

Требования к проекту «TaskMaster AI»

1. Введение

В современной динамичной среде студенты, фрилансеры и профессионалы сталкиваются с проблемой управления большим количеством задач, что часто приводит к стрессу, неэффективному распределению времени и профессиональному выгоранию. Стандартные списки дел (to-do lists) помогают отслеживать задачи, но не решают проблему оптимизации рабочей нагрузки.

Описание продукта:

TaskMaster AI — это интеллектуальная система управления личными задачами, разработанная как мобильное и веб-приложение. Основная цель продукта — помочь пользователю не просто отслеживать задачи, а визуализировать и оптимизировать свою недельную нагрузку для повышения продуктивности и предотвращения выгорания. Ключевой особенностью является интеграция с ИИ-ассистентом, способным анализировать расписание и выполнять команды пользователя по управлению задачами.

Границы проекта (чего продукт делать не будет):

Продукт не является инструментом для командной работы. Это исключительно персональный планировщик.

Продукт не является полнофункциональным календарём. Он не синхронизируется с внешними календарями (Google Calendar, Outlook) и не управляет событиями, не являющимися задачами.

Продукт не предназначен для сложного проектного менеджмента (например, диаграммы Ганта, управление зависимостями между задачами).

Продукт требует постоянного подключения к интернету для синхронизации данных и работы ИИ-ассистента.

2. Требования пользователя

Продукт будет взаимодействовать со следующими внешними системами и библиотеками:

Бэкенд-платформа Supabase:

Supabase Authentication: Для управления регистрацией, входом и сессиями пользователей через email/пароль и одноразовые коды (OTP).

Supabase Database (PostgreSQL): Для хранения всей пользовательской информации: профили, задачи, история общения с ИИ.

Supabase Edge Functions: Для безопасного выполнения серверной логики и взаимодействия с внешними API (в частности, с Google Gemini).

Google Gemini API:

Используется для обеспечения работы ИИ-ассистента. API будет получать отформатированный текстовый запрос (промпт), включающий контекст по задачам пользователя, и возвращать текстовый ответ или команду для выполнения.

Фреймворк Expo и библиотеки React Native:

Expo Router: Для организации навигации в приложении на основе файловой структуры.

@react-native-community/datetimepicker: Для предоставления нативного интерфейса выбора даты на мобильных устройствах.

@expo/vector-icons: Для использования стандартных иконок в интерфейсе.

Система взаимодействует с пользователем через графический интерфейс, состоящий из трех основных экранов, управляемых через панель вкладок:

Экран "Tasks" (Задачи):

Основной экран, отображающий горизонтально прокручиваемый недельный календарь.

Каждый день представлен в виде колонки, содержащей список задач, запланированных на этот день.

Над календарем расположены элементы навигации для переключения между неделями.

Под названием дня отображается индикатор выгорания (Burnout Indicator).

В правом нижнем углу находится плавающая кнопка (+) для создания новой задачи.

Экран "AI Assistant" (ИИ-Ассистент):

Представляет собой интерфейс чата с текстовым полем для ввода и областью для отображения истории сообщений.

Экран "Profile" (Профиль):

Отображает информацию о пользователе (email), статистику по задачам и предоставляет функционал для выхода из системы.

Таблица "Действие пользователя - Реакция системы":

Действие пользователя	Реакция системы
Открывает приложение (не авторизован)	Отображается экран входа/регистрации.
Регистрируется с новым email	На почту приходит письмо с 6-значным кодом.
Вводит email и пароль для входа	Перенаправляет на главный экран "Tasks".
Нажимает плавающую кнопку (+)	Открывается модальное окно "Новая задача"
Нажимает на существующую задачу	Открывается модальное окно "Редактировать задачу" с заполненными данными.
Нажимает на поле даты в модальном окне	На мобильном устройстве открывается нативный календарь; в веб-версии отображается текстовое поле.
Отправляет сообщение ИИ ("создай задачу...")	Система отправляет запрос на сервер, получает ответ, создает задачу и отображает подтверждение в чате. Календарь обновляется.
Нажимает "Выйти из системы"	Отображается диалог подтверждения, после чего система перенаправляет на экран входа.

Целевая аудитория — это индивидуальные пользователи, стремящиеся к повышению личной продуктивности.

•Группы пользователей: Студенты, фрилансеры, менеджеры, специалисты интеллектуального труда.

•Образование и опыт: Уровень образования не имеет значения. Предполагается, что пользователь имеет опыт использования современных мобильных приложений и понимает базовые концепции управления задачами (дедлайн, важность).

•Техническая грамотность: Средняя. Пользователь должен уметь устанавливать приложения из магазинов, регистрироваться по

email и взаимодействовать со стандартными элементами интерфейса. Специальных технических знаний не требуется.

- Предположение: Пользователь имеет постоянный доступ к сети Интернет для работы приложения.

- Зависимость: Функциональность приложения напрямую зависит от доступности и стабильной работы сервисов Supabase и Google Gemini API.

- Зависимость: Для работы ИИ-ассистента необходим валидный и активный API-ключ, сконфигурированный на сервере.

- Предположение: Пользователь имеет валидный адрес электронной почты для регистрации и получения кодов подтверждения.

3. Системные требования

Аутентификация пользователя

Система должна предоставлять возможность регистрации нового пользователя с помощью email, пароля и подтверждения через ОТР-код.

- Система должна предоставлять возможность входа для существующего пользователя по email и паролю.

- Система должна предоставлять возможность выхода из учетной записи.

- Система должна сохранять сессию пользователя между запусками приложения.

Управление задачами

- Пользователь должен иметь возможность создавать новую задачу.

- При создании задачи пользователь должен иметь возможность указать:

- Название (обязательно).
- Описание (опционально).
- Запланированную дату.
- Дедлайн (опционально).
- Важность (Низкая, Средняя, Высокая).
- Сложность (от 1 до 5).

- Пользователь должен иметь возможность просматривать все свои задачи в недельном календаре.

- Пользователь должен иметь возможность редактировать любое поле существующей задачи.

- Пользователь должен иметь возможность удалять задачу.
- Пользователь должен иметь возможность отмечать задачу как "выполненную" и снимать эту отметку.

Визуализация и навигация

- Система должна отображать задачи в виде колонок, соответствующих 7 дням текущей недели.
- Пользователь должен иметь возможность переключаться на предыдущую и следующую недели.
- Система должна подсвечивать просроченные задачи (красным цветом).
- Система должна подсвечивать задачи, дедлайн которых наступает на следующий день (оранжевым цветом).

Индикатор выгорания

- Для каждого дня в календаре система должна отображать графический индикатор рабочей нагрузки ("выгорания").
- Значение индикатора должно рассчитываться на основе суммы "сложностей" всех незавершенных задач на данный день.
- Индикатор должен автоматически пересчитываться после создания, изменения, выполнения или удаления задачи.

ИИ-Ассистент

- Система должна предоставлять интерфейс чата для взаимодействия с ИИ-ассистентом.
- ИИ-ассистент должен быть способен понимать запросы на естественном языке (русском).
- ИИ-ассистент должен иметь возможность создавать новую задачу по команде пользователя.
- ИИ-ассистент должен иметь возможность изменять запланированную дату существующей задачи по команде пользователя.

Профиль пользователя

Система должна отображать статистику по задачам пользователя (общее количество, выполнено, в процессе, просрочено).

Надёжность (Reliability):

Важность: Пользователи доверяют приложению свои планы. Потеря данных или некорректная работа могут привести к срыву сроков.

Измерение: Доступность бэкенд-сервисов должна быть $> 99.5\%$. Частота сбоев приложения на стороне клиента должна быть $< 1\%$ от всех сессий.

Безопасность (Security):

Важность: Приложение хранит персональные данные (email) и информацию о личных и рабочих планах пользователя.

Измерение: Все внешние API-ключи (Google Gemini) должны храниться исключительно на сервере и не быть доступны в клиентском коде. Взаимодействие с бэкендом должно происходить по защищенному протоколу HTTPS. Политики безопасности базы данных (RLS) должны гарантировать, что пользователь может получить доступ только к своим данным.

Производительность (Performance):

Важность: Приложение должно быть отзывчивым, чтобы не мешать пользователю быстро фиксировать и организовывать свои мысли.

Измерение: Время загрузки основного экрана с задачами не должно превышать 2 секунды на среднем мобильном устройстве при стабильном интернет-соединении. Ответ от ИИ-ассистента должен приходить в среднем за 5 секунд.

Удобство использования (Usability):

Важность: Так как продукт нацелен на повышение продуктивности, он сам не должен быть сложным в освоении.

Измерение: Новый пользователь должен быть способен создать свою первую задачу менее чем за 30 секунд без необходимости обращаться к инструкции. Процесс изменения даты задачи должен занимать не более 3 кликов/тапов.