**Prodigenius – AI Powered Task Management Application**

**Project Domain / Category**

Mobile App + Machine Learning

**Abstract / Introduction**

Prodigenius is a streamlined, AI-powered (Artificial intelligence based) task management app built with Flutter, designed to help users prioritize tasks, manage schedules, and track productivity. The app uses free AI libraries and pre-built models for basic functionality, making it ideal to develop without complex custom AI models or cloud costs.

**Functional Requirements:**

The functional requirements of this project are given below:

1. **Task Input and Simple Categorization:**

Manual Input: Users can manually create and categorize tasks (e.g., work, study, personal) and assign due dates.

Basic Prioritization Input: Users can set task priorities manually or let the app suggest priorities using AI-based logic e.g. according to their execution time or urgency.

1. **Basic AI-Powered Task Prioritization:**

Library Used: Firebase ML Kit (On-Device)

AI Work: The app uses Firebase ML Kit’s basic logic to prioritize tasks based on due dates and urgency. It automatically adjusts task priority to suggest which tasks need attention first, especially as deadlines approach.

How It Works: The AI evaluates tasks using a few rules, such as how close the deadline is and how important the task is, and prioritizes tasks accordingly.

1. **Simple Task Duration Estimation:**

Library Used: TensorFlow Lite (Pre-trained Model)

AI Work: TensorFlow Lite’s pre-built model predicts how long tasks will take based on task type. The model estimates time by considering task categories (e.g., work tasks generally take longer than personal tasks).

How It Works: Users can view estimated task durations, which the app calculates based on pre-trained AI models that recognize task types.

1. **Basic AI-Driven Task Scheduling:**

Library Used: Firebase ML Kit (On-Device)

AI Work: Firebase ML Kit helps schedule tasks based on available time and task priorities. It recommends when to complete tasks by analyzing task importance and user availability.

How It Works: The AI suggests task schedules, such as doing quick tasks first or focusing on high-priority tasks earlier in the day.

1. **Simple Reminders & Notifications:**

Library Used: Flutter Local Notifications

AI Work: Smart notifications remind users about high-priority tasks or those approaching deadlines. Simple logic, not advanced AI, is used here to trigger reminders.

How It Works: Notifications alert users when tasks are nearing deadlines or when it's time to focus on important tasks.

1. **Basic Productivity Tracking:**

Library Used: Firebase ML Kit (On-Device)

AI Work: The app tracks users’ task completion rates and generates simple insights, such as the number of tasks completed daily or weekly. It offers basic AI-generated insights like: "You are most productive on Wednesdays."

How It Works: Firebase ML Kit identifies user patterns and provides simple productivity reports based on task completion trends.

1. **Progress Visualization & Task Dashboard:**

Library Used: Flutter Charts (No AI needed)

No AI Work Here: Use basic charts and visualizations to show tasks completed, pending, and overall productivity in a clean UI dashboard.

**Tech Stack:**

1. Flutter: Cross-platform development framework for Android and iOS.
2. Firebase (On-Device ML Kit): Provides on-device machine learning for AI-based prioritization, task scheduling, and productivity insights.
3. TensorFlow Lite: For running pre-trained AI models on the device (e.g., task duration estimation).
4. tflite\_flutter Plugin: Helps integrate TensorFlow Lite models with the Flutter app.
5. Flutter Local Notifications: Provides local notifications and reminders for task deadlines and priority alerts.

**Tools:**

1. **Flutter SDK** – For app development.

https://flutter.dev/docs/get-started/install

1. **Android Studio / Visual StudioCode** – IDEs for coding.

Android Studio: https://developer.android.com/studio

Visual Studio Code: https://code.visualstudio.com/

1. **Firebase Tools** – For authentication, on-device ML Kit, and notifications.

https://firebase.google.com/products/ml-kit

1. **TensorFlow Lite Model Maker (Optional)** – For customizing AI models.

https://www.tensorflow.org/lite/guide/model\_maker

1. **tflite\_flutter Plugin** – For TensorFlow Lite integration in Flutter.

https://pub.dev/packages/tflite\_flutter

1. **Flutter Local Notifications Plugin** – For sending task reminders and notifications.

https://pub.dev/packages/flutter\_local\_notifications

1. **Dart DevTools** – For debugging and performance analysis.

https://dart.dev/tools/dart-devtools

1. **Git** – For version control and collaboration.

https://git-scm.com/

1. **Postman (Optional)** – For testing APIs.

https://www.postman.com/downloads/

**Supervisor:**

Name: Muhammad Bilal

Email ID: [bilal.saleem@vu.edu.pk](mailto:bilal.saleem@vu.edu.pk)

Skype ID: bilalsaleem101