



AIと一緒に設計・開発！ChatGPTで 基本設計から実装までを体験しよう

2回目

2025/12/09

Kazuma Sekiguchi

目標

- PythonのFlaskを利用してある程度のWebシステムを作成できる
- Pythonの初期文法などは触れません
- Pythonによる統計解析や自然言語処理、AIとの連携などは行ないません

前回のAgenda

- AIを使って「基本設計書」を自動生成する
- AI活用の概要と注意点
- 基本設計に必要な要素の解説
- AIへの指示（プロンプト設計）の方法
- 実践：AIで基本設計を作成する

基本設計書まで作成が完了している状態

今回のAgenda

- 第1回で作成した基本設計書をもとにFlaskアプリを構築
- Flask概要と環境準備
- AIにコードを生成させる
- コード修正と機能追加の体験

AIを使いつつプログラム作成

- 現場的には結構当たり前になりつつある
 - GoogleなどでもAIが生成するコード量が3割など
- 100%すべてをAIに任せるというのは、今のところ現実的では無い
 - 仕様が明確に決まっていない場合
 - 既にある程度コードができている場合
 - などはバグのあるコードを出しやすい
- ある程度AIにコードを生成させてテストを行ない、問題のある箇所を指摘しつつ、再度AIに修正させるのが現実的

AIによるコード作成

- ある程度動作するコードをAIで作成可能
 - ただし、動く、というだけ
 - セキュリティ的に安全なのか、他のシステムとの連携は考慮されているか、コードが保守しやすいものになっているかどうか、などはきちんとできあがったコードを精査する必要がある
 - 特に動くことだけに集中していると、セキュリティ的に不完全なものができやすい
 - コードができて動くことが確認できた段階で、改めてセキュリティチェックをおこなう、コードを精査する必要がある

いきなりコードを書くと危険

- いきなりコードから書き始めると大体失敗する
 - 全体像が把握できていないため、どこがゴールなのか分からぬ
 - 利用者が誰なのか、誰のための仕組みなのかが不明確になりやすい
 - データの構造、データベース構造などがその場その場で作成してしまうため、効率の悪い構造になりやすい
 - 権限などのセキュリティ部分が抜けやすくなる
- 結果としてコードが分かりづらく、作り直した方が早い、という結果になることも
- プロトタイプ的に一部分だけを作成するというケース以外は簡単な仕様でも良いから作成するべき
- 理想的には基本設計→詳細設計→コード作成

AIにコードを書いて貰う

- ・サンプルコードを書いて貰う程度なら問題はあまりない
 - ・ある程度動作するコードを書いてくれる率が高い
 - ・不得手なプログラミング言語もある
 - ・利用者が少ない言語ほど間違えが多い、ように感じる
- ・比較的正しいコードが出てくるプログラミング言語
 - ・JS、Java (Android) 、Python、Goなど
 - ・マイナーな言語は学習量が足りないので、結構怪しい
 - ・型指定のできる言語の方が間違えが少ない
 - ・その点ではTypeScriptやJavaなどがかなり優秀

AIにちゃんとしたコードを書いて貰う

- 設計書が必要
 - 要件定義書、基本設計書、画面一覧、DB設計（データ）などの一式を用意する
 - できるだけ詳細に定義をしておけばそれだけ望んでいる誤差が少なくなる
 - 曖昧な表現をしてしまうと、AIが勝手に判断してコードに反映することがあるため、曖昧さを極力排除することが重要
 - 設計書に矛盾が生じていると、更に変なコードになるので、矛盾が無いかどうかを確認しておく

AIにちゃんとしたコードを書いて貰う

- ・適宜確認すること
 - ・AIが誤ったコードを出すのは当たり前、という認識でいるべき
 - ・間違えたコードを出してきたときに、どこが間違えているのかを明確に指示する
 - ・正しく書けているところも明確に指示し、修正するべき箇所を的確に指摘する
 - ・正しく書けているところを指示しないと、勝手に修正することが多々ある
 - ・一気にすべてのコードを吐き出させようとしない
 - ・長いコードになると、エラー率が急激に上がる
 - ・できるだけ機能ごとなどに分割して与えていく

AIにちゃんとしたコードを書いて貰う

- プロンプトで不明確な点があれば、聞き直すように指示しておく
 - 不明確な点などは人が指示して解決するしかない
- セキュリティ、特に権限などは注意が必要
 - 基本的にすべての人が使えるように作成する傾向が強い
 - どの権限を持つユーザーがいて、どの権限だと何ができるのか、を明確にしておく必要がある
- 使うデータまたはデータ構造を明確にしておく
 - もし、データが既にあるのなら、プロンプトを通じて渡して、データ構造を最適にして貰うのも手ではあるが、AIによって流出する懸念がある点に注意が必要

AIがよくやる行動

- ・指示をしないにも関わらず勝手にAPI化する傾向が強い
 - ・今回のようにAPIを使わない場合は、指示を明確にしておく
- ・正常に動いていたコードを変更して動かなくなる
 - ・他を直そうとして指示をしたときに、他のコード部分も勝手に変えることがある
- ・言っていたことが変わる
 - ・これで完全に動きます、は大体動かない

AIの間違いを正す

- 出してきたコードを動かしたときにエラーとなつたときは、詳細に情報を与えること
 - 動きません、では解決できない
 - ログや動かしたときに出てきたエラーメッセージをコピーして渡す、画面のスクリーンショットを取得して渡すなど、できるだけ多量の情報を与える
 - 何度かトライアンドエラーを繰り返しても直らない場合は、AIを変える
 - ChatGPTならGeminiやClaudeにするなど
 - 何回か繰り返していると、同じ間違えをしているコードを提示して堂々巡りになることもある

AIへの指示の仕方

- 使うフレームワークやライブラリーは明確に指示する
 - 日本語を使う、と指示した方が良いこともある
 - 不要なのも伝える（既に作成済みのものなど）
 - 何を作成して欲しいかを明確に伝える
 - コード全体なのか、コードの一部のクラスなのか、マイグレーションファイルなのか

Flask

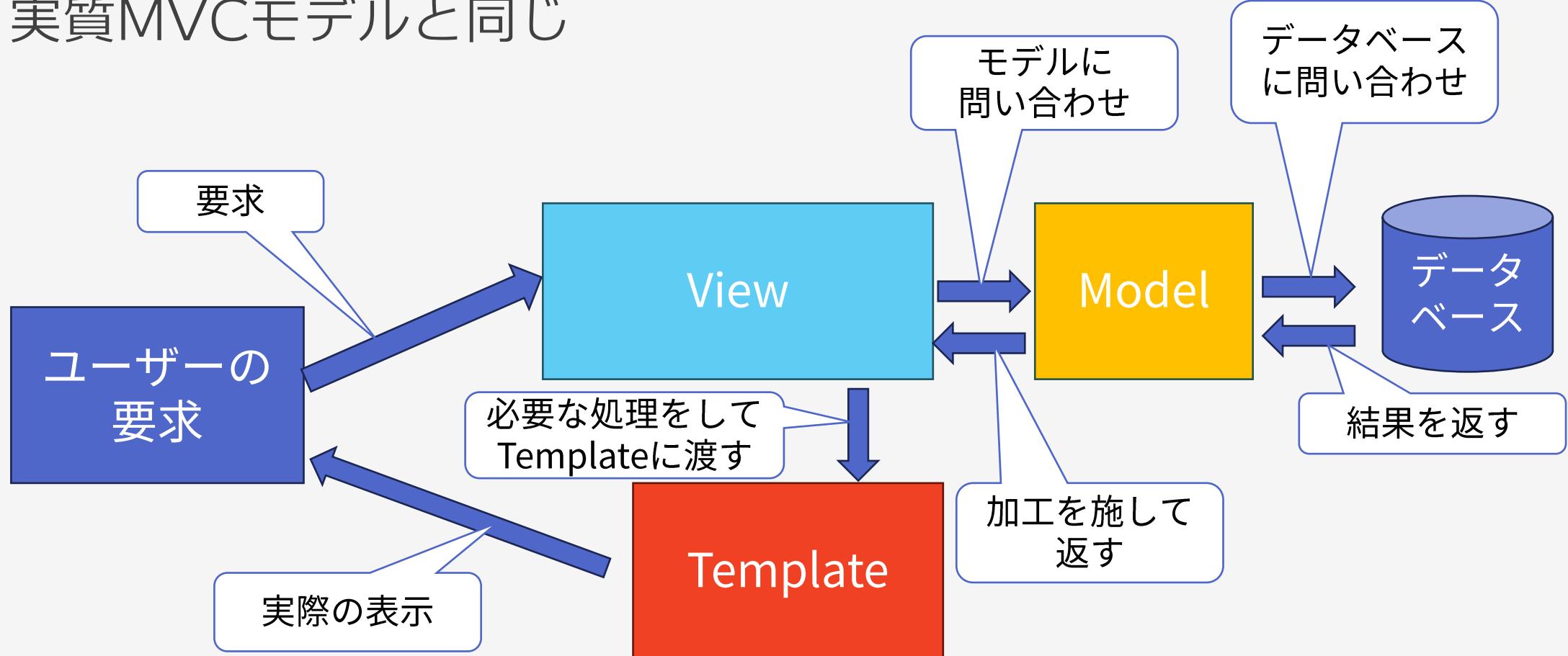
- ・マイクロフレームワークとも呼ばれていて、標準的に備わっている機能は最小限度に留まる
 - ・必要に応じて別途パッケージやライブラリーを追加する必要がある
 - ・拡張機能ライブラリーとしてFlask互換のものが提供されている
 - ・必要に応じて導入すれば動作する
 - ・要らないものは入れる必要が無いため、動作が軽量のままで済む
- ・最小限度しか提供されていないため、フレームワークの学習はしやすい

Flask

- 作者はオーストリア人のArmin Ronacher
 - エイプリルフールのジョークとして作ったのだけれど、有名になつて、いつのまにか本格的なアプリケーションになつてしまつた、らしい (wikipediaより)
 - 2010年4月に登場
- 今はFastAPIが人気もある
 - もっともREST APIに特化しているところがあるため、通常の Webシステムを作るには不適

Flask

- FlaskはMVTモデルを採用している
 - 実質MVCモデルと同じ



FlaskのMVTモデル

- View : ユーザーからのリクエスト（要求）を受け取って ModelやTemplateに処理を渡す
- Model : データベースにアクセスをし、データを保存したり取り出したりする
- Template : HTMLなどの表示用ファイルを作成する
実際にデータとして表示される部分を担う
- それぞれのファイルが別々の役割を担うように処理を分離して記述していく

Flaskのインストール

- Flask3.1.2がリリースされているため、これを利用する
- Python3.8以上が必要
- pipでインストールが可能
 - バージョンを指定する点に注意

```
> pip install flask==3.1.2
```

macなどではpip3
と指定する

Flaskの利用

- 先頭でFlaskをimportする
 - 他にライブラリーなどを使う場合もimportを行なう
- 拡張子はpyでOK
 - 別のWebサーバーなどと連携して動作させるのであれば、wsgiなどの拡張子を指定する

```
from flask import Flask
```

Flaskのプログラムを実行する

- Flaskを使ったPythonプログラムを実行する場合は、ターミナルなどからコマンドで実行する

```
> cd 記述したプログラムを保存しているフォルダー
```

- cdコマンドを入力してスペースを入れて、プログラムを保存しているフォルダーを指定してEnterキーで移動する

```
> flask --app app run --port=8080
```

- 上記を入力してEnterキーを押せば、画面に127.0.0.1のURLが表示されるので、そこにブラウザでアクセスする

Flaskのプログラムを実行する

- 終了するときは、Flaskを実行しているターミナルで Ctrl+C を押す (Windowsの場合。macの場合は、control+cを押す)
- ターミナルを閉じても終了するので、それでもOK

Flaskのプログラムを実行する

```
回目> flask --app app run --port=8080
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:8080
Press CTRL+C to quit
```

- --port=8080は省略可能
 - 記述しない場合5000番として起動する
 - その場合はhttp://127.0.0.1:5000としてアクセスする
 - マレに5000番を利用しているケースがあり、その場合はプログラムが起動しないので、--port=で番号を変更して起動させる
 - --debugを付与すると開発時に便利

今回作成するもの

- ・前回作成した基本設計書を元にプログラムコードを ChatGPT に記述してもらいつつプログラムを作成する
- ・言語：Python
- ・フレームワーク：Flask
- ・データベース：SQLite
- ・外部連携無し、社内専用ツール

作成手順

1. Flask最小構成の作成
2. DB接続 + models.pyの実装
3. ログイン機能
4. お問い合わせ登録
5. 検索・詳細・ステータス更新・履歴
6. 管理者機能の作成

表示確認時などにコマンドを使いますが、macの方は
pythonはpython3、pipはpip3として入力してください

ありがとうございました。
AIによるプログラムコード作成は強力ですし、
速度の向上に役立ちます。

きちんとした設計書を用意することで、
ブレのない、エラーの少ないプログラムを
書いて貰うことが可能になります。
上手く活用してみてください