# Al@Work WG1 Alプロジェクトのためプロジェクトマネジメント研究

# 2024.10.8 Vol.4 プロジェクトマネジャーのための AI DrivenによるPJ管理ツール開発①

~ VS Code + Claude Devで生成AIによるPJ管理用WEBアプリを自動生成~

# VS Code + Claude Devで生成AIによるWEBアプリを自動生成



■生成AIによるアプリのノーコードによる自動生成の進化には激しいものがあります。

■ついにPromptにより、様々なWebアプリを自動生成 する実効的な仕組みが、VS Codeの拡張機能で、 Claude Devを利用することで可能になって来ました。

■VS Code + Claude Devの組合せで、その破壊的 な効果を確認することが可能です。

■我々は、この仕組みをPM手法のAl-Drivenによる 一つの解法として検討を進めます。

■DifyなどのRAGのワークフローと組合せることで、 経験知を蓄積しながら、WEB I/FでPJを効率化する 仕組み作りを狙います。

<sup>※</sup> DALL-E3で生成した『Claude Devによるアプリ自動生成のイメージ』



- 1. Claude Devとは?
- 2. VS Codeのインストールから、Claude Devの拡張機能設定まで
- 3. Gemini 1.5 FlashからAPIキーの取得
- 4. Claude Devに対するWebアプリの仕様の入力
- 5. Webアプリフロントエンドへの適用結果と、修正方法
- 6. バックエンドへのClaude Code適用の準備: Python Flaskの利用
- 7. Anthropic Claude 3.5 Sonnetのインストール
- 8. Anthropic Claude によるバックエンド向けPrompt自動生成
- 9. バックエンドへのClaude Code適用
- 10. まとめ
- 11. おわりに

### Claude Devとは? ✓ 生成AIによる最新のアプリ作成機能により、ノーコードでPJ管理用アプリを作成する

Claude Dev は、Anthropic の Build with Claude 2024 年 6 月八ッカソンの一環としてSaoud Rizwan によって開発された VS Code 拡張機能です。Claude 3.5 Sonnet のエージェント コーディング機能を活用して、 複雑なソフトウェア開発タスクを段階的に処理します。AI がファイルを読み書きしたり、プロジェクト全体を最初から 作成したり、ターミナル コマンドを実行したり (許可後) できるツールにより、Claude Dev は単純なコード補完や テクニカル サポートを超えた方法でユーザーを支援できます。

### Claude Devの概要と利用方法:

①MicrosoftのVS Codeの拡張機能に、Claude Devを追加します。

②利用したい生成AIを選択して、APIキーを連携させます。

③Claude Devのメニューを立上げて、タスク指定欄から、生成AI利用と同じスタイルで、Promptで作成したいWebアプリの仕様を入力します。

④ヒューマンインザループの形で、生成された アプリの結果に、修正したい内容をタスク入力します。

⑤グラフなどのUIも美しく出力出来ます。

⑥一度に生成出来ない場合は、逐次Step by Stepで機能を追加・修正出来ます。

※ 以下の試行は、個人の試行レベルであり、商用環境によるフローではないことにご注意下さい。

### VS Codeのインストール

Vs Codeをインストールして、次に拡張機能 🕒 から日本語化する

← → C s https://code.visualstudio.com/download	メ ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H) ← →	۶
🗙 Visual Studio Code Docs Updates Blog API Extensions FAQ GitHub Copilot 🛛 🖉 🔎 Search Docs	Docs 批張機能 ひ…	
Version 1.94 is now available! Read about the new features and fixes from September.	→ Marketplace で拡張機能を検索する 示 7	7
	✓ √ 1/2k-l\\`ĝø	5
Download Visual Studio Code	Soon	ms /
Free and built on open source. Integrated Git, debugging and extensions.	Saoud Rizwan	33 23
	Docker  Makes it easy to create manage and debug containerized applications	
	docker & Microsoft	
	Japanese Language Pack for Visual Studio Code	
<u>↓</u> Windows <u>↓</u> .deb <u>↓</u> .rpm <u>↓</u> Mac Windows 10 11     Debian Ubuntu     Red Hat Federa SUSE     maxOS 10.15+	ti v se ∲ Microsoft	3
User Installer x64 Arm64 .deb x64 Arm64 .zip Intel chip Apple silicon Universal	S 1590	ms
System         .rpm         x64         Arm54           Installer         x64         Arm64         CLI         Intel chip         Apple silicon           zip         x64         Arm64         Snap         Snap         Star         Star	Open any totaler in the windows subsystem for Linux (wsc) and take advantage of visual studio code s fun leature set.	3
CLI x54 Arm54 CLI x54 Arm54 Arm54	Dev Containers	
	Open any tolder of repository inside a Docker container and take advantage of visual studio code s full feature set.     Alicrosoft	<b>3</b>
By downloading and using Visual Studio Code, you agree to the <u>license terms</u> and <u>privacy statement</u> . Marketplace で拡張機能を検索する		
✓ √ 1/2トール済み		
Claude Dev Autonomous coding agent right in your IDE capable of creating/editing files executing commands	ands and more with your permission every	
Saoud Rizwan	◎ 続いて拡張機能からClaude Devをインストールする	
Docker Makes it easy to create manage and debug containerized applications	た側にロボット のメニューが踊れる	
docker Microsoft		
Japanese Language Pack for Visual Studio Code	これがClaude Devのメニュー選択ボタンになる	
Microsoft		
WSL Open any folder in the Windows Subsystem for Linux (MSL) and take advantage of Visual Studie Cos	<sup>© 159ms</sup> Claude Devで利用する生成AIを選んでAPIキー設定	
Open any rolder in the windows subsystem for Enux (wsc) and take advantage of visual studio cod     A Microsoft	(P.12)	
Dev Containers	Sendela full familium ant	
A Microsoft		
	(P.7以降を参照)	

5

### VS Codeで新しいフォルダを作成する

∢	ファイル(F) 編集(E)	選択(S)	表示(V)	移動(G)	実行(R)	ターミナル(T)	ヘルプ(H)
<b>ট</b> ,০	新しいテキスト ファ・ 新しいファイル… 新しいウィンドウ New Window wi	イル Ctrl th Profile	+Alt+Win Ctrl+	Ctrl+N dows+N +Shift+N >			[
မို	ファイルを開く フォルダーを開く		Ctrl+	Ctrl+O			
₽ ₽	ファイルでワークスペ 最近使用した項目	ペースを開く 目を開く		>	,		
⊑ <sub>€</sub>	フォルダーをワークス 名前を付けてワーク ワークスペースを複調	ペースに追; クスペースをf 製	加… 呆存…				
•	保存 名前を付けて保存		Ctrl	Ctrl+S +Shift+S			
	すべて保存  共有			Ctrl+K S			
	自動保存 ユーザー設定			>			
	ファイルを元に戻す エディターを閉じる フォルダーを閉じる ウィンドウを閉じる			Ctrl+F4 Ctrl+K F Alt+F4			
	終了						

#### 新しいフォルダを作成する

🗙 フォルダーを開く	刘 フォルダーを開く						
$\leftarrow \rightarrow \land \uparrow \square \rightarrow PC$	$\leftarrow \rightarrow \checkmark \uparrow$ - PC > $\vec{\pi}$ J = - $(E:)$ > 000_Udemy > 0_VS Code						
整理 ▼ 新しいフォルダー							
OneDrive - Person	名前 ^	更新日時	種類				
<ul> <li>✓ ■ PC</li> <li>⇒ 3D オブジェクト</li> <li>⇒ ダウンロード</li> <li>⇒ デスクトップ</li> </ul>	AI_Driven_PJ PJ_Cost_Calculation	2024/10/06 9:51 2024/10/06 12:11	ファイル フォルダー ファイル フォルダー				
> 🛗 ドキュメント							

# フロントエンドアプリの自動生成

### Google Gemini 1.5 FlashでAPIキーを取得する①





Google Al Studio Un	titled prompt 🥒				
c⇒ Get API key	System Instructions Optional tone and style instructions for the model				
<ul> <li>Greate new prompt</li> <li>New tuned model</li> <li>My library</li> <li>Talking to the Machine View all</li> <li>Prompt Gallery</li> </ul>					
Developer documentation       Developer forum       Developer forum       Gemini API for Enterprise	et started a sample prompt or add your own input below				
	Listing recipes using JSON schema Create JSON based on specified schema.	Recipe to JSON Create recipe in JSON mode using an image.	হিন্দু Scavenger Hunt Create a curated list of scavenger hunt concepts.		

ここでプロジェクトを作成していないと、それを要求してくるので、次の画面でプロジェクトを作成する

# プロジェクトを作成する①



#### Google Cloud 2

。 https://cloud.google.com、...、プロジェクトの作成と管理:

#### プロジェクトの作成と管理 | Resource Manager

#### Documentation 🥥

2024/08/30 — 新しいプロジェクトは、Google Cloud コンソール、Google Cloud CLI または projects.create() メソッドを使用して作成できます。 コンソール gcloud ... プロジェクトの作成・プロジェクトの識別・プロジェクトの一覧表示・プロジェクトの更新

	← → C thttps://cloud.google.cd	m/resource-manager/docs/creating-managing-projects?hl=ja	-	다 ☆ 🖪 효   팩 🌒 :
	Google Cloud F==X>F	テクノロジー分野 ▼ プロダクト横断型のツール ▼ 関連サイト ▼	Q. 検索	/ ト 日本語 マ コンソール : ①
	Resource Manager ガイド リフ	ー アレンス サポート リソース		お問い合わせ
	〒 フィルタ			このページの内容 準備
初めてGemini APIを実行する	Resource Manager プロダクトの概要	プロジェクトの作成と管理 ロー	この旧報は役に立ちましたが? ロークロ	プロジェクトの作成 プロジェクトの作成時にタグを追加する サービス アカウントを使用したプロジェクトの作成
場合は、プロジェクトを作成する 必要がある	クイックスタート すべてのクイックスタート 組織リソースを使用したクイックスタ ート 制約の適用のクイックスタート 入門ガイド すべての入門ガイド ▼ Resource Manager ↓ プロジェクトの管理 プロジェクトの管理 IAM を使用したプロジェクトの	Google Cloud プロジェクトは、API の管理、課金の有効化、共同編集者の追加と肖 権限の管理など、すべての Google Cloud サービスの作成、有効化、使用の基礎とす このページでは、Cloud Resource Manager API と Google Cloud コンソールを使用 成および管理する方法について説明します。	削除、Google Cloud リソースに対する なります。 引して Google Cloud プロジェクトを作 R階層の設定については、Google	フロジェクトの割り当ての管理 プロジェクトの識別 既存のプロジェクトを取得する プロジェクトの一覧表示 リソースのすべてのプロジェクトの一覧表示 プロジェクトを検索 プロジェクトの更新 プロジェクトの刺跡のトラブルシューティング プロジェクトの復元
	<ul> <li>IAM を使用したりエクトの</li> <li>IAM を使用したりエクトの</li> <li>IAM を使用したりエクトの</li> <li>IAM を使用したりエクトの</li> <li>IAM とりエクトの</li> <li>IAM とりエクトの</li> <li>IAM とりエクトの</li> <li>IAM とりエクトの</li> <li>IAM とりエクトの</li> <li>IAM になって</li> <li>IAM とりエクトの</li> <li>I</li></ul>	<ul> <li>プロジェクトの識別には次のものが使用されます。</li> <li>プロジェクト名: 人が読めるブロジェクト名。</li> <li>プロジェクト名は Google API では使用されません。プロジェクト名は、プロ も編集できます。プロジェクト名は一意である必要はありません。</li> <li>プロジェクト ID: グローバルに一意のプロジェクト ID。</li> <li>プロジェクト ID は、プロジェクトを Google Cloud の他のすべてのプロジェク 文字列です。プロジェクト Aを入力すると、Google Cloud コンソールは文字 意のプロジェクト ID を生成します。生成されたプロジェクト ID を使用する プロジェクト ID を生成します。生成されたプロジェクト ID を使用する プロジェクト ID には次の要件があります。</li> <li>6~30 文字にする必要があります。</li> </ul>	コジェクトの作成中や作成後にいつで クトと区別するために使用する一意の 、数字、ハイフンの組み合わせで一 ことをおすすめしますが、この ID は ると、プロジェクト ID は不変になり	

# プロジェクトを作成する②

#### 自分で指定するプロジェクト名を入力する



### Google Gemini Flash 1.5 でAPIキーを生成する②

#### APIキーを作成の欄で検索すると、先ほど指定した プロジェクト名が表示されるので、それを選択する



GeminiのAPIキーが生成された

注意) この状態で作成したAPIによる結果は Googleにオープンなので、企業の機密情報、 個人プライバシーの情報を利用する用途には 使用しないこと!!!

### Claude Dev にGemini Flash 1.5 で取得したAPIキーを設定

4	ファイル(F)	編集(E)	選択(S)	表示(V)	移動(G)	実行(R)	ターミナル(T)	ヘルプ(H)		$\leftarrow \rightarrow$			,
¢	CLAUD	e dev										+	9 Z 🕸
p	Setti	ngs											Done
0.	API P	rovider											
ξs	Goo	gle Gemin	i Y										
~	Gemi	ni API Key											
¢>		•••••		•••••	••								
_	This ke	ey is stored	locally and	d only used	to make A	9 requests	from this exte	nsion.					
-0	Mode	I											
<u>منہ</u>	gem	ini-1.5-fla	sh-002										$\sim$
•	✓ Sup Max o * Free	oports imag utput: 8,19 up to 15 rec	<b>ges</b> )2 tokens quests per i	minute. Afte	er that, billi	ng depends	on prompt size	s. For more info, s	see pricing deta	iils.			
	Custo	m Instruct	tions										
	#シ	ステムコマ	マンド										
	۰E	本語で必ず	ず応答して	こください	•								
	These	instruction	s are adde	d to the end	d of the sys	tem promp	ot sent with eve	ry request.					
		lways allo	ow read-o	nly operat	ions								
	When	enabled, C	laude will a	automatical	ly read files	, view direc	tories, and ins	pect sites withou	ut requiring you	ı to click th	e Allow button.		

# PromptでClaude Devに生成したいアプリの要求仕様を入力する

#### PJのメンバー毎の稼働単価と工数を管理する 可視化を含むWebアプリを作成してみる

ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H) ← →		
CLAUDE DEV       What can I do for you?	+ 3 [] 🛱	# 役割
Thanks to Claude 3.5 Sonnet's agentic coding capabilities, I can handle complex software development ta create & edit files, explore complex projects, and execute terminal commands (after you grant permission simple code completion or tech support.	sks step-by-step. With tools that let me ), I can assist you in ways that go beyond	あなたはプロの経験豊富なフロントエンコーダーでHTML/CSS/Javascriptが得意です。 次の什様書に従い、ソースコードを書いて下さい。
		# 亜农
) # 役割		
あなたはプロの経験豊富なフロントエンコーダーでHTML/CSS/Javascriptが得意です。… Tokens: 120,535 ↓6,142		・フロントエンドのみで動作するPJ稼働コスト管理表(日々の稼働メンバ単位での稼働時間、 稼働単価、日々のトータル稼働コスト)を開発して下さい。
View all history		日々のトータルコスト=Σ(日々のメンバ毎の稼働単価×稼働時間)とします。
		・ソースはHTML/Javascript/CSSで記載して、一つのファイルに纏めて下さい。
		・稼働時間は、時間(HR)で入力して、稼働単価は円貨で入力します。また日々のトータ ル稼働コストは、千円単位(K¥)で出力してください。
		・トータル稼働コストを、メンバ別に色分けした積み上げ棒グラフで日単位・月単位・年単位 で可視化してください。
		・また月単位で、円グラフで数値の多い順に、トータル稼働時間、トータル稼働コストを可視 化してください。
		・サンプルとして擬似データを作成し入力した状態としてください。
y Type your task here (@ to add context) ⊗ 0 ▲ 0 ₩ 0	6 >	勿論この先には、工数分析や様々な機能の拡張もStep by Stepで可能です (様々なUIの追加・改良、グラフの生成なども含めて)

### Claude Devを実行した結果

JU	ue Dev 毫天们 UIL 响不				TESTデータを含んだHtmlファイルが	
×	ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H)	$\leftrightarrow$ $\rightarrow$		:	生成されている!	00 – ø ×
D	エクスプローラー (Ctrl+Shift+E) - 1 つの未保存のファイル	+ 5 🛛 😂	◇ pj_cost.html: New File (Editable) ●			↑↓¶∞⊡…
2	✓ Task # 役割	×	◇ pj_cost.html > ♀ html 1-		+ html !+ <html></html>	
ಿ ಗೆ	あなたはプロの経験豊富なフロントエンコーダーでHTML/CSS/Javascriptが得意です。… Tokens: ↑ 4,682 ↓ 1,700 8	See more EXPORT			H <head> H <title>PJ稼働コスト管理表</title> H <style> H body { font-family: sans-serif; } H table { width: 100%; border-collapse: coli H table { borden: 10% solid ■ #ddd: podding</td><td>Lapse; }.</td></tr><tr><td>H Co</td><td></td><td></td><td></td><td>9 10 11 12</td><td><pre>H th { background-color: ■#640; paulin H th { background-color: ■#f2f2f2; } H canvas { width: 800px; height: 600px; } + </style> 2+ </head>	, opx, text-allgn, lert,
۲	日々の総稼働コスト: <span id="total-daily-cost">0</span> K¥ <canvas id="bar-chart"></canvas>			13 14 15	++ <body> ++ ++ <hi>PJ稼働コスト管理表</hi> -+</body>	
	<pre><script> const data = [ { date: '2024-07-29', member: 'A', hours: 8, rate: 10000 }, { date: '2024-07-29', member: 'B', hours: 6, rate: 12000 }, { date: '2024-07-30', member: 'A', hours: 7, rate: 10000 }, { date: '2024-07-30', member: 'B', hours: 8, rate: 12000 }, ];</pre></td><td></td><td></td><td>18 19 20 21 22 23 24</td><td><pre></td><td></td></tr><tr><td></td><td><pre>let dailyCost = 0; data.forEach(item => { dailyCost += item.hours * item.rate; }); document.getElementById('total-daily-cost').textContent = Math.round(</pre></td><td>dailyCost / 1000)</td><td></td><td></td><td><pre></td><td></td></tr><tr><td></td><td>// チャート描画処理は省略(Chart.jsなどを使用) </script></pre>			32	+ >+ >10000+ >10000+ >80000	
	 				+  +  +  + 2024-07-29	
	作成したHTMLファイルは、サンプルデータを含むPI稼働コスト管理表です。チャート描画処 が、Chart.jsなどのライブラリを使用すれば簡単に実装できます。	理は省略しています		37 38 39 40	+ >+ <	
	Ch. Claude wants to create a new file:			41 → 42	+	
	pj_cost.html			43	+ 2024-07-30	
8	Save Reject				+ 7(td) + 7(td) + 10000	
503		6 >		47	+ (ta>/0000	
×	⊗o∆o ₩o				)+	ペース: 4 UTF-8 CRLF HTML Q

これを一旦、Saveする

### Saveした後の画面

) 🤉	ファイノレ(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターナナル(T) ヘルプ(H) ← →	P Al_Driven_PJ	
Ç	CLAUDE DEV + S		· · · ·
ρ	✓ Task # 役割	<pre>     pj_cost.html &gt;</pre>	
202	あなたはブロの経験豊富なフロントエンコーダーでHTML/CSS/Javascriptが得意です。 See m Tokens: ↑ 4,682 ↓ 1,700 EXPC	3 <head> 4 <title>PJ稼働コスト管理表</title> 5 <style></style></head>	

# 作成したWEBアプリを動作させてみる

#### エクスプローラで開くと、作成したhtmlファイルが出来ている

∢	ファイル(F)	編集(E)	選択(S)	表示(V)	移動(G)	実行(R)	ターミナル	(T)	ヘルプ(H)	
Ð	エクスブ	0- <del>5</del> -								
~	✓ AI_DK	IVEN_PJ								
$\mathcal{Q}$	♦ pj_0	cost.h*	横に並べて	開く		Ctrl+Er	iter			
Ŷ٥			エクスプロー	ラーで表示す	する	Shift+Alt	t+R			
ځې			ファイルを開	くアプリケー	ションの選抜	兄				l
å			統合ターミ	ナルで開く						
00			比較対象(	の選択						
			タイムライン	を開く						
Ē			切り取り			Ctrl	l+X			
حفت			⊐ピ–			Ctrl	+C			
			ເຮັງທຸງປະ			chift , Alt				
å			//YMJC-			Shiit+Ait	+C			
6			相対バスを	שצ–	Ctrl+k	Ctrl+Shift	t+C			
			名前の変更	Ē			F2			
			削除			Del	ete			



#### 作成したhtmlファイルを開くと⇒以下:

← → C ③ 7ァイル file:///	_VS%20Code/AI_Driven_PJ/pj_cost.html	☆	А	Ď ∣	=ı J	<u>.</u>	D

#### PJ稼働コスト管理表

日付	メンバー	稼働時間(HR)	稼働単価(円)	日々の稼働コスト(円)				
2024-07-29	Α	8	10000	80000				
2024-07-29	В	6	12000	72000				
2024-07-30	А	7	10000	70000				
2024-07-30	В	8	12000	96000				
日々の総稼働コスト: 3	日々の総稼働コスト: 318 K¥							

#### WEB画面に、取り込んだサンプルデータが計算されて表示される!

# 作成したWEBアプリに入力機能を追加する



ל 🕅	アイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H)	$\leftrightarrow$ $\rightarrow$	,  P Al_Driven_PJ	
Дı	CLAUDE DEV	+ 🤉 🛛 🤀	♦ pj_cost.html ×	□ …
ے مو	〜 <b>Task</b> # 役割 あなたはプロの経験豊富なフロントエンコーダーでHTML/CSS/Javascriptが得意です。 <b>Tokens: ↑</b> 4,682 ↓ 1,700	X See more EXPORT	<pre>◆ pj_cost.html &gt; 1</pre>	5557002====== 5557002====== 1056== 107;
* - - •	]; let dailyCost = 0; data.forEach(item => { dailyCost += item.hours * item.rate; }); document.getElementById('total-daily-cost').textContent = Math.round(d // チャート描画処理は省略 (Chart.jsなどを使用) 	ailyCost / 1000)	<pre>7 table { width: 100%; border-collapse: collapse; } 8 th, td { border: 1px solid ■#ddd; padding: 8px; text-align: left; } 9 th { background-color: ■#f2f2f2; } 10 canvas { width: 800px; height: 600px; } 11  12  13 <body> 14 15 <h1>PJ稼働コスト管理表</h1> 16 17  18</body></pre>	and the second s
	 作成したHTMLファイルは、サンプルデータを含むPJ稼働コスト管理表です。チャート描画処理 が、Chart.jsなどのライブラリを使用すれば簡単に実装できます。 C+ Claude wants to create a new file:	は省略しています	19      20 <th< td=""><td></td></th<>	
	<ul> <li>p_cost.html</li> <li>✓ Task Completed</li> <li>PI稼働コスト管理表を作成しました。pj_cost.html ファイルを確認してください。このフ: データを含む表と、チャート描画のためのキャンパス要素が含まれています。チャートの描画 ライブラリを使用する必要があります。</li> <li>▷ Claude wants to execute this command:</li> </ul>	∨ マイルには、サンブル には、Chartjsなどの	27        28      29     20       30     20       31     20       32     20       33     20       34        35      27	
	start pj_cost.html メンバー毎に日々の稼働メンバ単位での稼働時間、稼働単価を入力できる機能を追加してくた "INPUTボタンを新規に作成して、このボタンを押すと、上記の情報が入力できるようにして また"Save"ボタンを作成して、入力した情報がSave出来るようにして下さい。 、API Request	ださい。 ください。 〜	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
8 8			44	
× 8	0 <b>∆</b> 0 ₩ 0		49	1、列 1 スペース: 4 UTF-8 CRLF HTML Q
-	🔎 टटに入力して検索 🛛 💼 📑 📭 🕰 👢 🗴	<i>a</i> 🔊 🖸	🔫 🔿 🞯 🥂 🟹 🛜 S 🖬 🛛 🥥 🚍 😰 👀 💽 刘 🖂 🖟 🚛	へ <i>í</i> c

### Claude Devの実行結果

#### 結果を確認しながら、Step By Step、ヒューマンインザループでアプリのブラッシュアップが出来る!

CLAUDE DEV + ① [2 ②	
<pre>&gt; v Task</pre>	
rate: parseInt(rateInput.value),       7       table { width: 100%; border-collapse: collapse; }       8       th, 10         width: 2000; import the part the	!DOCTYPE html> html> head> title>PJ稼働コスト管理表 style> ody { font-family: sans-serif; }
10ctation age: settlem( p)_cost_uata , source ingriv(uata)),       12        15        16        16          11        chi>pinction updateTotalCost() {       14       17         12        15        16        16          13        chi>pinction updateTotalCost() {       18          14        17         15        chi>pinction updateTotalCost() {       18          16        19       10         17        table>       20          18        ctable>       20	<pre>able { width: 100%; border-collapse: collapse; } h, td { border: 1px solid ■#ddd; padding: 8px; text-align: left; } h { background-color: ■#f2f2f2; } anvas { width: 800px; height: 600px; } input-form { margin-top: 20px; } input-form label { display: block; margin-bottom: 5px; } input-form input { width: 100%; padding: 8px; margin-bottom: 10px; } /style&gt;</pre>
if if is a const storedData = localStorage.getItem('pj_cost_data');       19       (tr>       20       (th)日付       23         21       (th)日付       24         22       (th)日付       24         23       (th)日付       24         24       (th)日付       25         25       (th)日付       26         26       (th)日付       27         27       (th)日付       27         28       (th)日付       27         29       (th)日本の市       27         27       (th)日本の市       28         28       (th)日本の市       27         29       (th)日本の市       28         29       (th)日本の市       29         27       (thod>)       27         27       (thod>)       30+	<pre>/head&gt; body&gt; h1&gt;PJ稼働コスト管理表 table&gt;  <thead>   &gt;日付   &gt;日付   &gt;日付   &gt;日付   &gt;和働用   &gt;和働用   &gt;和働用   &gt;和   &gt;     &lt;</thead></pre>
<pre>}); updateTotalCost(); } // チャート描画処理は省略(Chart.jsなどを使用)  // ケート描画処理は省略(Chart.jsなどを使用)  // ケート描画処理は省略(Chart.jsなどを使用)  // ケート描画処理は省略(Chart.jsなどを使用)  // Claude wants to edit this file: // Claude wants to edit this file: //</pre>	
pj_cost.html     39-	
Save     Keject       Type a message (@ to add context)     Image: Context in the sector of the	

追加した機能の部分

#### またSaveする

#### AI@Work Community of Practice, PMI Japan 2024 - Copyright and all rights reserved.

# バックエンドアプリの自動生成

### Python Flaskとは?

Flask(フラスク)は、プログラミング言語Python用の、軽量なウェブアプリケーションフレームワークである。 標準で提供する機能を最小限に保っているため、自身を「マイクロフレームワーク」と呼んでいる。 Werkzeug WSGIツールキットとJinja2テンプレートエンジンを基に作られています。

フレームワークとは、動的なウェブサイト、ウェブアプリケーション、Webサービスの開発をサポートするために設計されたデータベースへのアクセスのためのライブラリやテンプレートエンジン(Webテンプレートも参照)、 セッション管理を提供し、コードの再利用を促進させる仕組みから構成されます。

大規模開発向けにはDjango、小規模で軽量な開発ではFlaskがメジャーであり、開発の効率化に寄与します。

バックエンドアプリの自動生成では、Python Flaskのフレームワークを活用します。

# Pythonのインストール

Python	のインスト・	ール					Windowsの環境変数に、 exeファイルを対応させて、	Pythonの コマンドプロンプ
← → C ≒ https://www.pytho	+ on.org/downloads/					- □ × ■ ☆ ☑ □ = ● = ●	トからすぐにPythonが実行	テ出来て、また
	Python	PSF	Docs	РуРІ	Jobs Community		特権セードが使用できるよ	つにして、
	🥏 pytl	non™		Donate Search	60 Socialize		(※様々な機能追加用]	にゅる コマンド)
		About Downloads	Documentation Comr	nunity Success Stories N	News Events	🛸 Python 3.12 🛛 (64-bit) Se	tup	- 🗆 🗙
	Dov Dow Lookin Linux/ Want t Docker	vnload the lates nload Python 3,12.7 gror Hytnon with a different ( UNIX, macOS, Other o help test development versi r images	<b>S</b> ? Python for <u>Windows</u> , ons of Python 3.13? <u>Prereleases</u> ,	s contractions			Install Python 3.12. (64-b) Select Install Now to install Python with default se Customize to enable or disable features.	t) ttings, or choose
	Active Python For more information	Releases on visit the Python Developer'	s Guide.				→ Customize installation Choose location and features	
	Python version	Maintenance status	First released	End of support	Release schedule	python	Use admin privileges when installing py.exe	
	3.13	prerelease	2024-10-07 (planned)	2029-10	PEP 719	windows	Add python.exe to PATH	Cancel
	3.11	security	2022-10-02	2027-10	PEP 664	WITIOOWS	la l	
	3.10	security	2021-10-04	2026-10	PEP 619			
	3.9	security	2020-10-05	2025-10	<sub>PEF</sub> 🔤 コマンド プロンプト - pyt	thon		- 🗆 × 🛛
	3.8	security	2019-10-14	2024-10	PEEMicrosoft Windows (c) Microsoft Cone C:¥Users¥DELL>pyth	[Version 10.0.19045.4894] poration. All rights reserved		<b>^</b>
					Type "help", "coo >>> print("Hello", Hello >>>	gs/vs.12.7:ubusead, oct 120 yright", "credits" or "licens )	e for more information.	j on win₃z
	Al@Work Com	munity of Practice, I	PMI Japan 2024 - Copyri	ght and all rights reserved.				× 21

### Anthropic Claude 3.5 Sonnetのインストール

フロントエンドの設計に続いて、バックエンドの設計も必要になる。ここではGemini Flash 1.5より高性能の生成AIが必要になるので、Anthropic Claude 3.5 Sonnetを利用します。APIキー取得の手順も示します。

#### https://www.anthropic.com/api

25 https://www.anthropic.com/api		25 https://console.anthropic.com/login?returnTo=%2F%3F	🔄 🌣
ANTHROPIC ANTHROPIC Build with Claude Create user-facing experiences, new products, and new ways to work with the most advanced AI models on the	Claude ~ Research Company C	https://console.anthropic.com/login?returnTo=%2F%3F ANTHROP\C Build with the Anthropic AP C Continue with Google OR	≥ >
Start building Developer docs		name@yourcompany.com Continue with email By continuing, you agree to Anthropic's <u>Commercial Terms</u> and <u>Usage Policy</u> , and acknowledge their <u>Privacy Policy</u> . Googleで認証	

# Anthropic Claude 3.5 SonnetでのAPIキー取得① プリペイドでチャージ



Al@Work Community of Practice, PMI Japan 2024 - Copyright and all rights reserved.

### Anthropic Claude 3.5 SonnetでのAPIキー取得②



### Anthropic Claude 3.5 Sonnetで取得したAPIキーを設定

4	ファイル(F)	編集(E)	選択(S)	表示(V)	移動(G)	実行(R)	ターミナル(T)	ヘルプ(H)	$\leftarrow \ \rightarrow$	
Ð	CLAU	DE DEV								+ 9 [] 🛱
ρ	Sett	ings								Done
	API P	Provider								
гJ	Ant	hropic	$\sim$							
~	Anth	ropic API Ke	⊧y							
₫>			•••••	•••••	•••••	•••••		•••••	 	
_		Use custom	base URL							
-0	This k	ey is stored l	ocally and	only used	to make A	PI requests	s from this exte	nsion.		
.ni	Mode	el								
	clau	ıde-3-5-son	net-2024	0620						~
<b>.</b>	✓ Su ✓ Su Max o Input Cache Outpu	pports image pports prom output: 8,192 price: \$3.00/ e writes price e reads price: ut price: \$15.	es pt caching tokens (million to \$3.75/mil \$0.30/million	<b>g</b> kens illion token illion tokens i tokens	15 5					
	Custo	om Instructi	ons							
	#シ	ィステムコマ	ンド							
	۰E	日本語で必す	『応答して	ください						
	These	instructions	are addec	to the end	d of the sys	tem prom	pt sent with ev	ery request.		
		Always allow	w read-or	nly operat	tions					
	14/1		م النبية مامين		he read file.		starias and in	in a starita suiti	 au ta aliala	

Al@Work Community of Practice, PMI Japan 2024 - Copyright and all rights reserved.

# VS CodeのClaude Devの画面でバックエンドタスクを割り当てる

Al@Work Community of Practice, PMI Japan 2024 - Copyright and all rights reserved.

כ 😒	rイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H) ← →	₽ PJ_Cost_Calculation	
Ф	claude dev + ೨ [] ₿	♦ index.html ×	II
2 <u>%</u> 入	What can I do for you? Thanks to Claude 3.5 Sonnet's agentic coding capabilities, I can handle complex software development tasks step-by-step. With tools that let me create & edit files, explore complex projects, and execute terminal commands (after you grant permission), I can assist you in ways that go beyond simple code completion or tech support.	<ul> <li>index.html &gt; ② html &gt; ② body &gt; ③ script</li> <li>2 <html></html></li> <li>14 <body></body></li> <li>32 <div></div></li> <li>33 <htps: div=""></htps:></li> <li>34 <convas id="daily-chart"></convas></li> <li>35 <!--/li--> </li></ul>	The second secon
¤′ ⊑⊚	oc RECENT TASKS OCTOBER 6, 12:34 PM # 役割 あなたはプロの経験豊富なフロントエンコーダーでHTML/CSS/Javascriptが得意です。 Tokens: 113,189 47,771	36 37 <div> 38   <h2>月別コスト</h2> 39   <canvas id="monthly-chart"></canvas> 40 </div> 41 42 <div></div>	
•	OCTOBER 6, 12:05 PM # 役割 あなたはプロの経験豊富なフロントエンコーダーでHTML/CSS/Javascriptが得意です。 Tokens: t20,535 46,142	43 <h2>年別コスト</h2> 44 <canvas id="yearly-chart"></canvas> 45  46 47 <div> 48 <h2>月別コスト内訳(円グラフ)</h2> 49 <canvas id="monthly-pie-chart-cost"></canvas> 59 <canvas id="monthly-nie-chart-bour"></canvas> 59</div>	
	View all history	<pre>51 <!--/div--> 52 53 <script></script></pre>	

# バックエンドアプリ生成用のタスクのPrompt

バックエンドアプリを生成するPromptは、機能コンポーネント毎に丁寧に他との関係性を整合とる必要があるため、正確性が要求されてかつ複雑になる。 これをAnthropicのPrompt生成機能により実現する。以下は、その前段階として、AnthropicのPrompt生成機能に要求するタスクを伝えるためのPrompt。



### Anthropic ClaudeでPromptを自動生成する



You are tasked with converting an HTML file (index.html) into a Flask web application using SQLite for database operations. The application should display the content of index.html when accessed at the root URL ("/") and allow editing and saving of the table data using SQLite.

Here's the content of the original index.html file:

<index\_html> {{INDEX\_HTML}} </index\_html>

To create the Flask application with SQLite integration, follow these steps:

- 1. Create a new directory for your project and set up the following structure:
  - app.py
  - templates/
  - index.html
  - instance/
  - database.db (This will be created automatically by Flask)
- 2. Set up the SQLite database:
  - In app.py, use Flask-SQLAlchemy to define a model for the table data
  - Create functions to initialize the database and add sample data
- 3. Modify app.py to include the following:
  - Import necessary modules (Flask, Flask-SQLAlchemy, etc.)
  - Configure the Flask application and SQLAlchemy
  - Define routes for displaying, adding, editing, and deleting data
  - Implement CRUD operations using SQLAlchemy

# Anthropic Claudeで自動生成されたPrompt②

- 4. Modify templates/index.html:
  - Add form elements for adding and editing data
  - Include JavaScript for handling AJAX requests to update the table dynamically
- 5. Here's a basic structure for app.py:

```python from flask import Flask, render\_template, request, jsonify from flask\_sqlalchemy import SQLAlchemy from datetime import datetime

app = Flask(\_\_name\_\_)
app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///database.db'
db = SQLAlchemy(app)

class TableData(db.Model):

id = db.Column(db.Integer, primary\_key=True)
date = db.Column(db.Date, nullable=False)
member = db.Column(db.String(100), nullable=False)
unit\_price = db.Column(db.Integer, nullable=False)
work\_hours = db.Column(db.Integer, nullable=False)

@app.route('/')
def index():
 data = TableData.query.all()
 return render\_template('index.html', data=data)

# Anthropic Claudeで自動生成されたPrompt③

```
@app.route('/add', methods=['POST'])
def add data():
   # Implement adding new data
  pass
@app.route('/edit', methods=['POST'])
def edit data():
   # Implement editing existing data
  pass
@app.route('/delete', methods=['POST'])
def delete data():
   # Implement deleting data
  pass
if __name__ == '__main__':
  with app.app_context():
     db.create all()
(app.run(debug=True)
```

# Anthropic Claudeで自動生成されたPrompt④

```
6. Modify templates/index.html to include forms and JavaScript for AJAX requests. Here's a basic structure:
```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Data Table</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
  <h1>Data Table</h1>
  <!-- Table headers and data rows -->
  <form id="add-form">
     <!-- Form fields for adding new data -->
  </form>
  <script>
    // JavaScript for handling AJAX requests
  </script>
</body>
</html>
. . .
```

## Anthropic Claudeで自動生成されたPrompt⑤

7. To set up the required modules, run the following commands in your terminal:

• • •

pip install flask flask-sqlalchemy

- 8. To run the application:
  - Open a terminal in your project directory
  - Run the command: `python app.py`
  - Open a web browser and navigate to `http://localhost:5000`

Remember to implement proper error handling, input validation, and security measures in your final application. This basic structure provides a starting point for creating a Flask web application with SQLite integration based on the given index.html file.

# Claude Devの画面でバックエンドタスクを割り当てる①

**×** ₽

ુટ

1

۲

8

ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) タ-	-ミナル(T) ヘルプ(H) ← →	ې	O PJ_Cost_Calculation	
CLAUDE DEV	+ :	9 Z 🕸	◊ index.html ×	□ …
What can I do for you? Thanks to Claude 3.5 Sonnet's agentic coding capabilities, I or create & edit files, explore complex projects, and execute ter simple code completion or tech support.	can handle complex software development tasks step-by-step. With tools th minal commands (after you grant permission), I can assist you in ways that g	at let me go beyond	index.html > ② html > ③ body > ③ script     2	
Anthropic Claudeで自 ただし冒頭に、フロントエン をメンション(@を付けて @index.html	自動生成したPromptを貼る ンジンで作成した @index.html )追加しておくこと!		57 (UIV) 38 〈h2>月別コスト〈/h2> 39 〈canvas id="monthly-chart"> 40 〈/div> 41 42 〈div> 43 〈h2>年別コスト〈/h2> 44 〈canvas id="yearly-chart"> 45 〈/div> 46 〈div>	
You are tasked with (index.html) into a l SQLite for database should display the c accessed at the root and saving of the ta	converting an HTML file Flask web application using operations. The application content of index.html when t URL ("/") and allow editing able data using SQLite.		<pre>48</pre>	Labora (USPARA) The second secon
Here's the content of <index_html> {{INDEX_HTML}} </index_html>	of the original index.html file: つづく・・・・		<pre>{ date: 2024/03-04; member: b; hours: 7, fate: 12000 }; { date: '2024-03-05', member: 'A', hours: 6, rate: 10000 }, { date: '2024-03-05', member: 'C', hours: 5, rate: 15000 }, ]; function renderTable(data) { const tableBody = document.getElementById('data-table'); tableBody.innerHTML = ''; //既存データのクリア data.forEach(item =&gt; { const row = tableBody.insertRow(); row.insertCell().textContent = item.date; row.insertCell().textContent = item.nember; row.insertCell().textContent = item.nember; row.insertCell().textContent = item.nember; row.insertCell().textContent = item.nember; row.insertCell().textContent = item.nete; row.insertCell().textContent = item.neter().textContent = item.neter().textContent = item.ne</pre>	
No A o ₩o			77 }); 78 }	
			1] 104, 7]	

### Claude Devの画面でバックエンドタスクを割り当てる②

>	ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H) ← →	*
Q	CLAUDE DEV + 3 🖸 🏟	Ċ
р	✓ Task ×	Q
မှ	You are tasked with converting an HTML file (index.html) into a Flask web application using SQLite for database operations. The applica See more	မိုစ
\$ ¢	Iokens: ↑ 4 ↓ 219 Cache: 🖯 +6,011 → 0 API Cost: \$0.0258	
Ē	✓ API Request \$0.0258	Ē
<b>.</b>	タスクを分析し、必要な手順を整理します:	<b></b>
۲	1. プロジェクト構造の作成 2. app.pyの作成とSQLiteデータベースのセットアップ 3. templates/index.htmlの修正 4. 必要なモジュールのインストール	ė
	まず、プロジェクト構造を作成するために、list_filesツールを使用して現在のディレクトリ構造を確認し、必要なディレクトリとファイルを作成 します。	
	Claude wants to recursively view all files in this directory:	
	PJ_Cost_Calculation ~	
	タスクを分析し、必要な手順を整理します:	
	フロンエクト構造の作成 app pyの作成とSOLiteデータベースのセットアップ	
	templates/index.htmlの修正	
	必要なモジュールのインストール	
	ま9、ノロンエクト傾這を作成9るにのに、IISt_fileSツールを使用しし現在のテイレクトリ 構造を確認し、必要なディレクトリとファイルを作成します。	
8	Approve Reject	8
572	Type a message (@ to add context)	503

ファイル(F)	編集(E)	選択(S)	表示(V)	移動(G)	実行(R)	ターミナル(T)	ヘルプ(H)		$\leftrightarrow \rightarrow$		
CLAU	DE DEV										+ ") 🛛 🍪
~ 1 @in	<b>Task</b> dex.html										×
You Tok Cac	are tasked ens: ↑8 ↓ he: Fl +6.2	with conv 424 83 → 6.01	verting an I	HTML file (	index.htm	l) into a Flask	web appl	cation using SQL	ite for datal	base operations. The	applica See more
API	Cost: \$0.0	317									EXPORT
~ A	PI Reques	t \$0.0258	3								~
タスク	を分析し、	必要な手	順を整理	します:							
1. プ 2. ap 3. ter 4. 必	1. プロジェクト構造の作成 2. app.pyの作成とSQLiteデータベースのセットアップ 3. templates/index.htmlの修正 4. 必要なモジュールのインストール										
まず、 します	プロジェク 。	クト構造を	作成する	こめに、 lis	t_filesツー	・ルを使用して	て現在のデ	ィレクトリ構造を	を確認し、必	&要なディレクトリ&	とファイルを作成
BO	laude wan	ts to recu	ırsively vie	w all files	in this di	rectory:					
PJ_C	ost_Calcula	ation									
~ A	PI Reques	t \$0.0059	•								
現在の	ディレク	トリ構造を	確認しま	した。必要	なディレ	クトリとファ	イルを作品	成する必要があり	ます。以下	の手順で進めます:	
1. ter 2. ap 3. ap 4. ter	nplatesディ p.pyファイ p.pyの内容 nplates/inc	ィレクトリ ルを作成 を実装 dex.htmlの	「を作成し、 の内容を修正	その中に E	index.htm	を移動					
まず、	templates	ディレク	トリを作成	し、index.	htmlを移	動させます。					
ΣC	laude wan	ts to exec	cute this c	ommand:							
mkd	ir templ	ates &&	move in	dex.html	templa	tes\					
			Run Cor	mmand						Reject	
		10.1									

# Claude Devの画面でバックエンドタスクを割り当てる③

エラー発生:

ל 🕅	アイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H)	P PJ_Cost_Calculation	
Ŋ	CLAUDE DEV + 3 🖸 🍪	◊ index.html ×	□ …
	✓ Task × @index.html	<ul> <li>o index.html &gt; ⊘ html &gt; ⊘ body &gt; ⊘ script</li> <li>2 <html></html></li> <li>14 <body></body></li> <li>32 <div></div></li> </ul>	
¢ ∽	You are tasked with converting an HTML file (index.html) into a Flask web application using SQLite for database operations. The applice See more Tokens: ↑ 12 ↓ 553 Cache: ⊕ +6,708 → 12,294 API Cost: \$0.0372 EXPORT	33 <hr/> 42>日別コスト 34 <canvas id="daily-chart"></canvas> 35  36 36 37 36 36 37 36 36 37 36 37 36 37 36 37 37 37 38 38 38 38 38 38 38 38 38 38 38 38 38	Jertonani Jertonani Jertonani Jertonani Jertonani Jertonani Jertonani Jertonani
_⊚ 	1. プロジェクト構造の作成 2. app.pyの作成とSQLiteデータベースのセットアップ 3. templates/index.htmlの修正 4. 必要なモジュールのインストール	37 <ul> <li><ul> <li><ul></ul></li></ul></li></ul></li></ul></li></ul></li></ul></li></ul></li></ul></li></ul></li></ul></li></ul></li></ul></li></ul>	
۲	まず、プロジェクト構造を作成するために、list_filesツールを使用して現在のディレクトリ構造を確認し、必要なディレクトリとファイルを作成 します。	43 <hr/> 43 <hr/> 44 <canvas id="yearly-chart"></canvas> 45  46 46 47 48 48 48 48 48 48 48 48 48 48 48 48 48	Notes
	PJ_Cost_Calculation ✓ ✓ API Request (\$0.0059) ✓	47 <div> 48 <div> 49 <canvas id="monthly-pie-chart-cost"></canvas> 50 <canvas id="monthly-pie-chart-hour"></canvas> 51 </div></div>	URAN PERSON AND A CONTRACT OF
	現在のディレクトリ構造を確認しました。必要なディレクトリとファイルを作成する必要があります。以下の手順で進めます: <ol> <li>templatesディレクトリを作成し、その中にindex.htmlを移動</li> <li>app.pyファイルを作成</li> <li>app.pyの内容を実装</li> <li>templatesディレクトリを作成し、index.htmlを移動させます。</li> </ol>	<pre>52 53 <script></script></pre>	

36

# Claude Devの画面でバックエンドタスクを割り当てる④

4	ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H) ← →									
¢	CLAUDE DEV + 3	[2]								
ρ	✓ Task @index.html									
₽ ₽	You are tasked with converting an HTML file (index.html) into a Flask web application using SQLite for database operations. The applical Se <b>Tokens:</b> $\uparrow$ 16 $\downarrow$ 660 <b>Cache:</b> $\bigcirc$ +7,016 $\rightarrow$ 19,002 API <b>Cache:</b> $\bigcirc$ +7,016 $\Rightarrow$ 19,002	e mo								
<u>_</u> @										
	✓ API Request \$0.0059									
٢	現在のディレクトリ構造を確認しました。必要なディレクトリとファイルを作成する必要があります。以下の手順で進めます: 1. templatesディレクトリを作成し、その中にindex.htmlを移動 2. app.pyの内容を実装 3. app.pyの内容を実装 4. templates/index.htmlの内容を修正 まず、templatesディレクトリを作成し、index.htmlを移動させます。 2. Claude wants to execute this command:									
	mkdir templates && move index.html templates\									
	<ul> <li>✓ API Request \$0.0054</li> <li>         申し訳ありません。Windows環境でのコマンド実行に問題がありました。コマンドを分割して実行し直します。まず、templatesディレクトリを 作成し、次にindex.htmlを移動させます。     </li> <li>         Claude wants to execute this command:     </li> </ul>									
	mkdir templates > Command Output									
	<ul> <li>✓ API Request 50.0048</li> <li>templatesディレクトリが正常に作成されました。次に、index.htmlファイルをtemplatesディレクトリに移動させます。</li> <li>▶ Claude wants to execute this command:</li> </ul>									
	<pre>move index.html templates\</pre>									
8	Run Command Reject									
522 2023		6								

# Claude Devの画面でバックエンドタスクを割り当てる⑤

7	アイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H)		PJ_Cost_Calculation			
1	CLAUDE DEV	+ 3 [] 🕸	◇ index.html	🕏 app.py: New File (Editable) 🔹		···□ ₩ IP ↓ ↑
)	✓ Task @index.html	×	app.py 1-			1+ from flask import Flask, render_template, request, json 2+ from flask_sqlalchemy import SQLAlchemy 3+ from datetime import datetime
>	You are tasked with converting an HTML file (index.html) into a Flask web application using SQLite for datab <b>Tokens:</b> ↑ 20 ↓ 1,505 <b>Cache:</b> ⊖ +14,217 → 19,002 <b>API Cost:</b> \$0.0816	ase operations. The applical See more				1+ 5+ app = Flask(name) 5+ app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///data 7+ db = SQLAlchemy(app) 24
S S						9+ class TableData(db.Model): 9+ id = db.Column(db.Integer, primary_key=True)
*	Command Output					<pre>1+ date = db.Column(db.Date, nullable=False) 2+ member = db.Column(db.String(100), nullable=False) 2+ mit aris, db Columa(db Ithana mullable False) 3- mit aris, db Columa(db Ithana mullable) 3- mit aris, d</pre>
•	✓ API Request \$0.0054					<pre>unit_price = db.Column(db.Integer, nullable=False) + work_hours = db.Column(db.Integer, nullable=False) 5+</pre>
	申し訳ありません。Windows環境でのコマンド実行に問題がありました。コマンドを分割して実行し直しま 作成し、次にindex.htmlを移動させます。	す。まず、templatesディレクトリを				5+@app.route('/') 7+def index(): 3+ data = TableData.query.all()
	Claude wants to execute this command:					0+ return render_template('index.html', data=data) 0+
	mkdir templates > Command Output					l+@app.route('/add', methods=['POST']) 2+ def add_data(): 3+
	✓ API Request \$0.0048					<pre>4+ member = request.form['member'] 5+ unit_price = int(request.form['unit_price']) 5+ work hours = int(request form['work hours'])</pre>
	templatesディレクトリが正常に作成されました。次に、index.htmlファイルをtemplatesディレクトリに移動 Claude wants to execute this command:	させます。				<pre>/+ /+ /+ /+ /+ /+ /+ /+ /+ /+ /+ /+ /+ /</pre>
	<pre>move index.html templates\</pre>				30 31 32	<pre>db.session.commit() L+ 2+ return jsonify({'success': True, 'id': new_data.id}</pre>
	✓ API Request \$0.0397				33 34 39	3+ 4+@app.route('/edit', methods=['POST']) 5+ def edit_data():
	index.htmlファイルがtemplatesディレクトリに正常に移動されたようです。次に、app.pyファイルを作成し、 後、templates/index.htmlの内容を修正します。	必要なコードを実装します。その	問題 出力 ラ PSE:\000 Udem	デバッグ コンソール <mark>ターミナル</mark> ポート ny\0 VS Code\PJ Cost Calculatio	n> mkdir templates	laude Dev +∨ □ 前 … へ ×
	まず、app.pyファイルを作成し、基本的な構造を実装します。					
	[]+       Claude wants to create a new file:		ディレクト	リ: E:\000_Udemy\0_VS Code\PJ_	Cost_Calculation	
	арр.ру		Mode 	LastWriteTime L	ength Name	i) Python 言語 用のおすすめ拡張機能 'Python' 拡張機能 提供元: 〇〇 × Microsoft ちくいてトーリー ますか?
۷	Save	Reject		2024 10/07 0.55	Compraces	With Osoft をインストールしよりカイー
22		A D	PS E:\000_Uden PS E:\000_Uden	y\0_VS Code\PJ_Cost_Calculatio y\0_VS Code\PJ_Cost_Calculatio	n> move index.html template n> []	S TEXT NOW
$\bigcirc$		Mark Community of Dractice DM	l lanan 2024 (	Converight and all rights record	aved	行らん 別 フルーフ・ルーHTE & CPLE Duthon (●

# Claude Devの画面でバックエンドタスクを割り当てる⑥

7	ァイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(	H)	℅ PJ_Cost_Calculatio	n			
Ç	CLAUDE DEV	+ 5 🛙 🔅	index.html	刘 ようこそ	🕏 app.py 2 🗙		$ ho$ ~ $\square$ …
	<ul> <li>✓ Task</li> <li>@index.html</li> <li>You are tasked with converting an HTML file (index.html) into a Flask web ap</li> <li>Tokens: ↑ 24 ↓ 1,650</li> <li>Cache: ① +22,323 → 19,002</li> <li>API Cost: \$0.1142</li> </ul>	X oplication using SQLite for database operations. The applical See more	<pre>     app.py &gt;         from f             2 from f             3 from c             4             5 app =             6 app.cc             7 db = 5             8             8</pre>	lask import Flask lask sqlalchemy i latetime import da Flask(name) onfig['SQLALCHEMY_ QLAlchemy(app)	, render_templa mport SQLAlchem tetime DATABASE_URI']	ate, request, jsonify my = 'sqlite:///database.db'	Particular and a second s
	<ul> <li>Claude wants to execute this command:</li> <li>mkdir templates</li> <li>&gt; Command Output</li> </ul>		9 class 10 ic 11 da 12 me 13 ur 14 wc	TableData(db.Mode   = db.Column(db.I  te = db.Column(db  mber = db.Column(  ti_price = db.Col  urk_hours = db.Col	l): nteger, primary .Date, nullable db.String(100), umn(db.Integer, umn(db.Integer,	y_key=True) e=False) , nullable=False) , nullable=False) , nullable=False)	
9	✓ API Request \$0.0048 templatesディレクトリが正常に作成されました。次に、index.htmlファイルを S Claude wants to execute this command:	~ をtemplatesディレクトリに移動させます。	15 16 @app.r 17 def ir 18 da 19 re 20	route('/') Wdex(): Wta = TableData.qu Wturn render_templ	ery.all() ate('index.html	l', data=data)	
	<ul> <li>move index.html templates\</li> <li>&gt; Command Output</li> <li>✓ API Request 500397</li> <li>index.htmlファイルがtemplatesディレクトリに正常に移動されたようです。2 後、templates/index.htmlの内容を修正します。</li> <li>まず、app.pvファイルを作成し、基本的な構造を実装します。</li> </ul>	〜 欠に、app.pyファイルを作成し、必要なコードを実装します。その	21 @app.r 22 def ac 23 da 24 me 25 ur 26 wc 27 28 ne 29 dt 30 dt	<pre>voute('/add', meth ld_data(): rte = datetime.str ember = request.fo it_price = int(re ork_hours = int(re w_data = TableDat o.session.add(new_ o.session.commit()</pre>	ods=['POST']) ptime(request.f rm['member'] quest.form['uni quest.form['wor a(date=date, me data)	form['date'], '%Y-%m-%d').date() it_price']) rk_hours']) ember=member, unit_price=unit_price, work_hours=w	ork_hours)
	[*]     Claude wants to create a new file:       app.py		31 32 re 33 34 @app.r 35 def ec	<pre>sturn jsonify({'su poute('/edit', met lit_data():</pre>	ccess': True, ' hods=['POST'])	'id': new_data.id})	
	<ul> <li>✓ API Request 300326</li> <li>app.pyファイルが正常に作成されました。次に、templates/index.htmlファイ、ます。まず、現在のindex.htmlの内容を確認し、必要な修正を加えます。</li> <li>〇 Claude wants to read this file:</li> </ul>	~ ルの内容を修正して、Flaskアプリケーションと連携できるようにし	問題 2 出力 PSE:\000_Ude ディレクト	デバッグ コンソール タ my\0_VS Code\PJ_Cos ・ リ: E:\000_Udemy\0	<u>ーーミナル</u> ポート t_Calculation> m _VS Code\PJ Cost	kdir templates	é∰ Claude Dev +∨ [] 前 … ∧ X
3	templates/index.html Approve	[2] Reject	Mode  d	LastWriteTin  2024/10/07 0:	me Length 	– h Name  templates	
20 0	Type a message (@ to add context)	 © ⊳	PS E:\000_Ude PS E:\000_Ude	my\0_VS Code\PJ_Cos my\0_VS Code\PJ_Cos	t_Calculation> mo t_Calculation> []	wove index.html templates\ 行1、列1 スペース:4 U	TF-8 CRLF {} Python 3.12.7 <u>64-bit</u> Ω

# Claude Devの画面でバックエンドタスクを割り当てる⑦

7	マイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H) ← →	,	Iculation			l	
	claude dev + 3 [2 §	용 🔹 index.ht	tml 📢 ようこそ	🕏 app.py 2	index.html: Original	↔ Claude's Changes (Editable) ●	$\Box \Leftrightarrow \mathbb{P} \checkmark \uparrow$
)	✓ Task @index.html	templates <b>1</b> !<br 2 <h< th=""><th>&gt; <pre>&gt; <pre> index.html &gt; DOCTYPE html&gt; tml&gt;</pre></pre></th><th></th><th></th><th>1 <pre>1 {DOCTYPE html&gt; 2 <html></html></pre></th><th></th></h<>	> <pre>&gt; <pre> index.html &gt; DOCTYPE html&gt; tml&gt;</pre></pre>			1 <pre>1 {DOCTYPE html&gt; 2 <html></html></pre>	
>	You are tasked with converting an HTML file (index.html) into a Flask web application using SQLite for database operations. The applical See more Tokens: ↑ 28 ↓ 4,490 Cache: ⊕ +24,728 → 27,108 API Cost: \$0.1683	3 <he 4 <t 5 <so 6 <s< th=""><th>ead&gt; itle&gt;PJ稼働コスト管理 cript src="<u>https://c</u> //////////////////////////////////</th><th>表 dn.jsdelivr.net/n</th><th>npm/chart.js'</th><th>3 <head> 4 <title>PJ稼働コスト管理表5 <script <u="" src="https://cdn.jsde&lt;br&gt;6+ &lt;script src="></script></title></head></th></s<></so </t </he 	ead> itle>PJ稼働コスト管理 cript src=" <u>https://c</u> //////////////////////////////////	表 dn.jsdelivr.net/n	npm/chart.js'	3 <head> 4 <title>PJ稼働コスト管理表5 <script <u="" src="https://cdn.jsde&lt;br&gt;6+ &lt;script src="></script></title></head>	

# Claude Devの画面でバックエンドタスクを割り当てる⑧

4 :	ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H)	PJ_Cost_Calculation		
¢	claude dev + 3 [2 🍪	♀ <del>index.html \</del> ▲ ▲ ようこそ   � app.py 2	♦ index.html templates ×	□ …
		templates > 🔇 index.html >		
Q	∼ Task ×	1 html		The State of the S
	@index.html	2 <html></html>		1742 -
وع	Volu are tacked with converting an HTML file (index html into a Flack web annligation using SOL its for database operations. The annline See more	3 <head> 4 &lt;+i+laspi按師コフト管理表(+i+las</head>		Tana -
ð	to be the state of the content and an intermediate indexing and a mask web appreciation asing state for database operations. The appreciation and the state of th	4 、CITEPPS報酬コスト日荘(ス/CITEP 5 〈Script src="https://cdn.isdelivr.net/r	nom/chart.is">	The second se
$\sim$	Cacher 1 + 27 63 - 37 619	6 <script src="https://code.jquery.com/jc&lt;/th&gt;&lt;th&gt;uery-3.6.0.min.js"></script>	ger in Katana san Marika sa san Marika sa san	
Ġ∕	ADI Cost \$0.1070 EXPORT	7 <style></style>		

### Claude Devの画面でバックエンドタスクを割り当てる⑨

4	ファイル(F) 編集(E) 選択(S) 表示(V) 移動(G) 実行(R) ターミナル(T) ヘルプ(H)	P PJ_Cost_Calculation	o ×
β	CLAUDE DEV + D [] 🕸	③ ◇ index.html ÷ ズ ようごそ ◆ app.py 2 ◇ index.html templates ×	□ …
ے چ	<ul> <li>✓ Task</li> <li>@index.html</li> <li>You are tasked with converting an HTML file (index.html) into a Flask web application using SQLite for database operations. The applica See more</li> <li>Tokens: ↑ 32 ↓ 5,524</li> <li>Cache: ⊕ +27,638 → 37,619</li> </ul>	<pre>templates &gt; ◇ index.html &gt; 1 <!DOCTYPE html>     2 <html> 3 <head> 4 <title>PJ稼働コスト管理表</title> 5 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script> 6 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> </head></html></pre>	The second secon
~ ⊑⊘ ⊒	API Cost: \$0.1979       EXPORT         ・ データの追加:フォームに新しいデータを入力し、「追加」ボタンをクリックすることでデータを追加できます。       ・         ・ データの編集: 各行の「編集」ボタンをクリックすると、データを編集できます。       ・         ・ データの削除:各行の「削除」ボタンをクリックすると、データを削除できます。       ・         ・ グラフの表示:日別、月別、年別のコストグラフと、月別コスト内訳の円グラフが表示されます。       ・	<pre>7 <style> 8 body { font-family: sans-serif; } 9 table { width: 100%; border-collapse: collapse; } 10 th, td { border: 1px solid</th><th></th></tr><tr><th>نني ۲</th><th>注意点: 1. 初回実行時は、データベースが空の状態で始まります。いくつかのサンプルデータを追加して、グラフの動作を確認してください。 2. エラーが発生した場合は、ターミナルのエラーメッセージを確認し、必要に応じて修正を行ってください。 3. アプリケーションの改善や機能の追加が必要な場合は、app.pyやtemplates/index.htmlを適宜修正してください。</th><th>13 </style> 14  15 <body> 16 <h1>PJ稼働コスト管理表</h1></body></pre>	In an
	これで、Flask Web アプリケーションのセットアップと実行手順の説明が完了しました。アプリケーションを実行し、正常に動作することを確認 してください。 タスクが完了したので、attempt_completionツールを使用して結果を提示します。	18        19 <thead>       20      </thead>	
	✓ Task Completed Flask Web アプリケーションの作成が完了しました。以下の手順でアプリケーションを実行できます:	21 <th< th=""><th></th></th<>	
	1.必要なモジュールをインストール: pip install flask flask-sqlalchemy	25 <th< th=""><th></th></th<>	
	2. アプリケーションを実行: python app.py 3. ブラウザで http://localhost:5000 にアクセスしてアプリケーションを確認してください。	28 29 30   データはJavaScriptで動的に追加 31	
	アプリケーションには以下の機能があります: ・ データの表示、追加、編集、削除 ・ 日別、月別、年別のコストグラフ ・ 月別コスト内訳の円グラフ	32 33 34 <h2>データ追加</h2> 35 <form id="add-form"></form>	
	初回実行時はデータベースが空なので、サンプルデータを追加してグラフの動作を確認してください。エラーが発生した場合は、ターミナルのエ ラーメッセージを確認し、必要に応じて修正を行ってください。	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	<u>∎</u> … ∧ ×
	Claude wants to execute this command:         python app.py	Mode         LastWriteTime         Length         Name                 d         2024/10/07         0:35         templates	
	Command Output	PS E:\000_Udemy\0_VS Code\PJ_Cost_Calculation> move index.html templates\ PS E:\000_Udemy\0_VS Code\PJ_Cost_Calculation> python app.py	
8 8	Start New Task       Type a message (@ to add context)	<pre></pre>	
•u•		OPS F:\000 Udemv\0 VS Code\P] Cost Calculation>	

# 作成したWEBアプリを動作させてみる①

#### フロントエンドとバックエンドの連携版

#### エクスプローラで開くと、作成したhtmlファイルが出来ている

⋈	ファイル(F)	編集(E)	選択(S)	表示(V)	移動(G)	実行(R)	ターミナル	(T)	ヘルプ(H)	
¢	エクスプ 〜 AI_DRI	d-j- Ven_pj								
$\mathcal{P}$	<b>◇</b> pj_o	cost.h*'	構に並べて	盟へ		Ctrl+Fr	oter			
မိုစ				で表示: で表示:	する	Shift+Al	t+R			ĺ
å			ファイルを開 統合ターミナ	マアフリクー ナルで開く	ショノの選り	<b>т</b>				
00			比較対象(	の選択						
			タイムライン	を開く						
Ē			切り取り			Ctr	I+X			
			コピー			Ctr	l+C			
			パスのコピー 相対パスを	วピ–	Ctrl+k	Shift+Alt Ctrl+Shift	t+C t+C			
			名前の変更	Ē			F2			
			削除			Del	lete			



#### 作成したhtmlファイルを開くと⇒次ページ



### 作成したWEBアプリを動作させてみる②

#### フロントエンドとバックエンドの連携版

•	S PJ稼働□	スト管理表	×	+			-	
$\leftarrow$	$\rightarrow$ G	(i) 771lk	file:///E:/00	0_Udemy/0_VS%20Code/PJ_Cost_Calculation/templates/index.html	☆	1 🕹	}   ≡	=7 ●

#### PJ稼働コスト管理表

日付	メンバー	稼働時間(HR)	稼働単価(円)	日別コスト(円)	操作
データ追加					
年 /月/日 📋 メンバー	稼働時間	稼働単価			
日別コスト					

#### グラフ機能を実装しているが、まだ上手く表示されないので、この部分の追加タスク指定が必要! ヒューマンインザループで、設計者が確認しながら、フロントエンドとバックエンドの仕様をブラッシュアップしていく

月別コスト

#### 年別コスト

# Claude DevによるWebアプリ構築の流れ (纏め)

### Claude DevによるWebアプリ構築の流れ

※ フロントエンドは、軽量・効率的なGemini Flash1.5を利用
 バックエンドは、性能が要求されるので、Anthropic Claude 3.5 Sonnetを利用



46

# まとめ

- VS Code + Claude Devにより、生成AIを活用して、PJを効率的に管理するWebアプリの自動生成の試行を 行った。
- Web上のUIを実現するフロントエンド機能、SQLiteをDBとしてデータの蓄積・再利用に利用するバックエンド機能の両方を、Prompt入力により自動生成出来ることを確認した。
- ➤ バックエンド機能の実現には、Anthropic ClaudeによるPrompt自動生成機能を利用した。これはバックエンドの 各機能要素を、設計者の確認を得ながら逐次自動生成出来る。
- ▶ PJ管理を支援するWebアプリケーションは、情報の共有、オープン化で、PJに関係する全てのステークホルダ間の Community力を強化することに役立つ。
- ▶ 今回の試行は、完全なアプリケーションを作ったものではないが、生成AIの進化により、今後のPJ管理の効率化に大きく貢献出来そうな感触を得た。
- PJの各種実績をRAGやSQLiteなどのDBに蓄積しつつ、PJのコスト予想・リスク分析、ポートフォーリオ選択に繋げるよう、生成AIの進化をWatchしつつ取り込んで行きたい。
- ▶ ただし「自動生成」による効率化は大きいが、PMや設計者がPJの方向を定めるために、正しいINPUTを生成AL にStep By Stepで与えることは益々重要であり、PM自身もこの動きを常に学んで取り込んでいく必要がある。

本資料の作成グループ:AI@Work WG1 D0チームでは、プロジェクトマネジメントの観点で、 AIを含むプロジェクトを成功させて、価値を実現するための手法を研究しています。 研究の成果を、コンフェレンスでの発表、ブログ記事、スライドなど様々な形式で発信していきます。

我々と一緒に研究したいという仲間の方を常時募集しています。 またお気付きの点、アイデアとか、前向きに取り組みを進めるためのご意見をお待ちしております。 (Al@Workまで)