Time Buddy

Time Buddy is an intelligent, full-stack daily scheduling application designed to bridge the gap between a simple to-do list and a complex project manager. It uses an AI core to optimize a user's daily schedule and leverages machine learning to provide personalized suggestions for task priority and duration, adapting to the user's work habits over time.

Core Features

- **AI-Powered Scheduling:** Takes a list of tasks with priorities and constraints and generates an optimized daily timeline.
- **Intelligent Suggestions:** Learns from your completed tasks to predict the duration and suggest the priority for new tasks.
- **Productivity Tracking:** A visual calendar heatmap shows your daily consistency, motivating you to stay on track.
- **Modern Full-Stack Architecture:** Built with a React frontend, a Node.js backend for data management, and a Python backend for all AI and ML processing.
- **Seamless User Experience:** A beautiful, responsive interface with fluid animations.

Application Architecture

The application is built using a microservice architecture, with three distinct parts that communicate with each other:
[React Frontend] <--> [Node.js Backend API] <--> [Python AI & ML API]

Technology Stack

Component	Technologies Used
AI Core	Python, FastAPI, Scikit-learn, Pandas, Joblib
Backend & Database	Node.js, Express.js, SQLite, Axios
Frontend	React (with Vite), Framer Motion, Axios

Getting Started: Setup and Installation

To run this project, you will need to set up and run all three services. **You will need three separate terminal windows.**

Part 1: The Python AI Core (core_algorithm)

This service is the "brain" of the application.

1. **Navigate to the AI folder:**
   cd core_algorithm

2. **Create a Python Virtual Environment:** This creates an isolated environment for the Python packages.

python3 -m venv venv

3. **Activate the Virtual Environment:**
    - **On macOS/Linux:**
      source venv/bin/activate

    - **On Windows:**
      .\venv\Scripts\activate

   (You will see (venv) at the start of your terminal prompt.)
4. **Install Python Dependencies:**
   pip install fastapi "uvicorn[standard]" scikit-learn pandas joblib

Part 2: The Node.js Backend (timebuddy-backend)

This service handles data and communication.

1. **Navigate to the backend folder:**
   cd timebuddy-backend

2. **Install Node.js Dependencies:** This command will read the package.json file and install all necessary libraries.
   npm install

Part 3: The React Frontend (timebuddy-frontend)

This is the user interface.

1. **Navigate to the frontend folder:**
   cd timebuddy-frontend

2. **Install Node.js Dependencies:**
   npm install

Running the Application

All three servers must be running at the same time for the application to work.

1. **Terminal 1: Start the Python AI**
    - Navigate to core_algorithm.
    - Make sure your virtual environment is active (source venv/bin/activate).
    - Run the server:
      uvicorn main:app --reload

    - *This will run on http://127.0.0.1:8000.*
2. **Terminal 2: Start the Node.js Backend**

- Navigate to timebuddy-backend.
- Run the server:
  npm run dev

- *(Or nodemon index.js if you have it installed globally)*
- *This will run on http://localhost:3001.*

3. **Terminal 3: Start the React Frontend**
   - Navigate to timebuddy-frontend.
   - Run the app:
     npm run dev

   - *This will automatically open your web browser to http://localhost:5173 (or a similar port).*