プロジェクト進行計画

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **期間** | **概要** | **タスク** | **工数**  **(目安)** | **Phase**  **総工数** |
| 1 | 10/17(金), 10/18(土) | ChatGPTを活用する準備を整え、開発環境と全体計画を完成させる | ① 仕様書と要件の理解（全体像の把握） | 1h | 4h |
| ② スケジュールの作成（タスク・工数見積もり）, 提出 | 1.5h |
| ③ VSCode + Python + Git の開発環境構築 | 1h |
| ④ GitHubリポジトリ作成と初期設定 | 0.5h |
| 2 | 10/19(日), 10/20(月) | Codexが理解しやすい全体設計書・モジュール設計書を作成し、依存関係を整理する | ⑤ 全体設計書の作成（アーキテクチャ、データ構造、API仕様） | 2h | 6h |
| ⑥ 各モジュールの設計書作成 | 2h |
| ⑦ モジュール間の依存関係・データフローの整理 | 2h |
| 3 | 10/21(火), 10/22(水), 10/23(木) | 設計をもとにタスクを整理し、アプリの主要機能を開発する | ⑧ 実装タスクの分解・優先度整理 | 2h | 20h |
| ⑨ 開発（main / db / models / routes / rss / summarizer / scheduler など（Phase2での設計書による）を含む全体実装＋動作確認） | 18h |
| 4 | 10/24(金),  10/25(土) | APIおよび自動ジョブの動作を確認し、安定稼働を実現する | ⑩ 単体テスト（RSS取得・要約処理など） | 2h | 5h |
| ⑪ 結合テスト（要人登録→RSS収集→DB保存） | 2h |
| ⑫ ログ・例外処理の確認（新規数・スキップ数・エラー数） | 1h |
| 5 | 10/26(日),  10/27(月) | Railwayでアプリを常時稼働させ、提出物（GitHub＋URL＋共有履歴）を整える | ⑬ GitHubリポジトリとRailwayの連携・デプロイ確認 | 1.5h | 4h |
| ⑭ Procfile・.env設定・起動確認 | 1h |
| ⑮ README / 運用手順書の作成 | 1h |
| ⑯ 最終確認・提出（GitHub＋Railway＋ChatGPT共有） | 0.5h |
| 合計作業時間：39時間（11日間）  最終提出予定日：2025年10月27日（月） | | | | | |