

# Trang 5.2: Lập kế hoạch dự án chi tiết (Blueprint) với /plan.

## 1. Mục đích và triết lý

`/plan` là lệnh kích hoạt chế độ "**Lập kế hoạch thuần túy**" — AI trở thành `project-planner` và nhiệm vụ duy nhất là tạo ra một kế hoạch chi tiết, không viết một dòng code nào.

“**Quy tắc sắt:** `/plan` kết thúc = Chỉ có file kế hoạch `PLAN.md`. Không có code, không có implementation. Đây là ranh giới nghiêm ngặt.

## 2. Khi nào nên dùng `/plan`?

### ☐ Nên dùng khi:

- Bắt đầu một dự án hoặc tính năng lớn (hơn 2-3 ngày công).
- Cần trình bày kế hoạch cho đội hoặc cấp trên phê duyệt.
- Muốn phân rã công việc thành các task nhỏ có thể giao cho thành viên.
- Cần ước tính thời gian và phụ thuộc giữa các tác vụ.

## 3. Quy trình thực hiện (4 bước)

### Bước 1: Khởi động lệnh

```
bash
```

```
/plan [mô tả ngắn gọn về dự án hoặc tính năng]
```

Ví dụ:

```
bash
```

```
/plan hệ thống quản lý nhân sự với đăng nhập SSO,
```

```
phân quyền RBAC và báo cáo Excel
```

## Bước 2: Trả lời Câu hỏi Socratic

AI (`project-planner`) sẽ đặt một loạt câu hỏi làm rõ yêu cầu trước khi lên kế hoạch. Đây là **Cổng Socratic (Socratic Gate)** — bước bắt buộc để đảm bảo kế hoạch đúng ngay từ đầu.

Các câu hỏi thường xoay quanh:

- **Người dùng và quy mô:** Bao nhiêu người dùng? Loại người dùng nào?
- **Tech stack:** Đang dùng framework gì? Database nào?
- **Ràng buộc:** Deadline? Budget? Team size?
- **Tích hợp:** Có cần kết nối hệ thống nào khác không?
- **Ưu tiên:** MVP tối thiểu là gì? Tính năng nào có thể trì hoãn?

## Bước 3: AI lên kế hoạch

Sau khi bạn trả lời đủ câu hỏi, AI tổng hợp thông tin và tạo ra file kế hoạch.

## Bước 4: Xem xét và chỉnh sửa kế hoạch

File kế hoạch không phải là tài liệu cố định. Chỉnh sửa trực tiếp `PLAN.md` để:

- Điều chỉnh ưu tiên.
  - Thêm/bớt tính năng.
  - Phân công người phụ trách từng task.
-

# 4. Cấu trúc file kế hoạch đầu ra ( `PLAN-{tên}.md` )

File kế hoạch thường được tạo tại `docs/PLAN-tên-dự-án.md` với nội dung đầy đủ:

```
markdown
```

```
# Kế hoạch: Hệ thống Quản lý Nhân sự
```

```
## Tổng quan
```

- Mô tả ngắn gọn về dự án
- Tech stack: Next.js 14, PostgreSQL, Prisma, NextAuth

```
## Phân tích yêu cầu
```

- Người dùng: HR Admin, Manager, Employee (3 vai trò)
- Quy mô: ~500 nhân viên, 50 manager
- ...

```
## Kiến trúc hệ thống
```

```
[Sơ đồ kiến trúc đơn giản]
```

```
## Phân rã tác vụ
```

```
### Giai đoạn 1: Nền tảng (Sprint 1-2)
```

- [ ] Thiết lập dự án Next.js
- [ ] Cấu hình database schema
- [ ] Triển khai SSO với NextAuth
- [ ] RBAC cơ bản (3 vai trò)

```
### Giai đoạn 2: Tính năng cốt lõi (Sprint 3-4)
```

- [ ] Module quản lý nhân viên (CRUD)

- [ ] Module chấm công
- [ ] Module nghỉ phép

### ### Giai đoạn 3: Báo cáo (Sprint 5)

- [ ] Dashboard tổng hợp
- [ ] Export Excel

### ## Phụ thuộc tác vụ

- RBAC cần hoàn thành trước các module nghiệp vụ
- Dashboard cần dữ liệu từ tất cả module khác

### ## Ước tính thời gian

- Tổng: ~5 sprint (10 tuần)
- MVP (Giai đoạn 1+2): 8 tuần

### ## Agent phụ trách

- database-architect: Schema thiết kế
- backend-specialist: API + Business logic
- frontend-specialist: UI/UX
- security-auditor: Review xác thực và phân quyền

## 5. Ví dụ phiên làm việc thực tế

Bạn: /plan tích hợp cổng thanh toán VNPAY vào ứng dụng Next.js của tôi

AI (project-planner): Tôi cần làm rõ một số điểm để lên kế hoạch chính xác:

1. Ứng dụng của bạn hiện đang ở giai đoạn nào?

(MVP mới, đang hoạt động, hay đang scale)

2. Bạn cần hỗ trợ những loại giao dịch nào?

(Thanh toán 1 lần, đăng ký định kỳ, hay cả hai)

3. Tech stack backend hiện tại là gì?

(Next.js API Routes, Express, hay gì khác)

4. Bạn cần xử lý hoàn tiền (refund) không?

5. Timeline mong muốn?

Bạn:

1. Đang hoạt động, có ~500 đơn hàng/tháng

2. Thanh toán 1 lần cho đơn hàng

3. Next.js API Routes + PostgreSQL

4. Cần hoàn tiền một phần và toàn bộ

5. 2 tuần

AI: Tôi đã hiểu đủ thông tin. Đây là kế hoạch:

[Tạo file docs/PLAN-vnipay-integration.md với đầy đủ nội dung...]

Đã tạo kế hoạch tại docs/PLAN-vnipay-integration.md

Kế hoạch bao gồm 3 giai đoạn, ước tính 8 ngày công.

Bạn có muốn điều chỉnh gì không?

## 6. Mẹo lập kế hoạch hiệu quả

### Trả lời đầy đủ câu hỏi Socratic

Câu hỏi càng chi tiết, kế hoạch càng chính xác. Đừng trả lời ngắn gọn như "bình thường" hay "tùy bạn".

### Định nghĩa MVP rõ ràng

"MVP tối thiểu cho tôi là: có thể đăng nhập, xem danh sách nhân viên và thêm mới. Các tính năng còn lại có thể phát hành sau."

## Chỉn sửa kế hoạch ngay

Sau khi AI tạo kế hoạch:

- Mở file `PLAN.md` và đọc kỹ.
- Thêm người phụ trách (`@team-member`) vào từng task.
- Điều chỉnh độ ưu tiên nếu cần.

## Commit kế hoạch vào Git

```
bash
git add docs/PLAN-tên-dự-án.md
git commit -m "plan: add feature X implementation plan"
```

Điều này tạo lịch sử quyết định cho cả đội có thể tham khảo sau này.

# 7. Sau khi có kế hoạch: Bước tiếp theo

Kế hoạch được phê duyệt

|



Triển khai:

/create [task đầu tiên trong kế hoạch]

hoặc

/enhance [tính năng cụ thể cần thêm vào]

Phiên bản #2

Được tạo 2026-03-04 06:11:46 UTC bởi Nam Đặng

Được cập nhật 2026-03-04 09:26:10 UTC bởi Nam Đặng