

# 株式会社 iDea コーディング課題

## 本課題の目的

この課題は、株式会社 iDea のインターンシップ参加希望者向けのものです。

課題では **ChatBot Webアプリケーション** を開発していただきます。

iDea では大規模言語モデルを活用したプロダクトを多数開発しており、本課題を通じて得られる知識や技術は、実際の開発業務に直結します。

最終的には、iDea の開発メンバーとしてスムーズに参画できるだけの **技術力・実装力を習得すること** が目的です。

## 課題内容

### 課題1：起動準備とシステム起動

#### 1. 環境ファイルの作成

- リポジトリ内の `.env.example` を参考に、環境ファイル `.env` を作成してください。
- 作成した `.env` を `backend/.env` として配置してください。

#### 2. システムの起動

- 本システムは **Docker コンテナ** として動作します。
- 起動には **docker compose** を使用します。
- `backend/Dockerfile` と `frontend/Dockerfile` に、それぞれのビルド方法が定義されています（今回は編集不要）。
- `docker-compose.yml` にサービス構成・依存関係・ポート公開が定義されています。
- 起動手順:

```
docker compose build
docker compose up
# または一発で
docker compose up --build
```

- 起動後の確認:
  - フロントエンド: <http://localhost:5173>
  - バックエンド API ドキュメント（任意）: <http://localhost:8000/docs>

### 課題2：OpenAI APIの利用

- `backend/app/services/ai_client.py` の `generate_reply` メソッド内にある `TODO` を実装してください。
- OpenAI API から応答を取得できるようにします。
- 実装の参考: [公式ドキュメント](#)
- 実装後、`backend/.env` の環境変数 `OPENAI_DRYRUN` を `1` → `0` に変更すると、実際にAIが応答するようになります。

## 課題3：DB構築・利用

- `backend/app/db.py` の `init_db_with_retry` などの `TODO` を実装し、DBの初期化とテーブル作成を行ってください。
- `backend/app/models.py` に `ChatMessage` モデルを定義してください。

### 1. DB操作（CRUD）

- `backend/app/repositories/history.py` の CRUD 関数（`create_message` など）の `TODO` を実装してください。

### 2. チャット履歴との連携

- `backend/app/routers/chat.py` の `TODO` を実装し、チャット履歴の保存・取得・編集・削除が DB 経由でできるようにしてください。
- 実装後、`backend/.env` の `ENABLE_DB` を `false` → `true` に変更すると、ChatBot にDB機能が適用されます。

## 課題4：リセットボタンの実装

- `frontend/src/App.tsx` のリセットボタンの `TODO` を実装してください。
- 履歴削除 API と連携させます。
- 併せて `frontend/src/api.ts` の `deleteHistory` 関数も実装してください。

## 課題の進め方

- 各 `TODO` にはガイドコメントが付いています。順番に実装を進めてください。
- **開発未経験の方は、AIツールの活用を強く推奨します。**
  - AIツールを使うことで効率的に開発を進められます。
  - 回答を理解しきれなくても、「何が分からないのか」を整理して再質問すれば、解決できることがあります。
  - コードをそのまま丸投げするのではなく、**生成されたコードの意味を必ず理解すること**が重要です。
  - 理解できないときは、生成コードの解説もAIに依頼しましょう。

本課題で実装する内容は、iDea の実務に直結します。

完全に暗記する必要はありませんが、「やったことがある」「調べれば思い出せる」状態を目指してください。

## 推奨AIツール例

- [ChatGPT](#)  
汎用性が高く、幅広い場面で活用できます。有料プランも検討の価値があります。
- [GitHub Copilot](#)  
VSCode 拡張機能として利用でき、開発中のコードを読み取って自動補完や提案をしてくれます。  
勝手にコードが書き換わることはなく、必ずユーザーの操作で適用される仕組みです。
- その他の生成AIツールもぜひ試し、自分に合った方法を見つけてください。

## 完了の目安

- ChatBot システムを起動できること
- 以下がすべて正常に動作すること
  - チャット送信
  - AI応答
  - 履歴の保存 / 取得 / 編集 / 削除
  - リセットボタン

---

これで課題の説明は終わりです。

自分のペースで取り組んでください。応援しています！🚀