PyNavigator 仕様書・基本設計書

# 1. アプリ概要

■ アプリ名称案: PyNavigator（パイ・ナビゲーター）

初心者からAI開発者への航海をサポートするWebアプリ

# 2. アプリ構成図（機能一覧）

|  |  |  |
| --- | --- | --- |
| 画面カテゴリ | 主な機能 | 技術的ポイント |
| ホーム | 学習進捗表示・おすすめレッスン | Firestore連携、ユーザーセッション管理 |
| レッスン | 基本文法／Flask／機械学習などの単元ごとに章立て | Markdown＋コードエリア＋実行 |
| クイズ | 各単元ごとの理解度チェック（選択式・○×） | Streamlit formで作成可 |
| 実践課題 | WebアプリやAIモデル作成の実践課題 | ファイルアップロード、コード提出、自動評価 |
| AI学習アシスタント | OpenAI APIによるQ&Aチャット | GPT API連携、コード補助・ヒント提示 |
| マイページ | バッジ・到達スコア・学習履歴 | Firestoreで保存管理 |
| プラン選択 | 広告付き無料プラン／サブスクでAIサポート | Stripe連携またはサンプルの擬似課金 |

# 3. 初期実装ステップ案（Streamlit）

* Step 1: ベースUIとログイン機能  
  Streamlit + streamlit\_authenticatorで認証  
  Firestore or Supabaseでユーザーデータ管理
* Step 2: レッスン構成  
  MarkdownまたはJSONで定義  
  st.code\_editor＋execでコード実行
* Step 3: クイズモジュール  
  選択式問題＋即時フィードバック＋スコア保存
* Step 4: AIチャット支援  
  OpenAI GPT-4o＋チャットUI（streamlit\_chat）
* Step 5: 実践課題提出機能  
  コード提出＋自動テスト or 人的レビュー
* Step 6: サブスク連携  
  広告表示／制限付き、Stripeなどと連携

# 4. ディレクトリ構成案

* /pynavigator\_app
* ├── main.py
* ├── pages/
* │ ├── Home.py
* │ ├── Lessons.py
* │ ├── Quiz.py
* │ ├── Practice.py
* │ ├── Assistant.py
* │ └── Profile.py
* ├── data/
* │ ├── lessons/
* │ └── quizzes/
* ├── utils/
* │ ├── auth.py
* │ ├── lesson\_loader.py
* │ └── gpt\_helper.py
* ├── static/
* │ └── icons/
* └── config.toml

# 5. UIイメージ（Home画面）

[🧭 PyNavigatorへようこそ]  
  
👤 ユーザー：たかぴー さん  
🎯 学習ステージ：Webアプリ開発（Flask）  
  
[📚 レッスンを続ける]  
▶ 変数とデータ型（完了）  
▶ if文とループ（進行中）  
▶ 関数とモジュール（未開始）  
  
[🧪 今日のクイズ]  
Q. forループの基本構文は？  
  
[🤖 AIチューターと学習]  
「Flaskのルーティングが分かりません」→ チャット起動