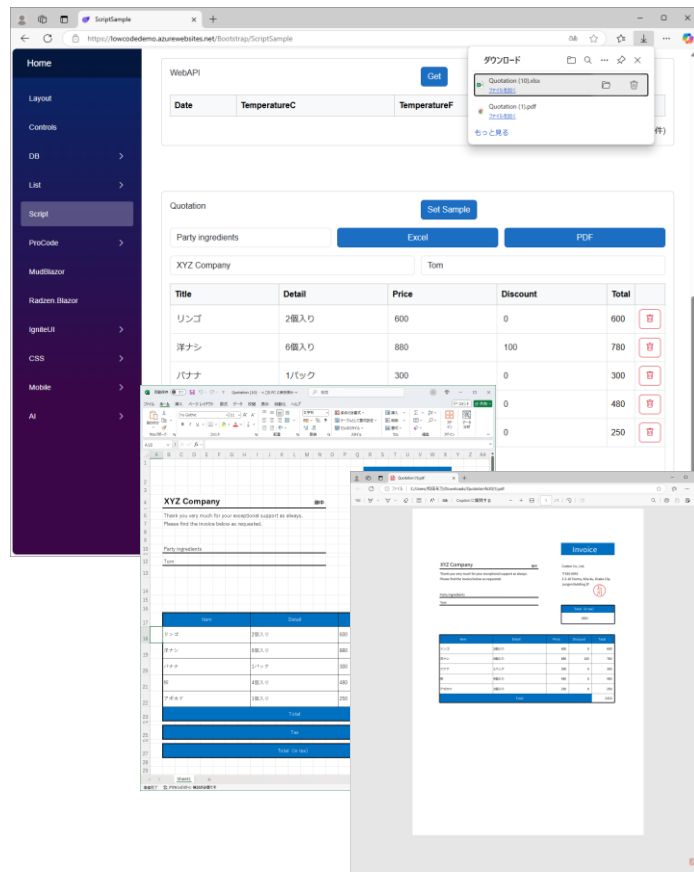
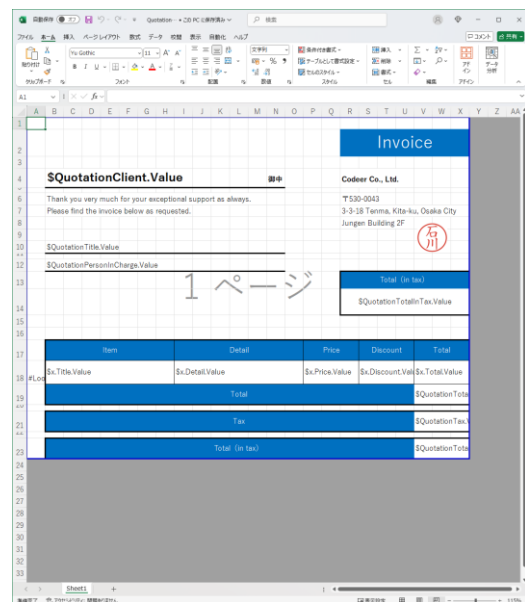
カスタマイズができてプロコードで実装した機能呼び出すこともできます。

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。

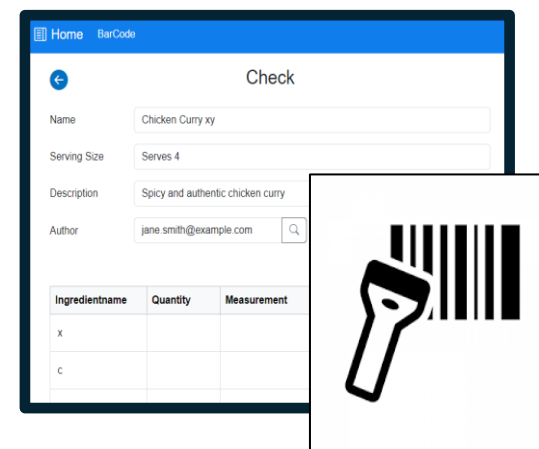
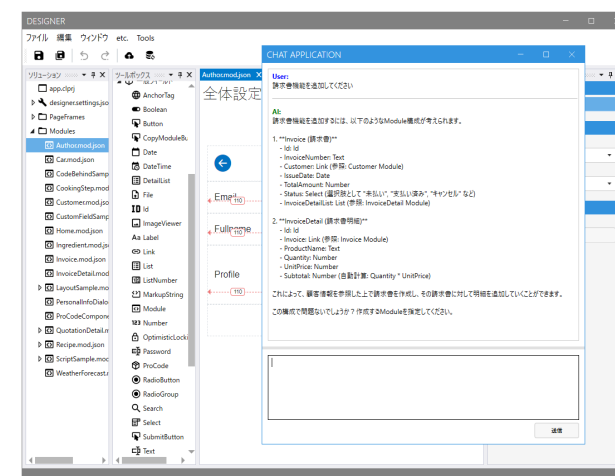
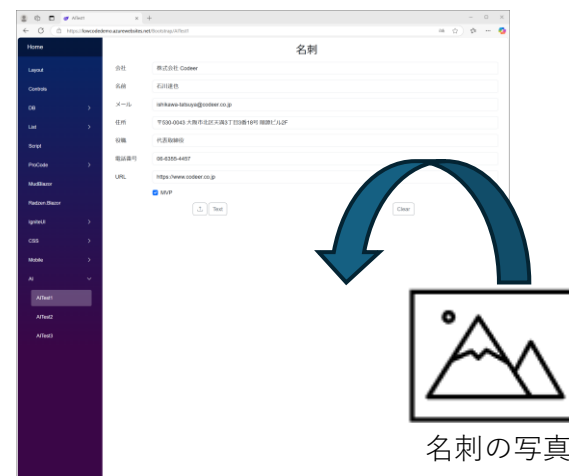


テンプレートのExcel

サンプルサイトURL : [Invoice](#)

AI連携

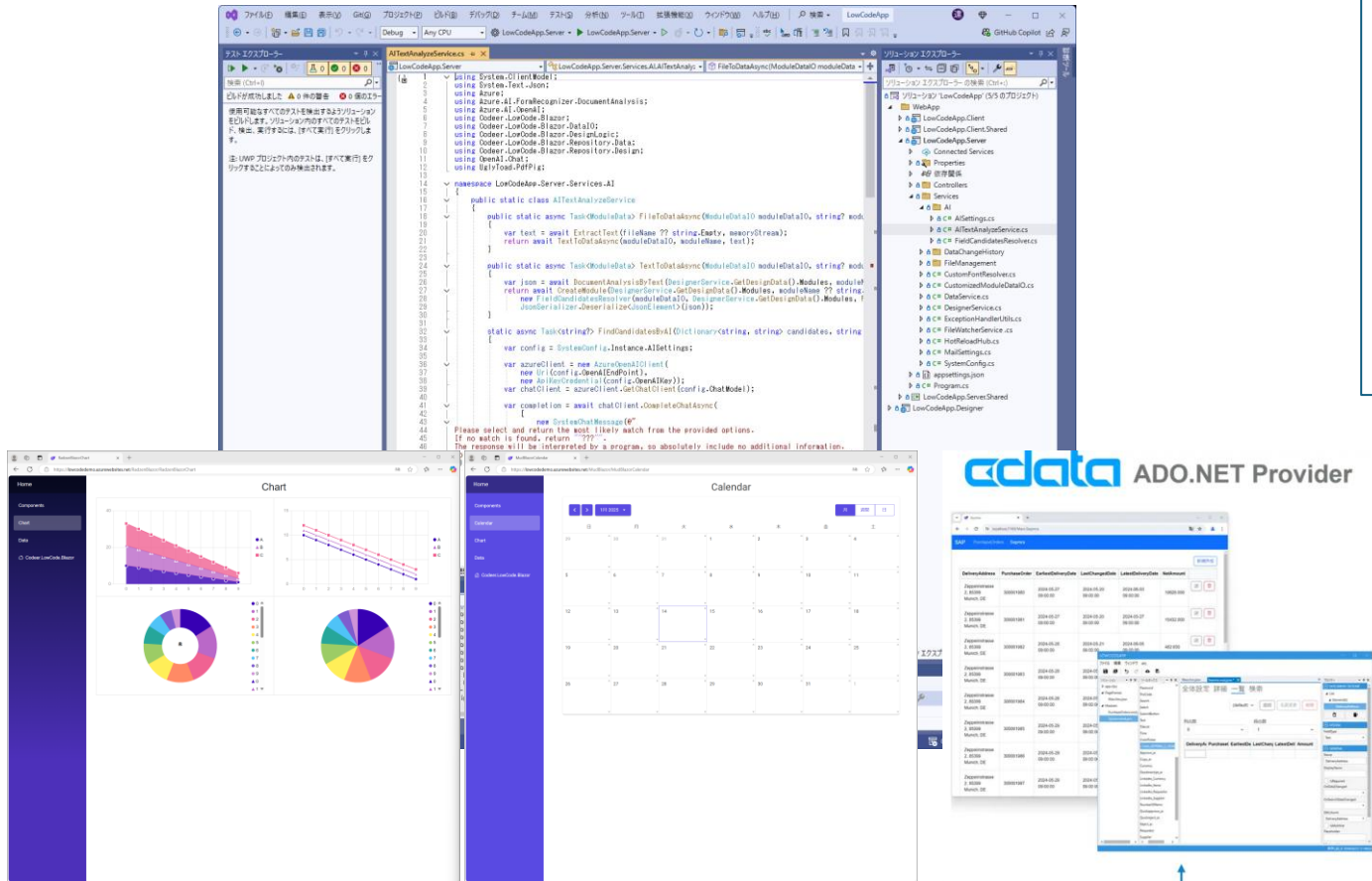
生成AIと連携することで、画像やPDFなどからデータを取り込んだり、開発の効率化を促すことができます。



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-

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



サンプルサイトURL：
[RadzenBlazorChart](#)

サンプルサイトURL：
[MudBlazorCalendar](#)

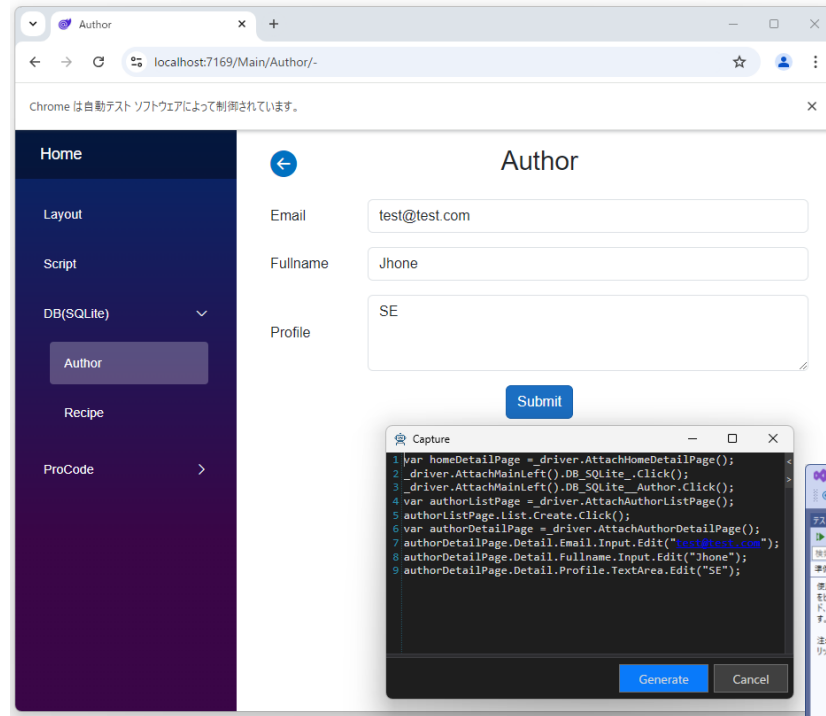
標準SQLでアクセス
<https://arc.cdata.com/jp>

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、3rd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

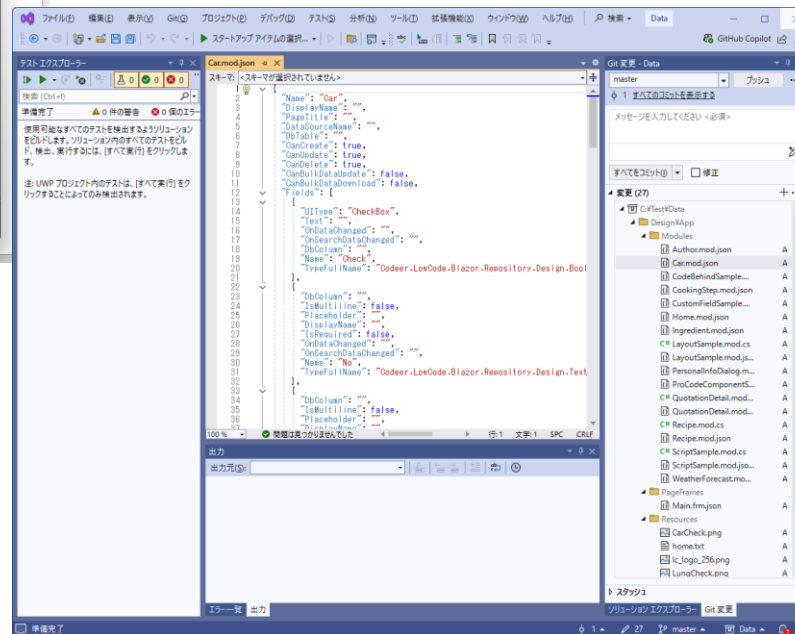
-自動テスト、バージョン管理-



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おこなえます。

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。

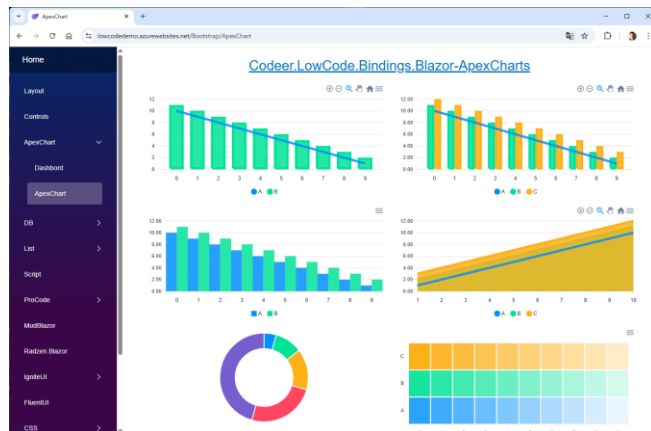


OSSとの連携

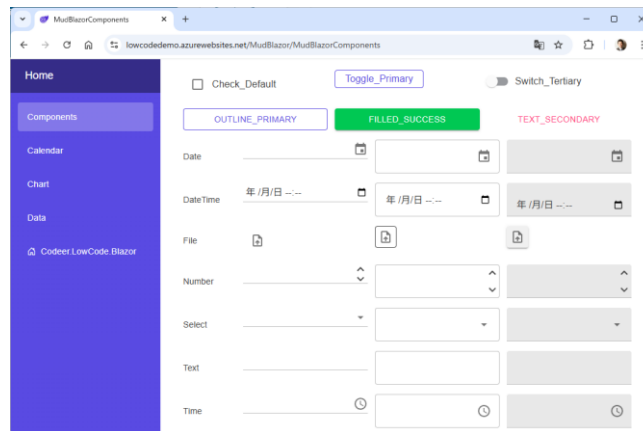
-BlazorのOSSのコンポーネントと連携-

MITライセンスで公開中

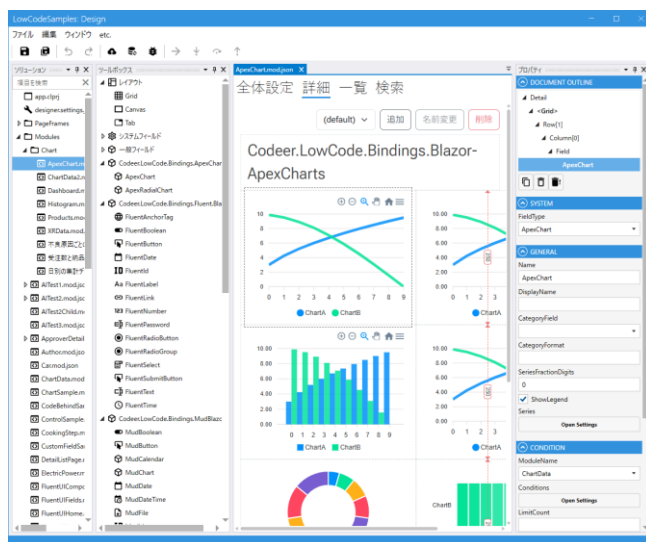
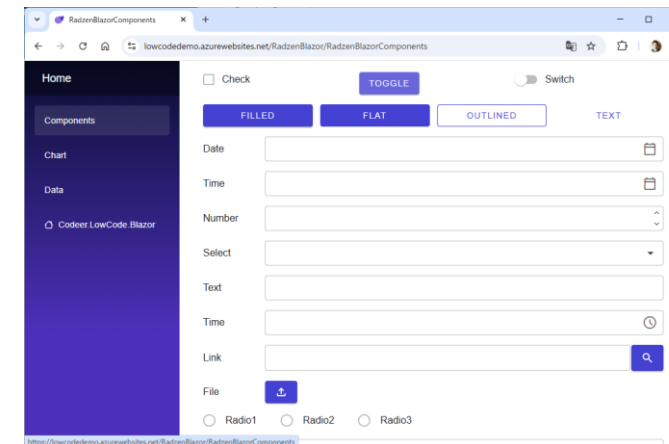
Codeer.LowCode.Bindings.
Blazor-ApexCharts



Codeer.LowCode.Bindings.
MudBlazor



Codeer.LowCode.Bindings.
Radzen.Blazor



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

デザイナーで画面に配置可能

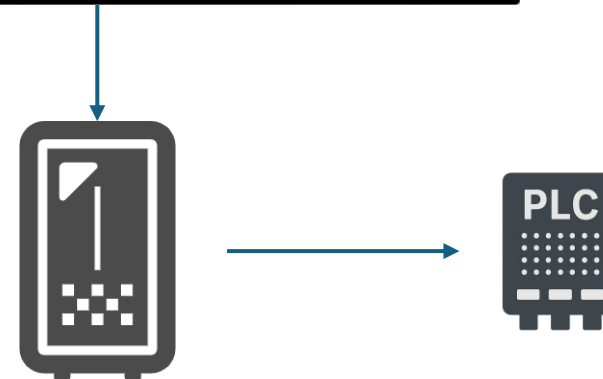
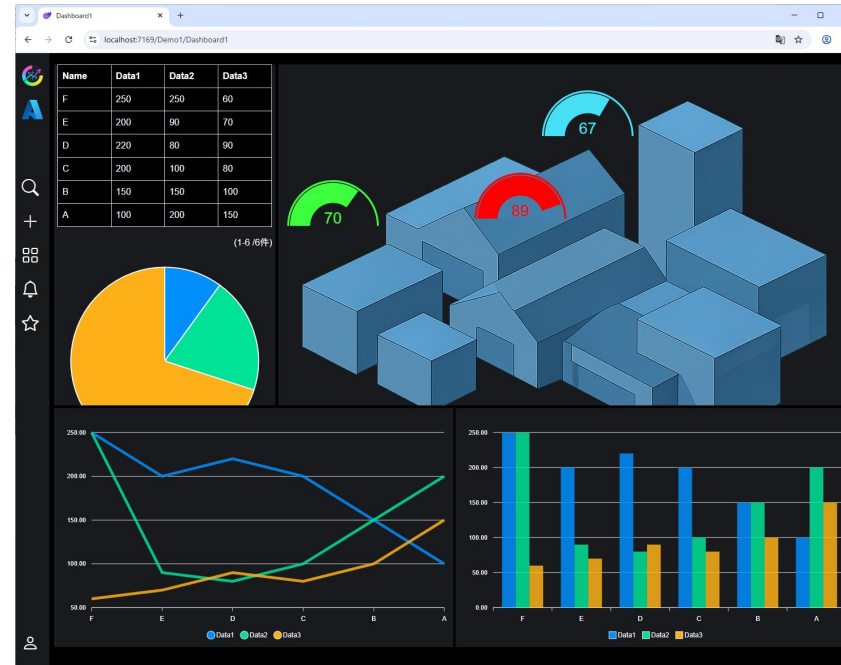
-組み込み機器の分野でも-

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。

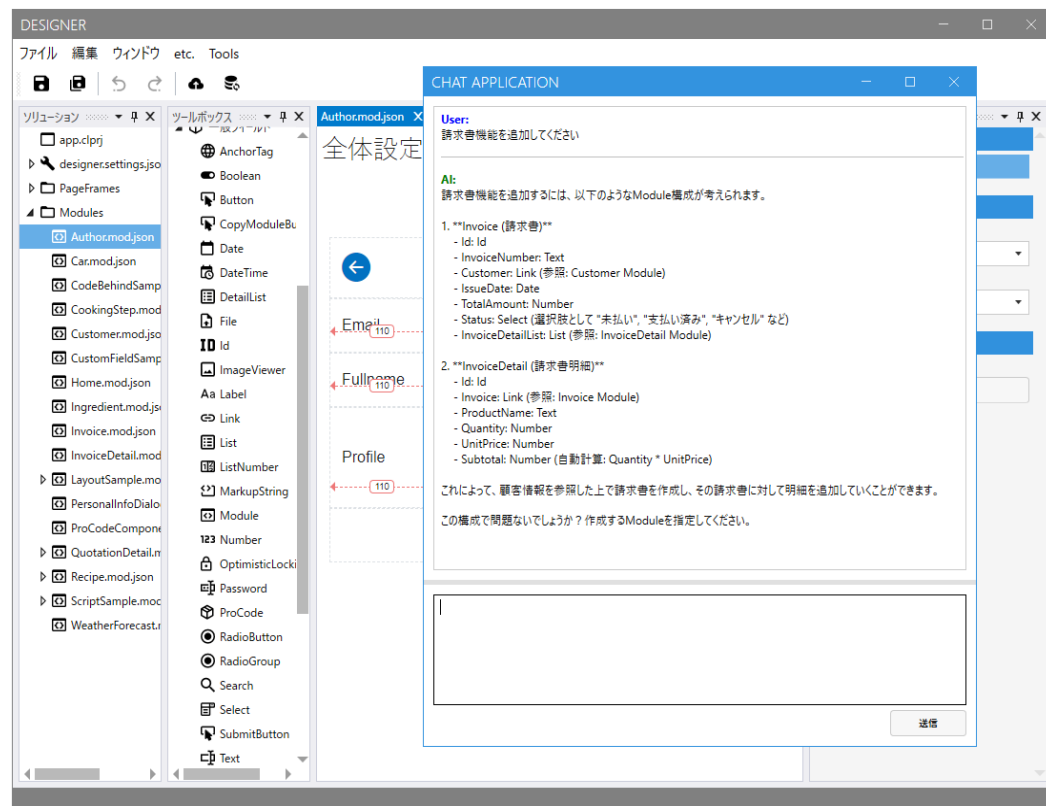
デンソーウェーブの社員さんとMITライセンスで開発中



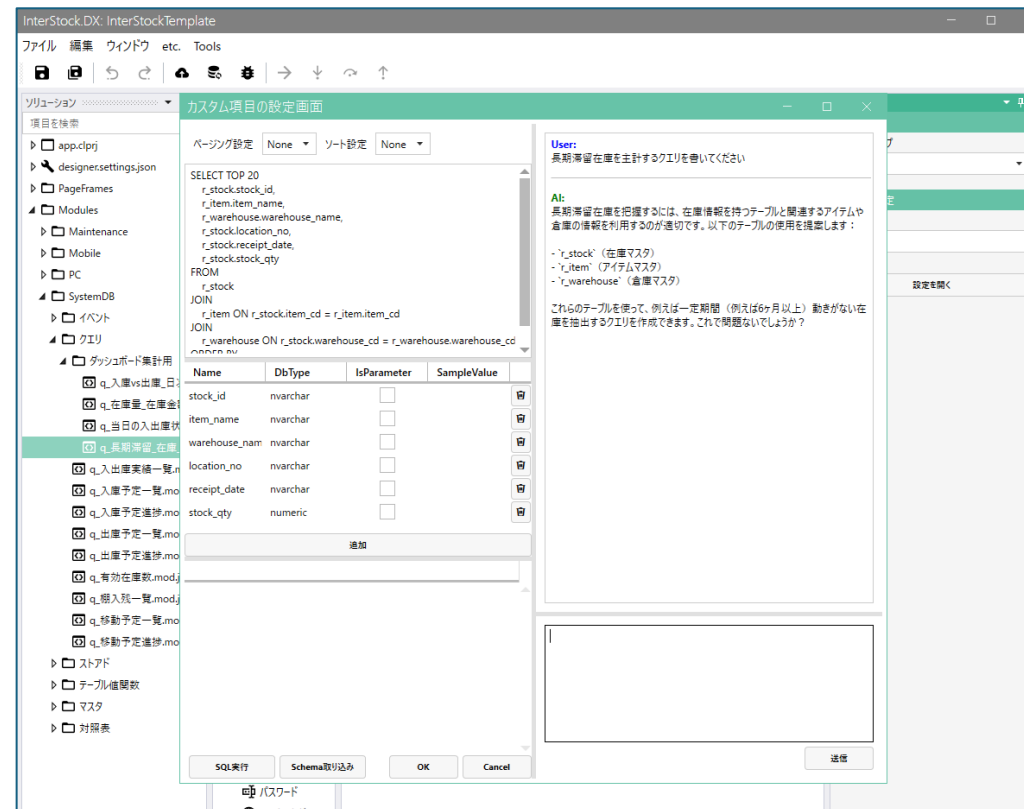
AIを使って開発

-AIで作成効率UP-

モジュールを作成



集計のクエリを作成



ロー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！

成長録

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**■ 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

1～2週間**■ 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

■ 展示会参加

製品の特長を学ぶ

デモ作成**■ 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

■ アウトプット

開発での気づきをもとにマニユア
ル作成やブラッシュアップを行う

案件**■ 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

■ 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

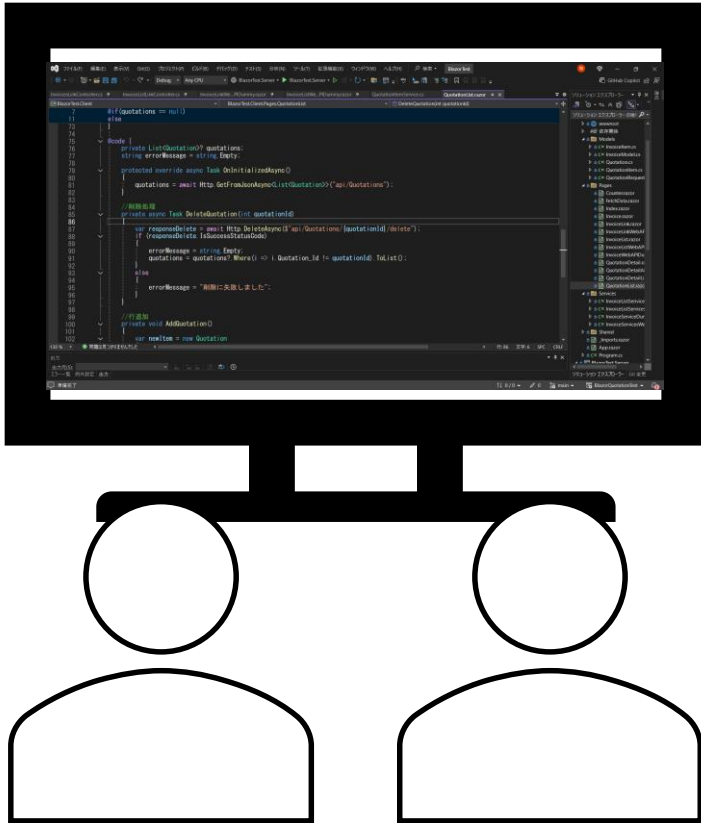
- ・ プロコードでうまくいかない時

■ 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

- ・ 考え方を学べる
- ・ うまくいかない時にどこを見るのか学べる
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 !