

(MVP) خلاصه پروژه

بک‌اند با سه فیچر: (1) اولویت‌بندی تسک‌ها، (2) چت‌بات با حافظه و اتصال به MVP محصول: اپلیکیشن «فرم» - نسخه پروژه/افراد/لایبرری، (3) یادآوری/نوتیف با صدا

هدف: پیاده‌سازی سریع، تمیز و قابل‌دمو در کمتر از 2 هفته. تحویل به‌صورت کد قابل‌دبیلوی + مستندات

Push و REST/JSON API طراحی، OpenAI API فریلنسر/تیم بک‌اند باتجربه در ادغام RFP مخاطب این Notifications (FCM/APNs/Web Push).

محدوده (Scope)

1) اولویت‌بندی تسک‌ها (Prioritizer)

- ورودی: لیست تسک‌های امروز کاربر + متادیتا (ددلاین، برآورد زمان، انرژی، وابستگی‌ها)
- تولید خروجی ساختیافته + (function/tool calling) سرور Gateway از طریق (OpenAI) پردازش: تماس با مدل
- tips و (when) پیشنهاد زمان اجرای بلوک (why) دلیل A+/A/B/C/N شامل امتیاز JSON: خروجی
- Focus Block اکشن: امکان «اعمال» امتیازها روی تسک‌ها + ساخت

2) چت‌بات با حافظه (Chatbot with Memory & Linking)

- (جلسه فعلی با خلاصه خویکار) Thread Memory، (بلندمدت) Profile Memory: حافظه دو لایه
- task | note | question | reminder | share | assign | other. تشخیص نیت
- استخراج موجودیت‌ها: پروژه‌ها، افراد، تگ‌ها
- (inbox/docs/ snippets/quotes) اکشن‌های ساختیافته: لینک‌کردن پیام به پروژه/افراد، ساخت تسک از چت، ذخیره در لایبرری (Email/Telegram/Workspace) اشتراک در کانال، Reminder ایجاد

3) یادآوری/نوتیف با صدا (Reminders & Notifications)

- تکرار (اختیاری)، کانال نوتیف، Snooze، ایجاد، لغو
- Push (sound_id) انتخاب صدای اعلان + (وب/موبایل) Push

خارج از محدوده (Out of Scope)

- (نمونه پاسخ‌ها + Postman/Insomnia برای تست endpoint فقط چند). فرانت‌اند کامل
- کفایت می‌کند Bearer ساده Token فقط (Social/OAuth). احراز هویت پیچیده
- چندزبانه و چندپلتفرم پیچیده

الزامات فنی

- اولویت با). Go (Fiber/Gin) یا Python (FastAPI) یا Node.js (Express/Fastify) زبان/فریم‌ورک بک‌اند FastAPI یا Express
- ORM (Prisma/SQLAlchemy/GORM) سبک MySQL یا (اولویت) PostgreSQL: پایگاه داده
- تمام درخواست‌ها از سرور (کلید در سرور، نه کلاینت) OpenAI: مدیریت
- معماری: ماژولار، آماده توسعه. لاگبرداری و هندل خطا الزامی
- env نمونه. فایل Dockerfile + docker-compose: دبیلوی
- Push: FCM حداقل MVP در iOS برای APNs، برای اندروید/فایر‌بیس

مدل داده (حداقلی)

tasks

id (uuid), title, description, project_id, assignee_id, deadline (timestampz), estimate_min (int), energy (1..5), priority (enum: Aplus|A|B|C|N), deps (jsonb), created_at, updated_at

projects

id, name, status (active|archived), members (jsonb), created_at, updated_at

people

id, name, role, contacts (jsonb), created_at, updated_at

chat_messages

id, user_id, text, reply_text, meta (jsonb), created_at

links (message ↔ project/person/library)

id, message_id, project_id?, person_id?, library (inbox|docs|snippets|quotes)?, created_at

reminders

id, target_type (msg|task|note), target_id, when (timestampz), repeat (cron/null), sound_id (focus_bell_01|urgent_ping_02|soft_chime_03), channel (push|email|telegram), status, created_at

های مورد نیاز (نمونه قرارداد)

Auth ساده

POST /auth/token -> {token}

Prioritizer

POST /ai/prioritize

Req: { user_id, date, tasks: [{task_id, title, deadline, estimate_min, energy, deps[]}], calendar?: {...}, energy_profile?: {...} }

Res: {
 date, prioritized: [{task_id, score: "A+|A|B|C|N", why, when?}],
 tips: [{task_id, tip}]
}

POST /tasks/:id/apply-priority -> {priority}

POST /focus-blocks -> {task_id, start, end}

Chatbot

POST /chat/send

Req: { user_id, text, thread_id?, context_opt?: {project_id?, person_id?} }

Res: {
 reply_text,
 actions: [
 {type:"LINK_ENTITIES", projects:[], people:[]},
 {type:"SAVE_TO_LIBRARY", lib:"snippets"},
 {type:"SUGGEST_TASK", title, project, assignee, due?},
 {type:"CREATE_REMINDER", text, when}
],
}

```
confidence: 0.xx,  
memory_updates?: {profile?: {}, thread_summary?: "..."}  
}
```

بدنه اکشن انتخابی و اعمال روی دیتابیس -> POST /chat/confirm-action

Reminders

POST /reminders

Req: { target_type, target_id, text, when, repeat?, notify_channel, sound_id }

Res: { id, status }

POST /reminders/:id/snooze -> {+minutes}

DELETE /reminders/:id -> {status}

OpenAI (Gateway) ادغام با

- با این توابع ai-gateway یک سرویس داخلی:
 - call_prioritizer(tasks, calendar, energy) -> JSON (system prompt آماده + structured output)
 - chat_with_memory(user_id, text, thread_id) -> {reply_text, actions[], memory_updates}
- معتبر JSON مدیریت کلید در سرور. خروجی‌ها همیشه.

پذیرش کار (Acceptance Criteria)

- Prioritizer:** روی Apply معتبر با حداقل 80% پرشدگی فیلدها + امکان JSON با ورودی نمونه 10 تسک، خروجی: 5 تسک
- Chatbot:** درست تشخیص داده شود و حداقل یک اکشن معتبر پیشنهاد گردد intent برای 10 پیام نمونه (LINK/SAVE/REMINDER/TASK).
- Reminders:** لحاظ Push در پاسخ sound_id کار کند؛ پوش وب برای اندروید/وب تست شود؛ Snooze/ایجاد/لغو.
- (env + کالکشن) Postman پایدار: هندل خطای واضح، لاگ سرور، تست‌های.
- ها پاسخ بدهند endpoint بالا بیاید و همه Docker Compose: دیپلوی.

تحویل‌ها (Deliverables)

- و دستورالعمل دیپلوی README + (Git) سورس‌کد.
- Dockerfile/Compose + نمونه env فایل‌های.
- (اختیاری) Diagrams + مایگريشن SQL اسکیمای.
- با نمونه درخواست/پاسخ Postman/Insomnia کالکشن.
- به‌صورت فایل (prioritizer/chatbot) System prompts.

زمان‌بندی پیشنهادی (مایلستون)

- ساده Auth + روز 1-2: دیتامدل + اسکلت پروژه.
- Prioritizer (Endpoint + Gateway + Apply): روز 3-5.
- Chatbot (Memory + سیک Actions): روز 6-8.
- Reminders (Scheduler + Push FCM) + Snooze: روز 9-10.
- روز 11-12: تست/سخت‌گیرانه + مستندات + تحویل نهایی.
- اگر مجری قبلاً این مسیر را رفته، زمان به 6-8 روز کاهش می‌یابد.

الزامات غیرعملکردی

- عدم نشت کلیدها، ورودی‌زدایی، .env: امنیت پایه
- header بر info/error + trace id لاگ و مانیتورینگ: سطح
- کیفیت کد: ماژولار، حداقل 10 تست واحد/یکپارچه

معیارهای ارزیابی پیشنهادها

- نمونه کار + OpenAI/LLM سابقه ادغام
- کیفیت پاسخ به پرسش‌های فنی زیر (کوتاه، دقیق)
 1. معتبر از مدل چه راهکاری می‌دهید؟ JSON برای تضمین
 2. را چطور انجام می‌دهید که مقیاس‌پذیر باشد؟ Memory طراحی
 3. را چطور هندل می‌کنید؟ OpenAI مدیریت خطا و ریتیمیت
 4. را روی وب/اندروید چطور تست و تحویل می‌دهید؟ Push
 5. چه لاگ/متریکی برای پذیرش و پایش پیشنهاد می‌دهید؟

الگوی قیمت‌گذاری پیشنهادی (پر شود)

- تومان ... MVP هزینه کل
- مدت: ... روز کاری
- تقسیم مایلستون و پرداخت
 - شروع کار 30%
 - Prioritizer + Chatbot بعد از 40%
 - Reminders پس از 30% تحویل نهایی
- کارانتی رفع باگ: ... روز پس از تحویل

نکات قراردادی

- مالکیت کامل کد و مستندات برای کارفرما
- (ساده NDA) تعهد به محرمانگی
- عدم استفاده از کلید یا حساب کارفرما خارج از پروژه

ضمائم (اختیاری)

- Chatbot و Prioritizer نمونه ورودی/خروجی (JSON).
- های اولیه sound_id لیست
- نمونه سناریوهای تست (10 پیام چت، 10 تسک امروز)