

# Trang 5.3: Xây dựng ứng dụng/tính năng mới từ đầu với /create.

## 1. Mục đích

`/create` đóng vai trò như một **trình thuật sĩ tạo ứng dụng (Application Creation Wizard)** — dẫn dắt bạn qua một cuộc hội thoại tương tác để hiểu yêu cầu, lên kế hoạch kiến trúc và điều phối các Agent chuyên gia cùng nhau xây dựng ứng dụng hoàn chỉnh.

### “ Điểm khác biệt quan trọng:

- `/create` = Xây dựng từ **trang giấy trắng (greenfield)**.
- `/enhance` = Thêm vào hệ thống **đang hoạt động (brownfield)**.

## 2. Khi nào nên dùng `/create`?

### □ Sử dụng `/create` khi:

- Bắt đầu một dự án hoàn toàn mới.
- Tạo một module/service độc lập mới trong dự án lớn.
- Cần tạo nhanh một prototype để kiểm tra ý tưởng.
- Muốn triển khai một tính năng phức tạp từ đầu mà không lo ảnh hưởng code cũ.

## 3. Quy trình thực hiện (3 bước)

# Bước 1: Mô tả ứng dụng cần tạo

```
bash
/create [mô tả ứng dụng và các yêu cầu chính]
```

Cung cấp thông tin càng đầy đủ thì AI tạo ra code càng chính xác với nhu cầu của bạn.

# Bước 2: Xem xét và phê duyệt kế hoạch

Agent `app-builder` sẽ tổng hợp yêu cầu và đề xuất:

- **Tech stack:** Framework, database, thư viện chính.
- **Cấu trúc thư mục:** Cách tổ chức code.
- **Database schema:** Các bảng và quan hệ chính.
- **Danh sách tính năng:** Những gì sẽ được xây dựng.

Bạn cần phê duyệt (Y) hoặc yêu cầu điều chỉnh trước khi AI bắt đầu viết code.

# Bước 3: Các Agent cùng xây dựng

Sau khi phê duyệt, `orchestrator` sẽ điều phối các Agent chuyên gia:

```
orchestrator điều phối:
├─ database-architect → Tạo database schema và migrations
├─ backend-specialist → Viết API endpoints và business logic
├─ frontend-specialist → Xây dựng UI components và pages
└─ test-engineer → Tạo bộ test cơ bản
```

# 4. Ví dụ phiên làm việc thực tế

## Ví dụ: Tạo ứng dụng quản lý công việc (Todo)

```
☐☐ Bạn: /create ứng dụng todo có dark mode, xác thực người dùng,
phân loại task theo dự án, và có thể chia sẻ task với người khác
```

AI (app-builder):

Kế hoạch xây dựng:

Tech Stack:

- Framework: Next.js 14 (App Router)
- Database: PostgreSQL + Prisma ORM
- Auth: NextAuth.js (Email + Google OAuth)
- UI: Tailwind CSS + shadcn/ui
- State: Zustand

Database Schema:

- User (id, email, name, avatar)
- Project (id, name, color, owner\_id)
- Task (id, title, status, due\_date, project\_id, assignee\_id)
- ProjectMember (project\_id, user\_id, role)

Tính năng sẽ tạo:

- Đăng ký/đăng nhập (Email + Google)
- Dark mode toggle
- CRUD Projects
- CRUD Tasks với drag-and-drop
- Chia sẻ project và gán task
- Bộ test cơ bản

Bạn có muốn tiến hành không? (Y/N)

Bạn: Y (nhưng dùng MySQL thay vì PostgreSQL)

AI: Đã điều chỉnh sang MySQL. Đang bắt đầu xây dựng...

☐ database-architect: Đang tạo schema Prisma cho MySQL...

⚙️ backend-specialist: Đang tạo API routes...

☐ frontend-specialist: Đang xây dựng components...

☐ test-engineer: Đang viết test cases...

☐ Hoàn thành! Chạy lệnh sau để khởi động:

```
npm install && npx prisma migrate dev && npm run dev
```

## 5. Cấu trúc dự án được tạo ra (ví dụ Next.js)

```
my-todo-app/  
├─ app/  
│ ├─ (auth)/  
│ │ ├─ login/page.tsx  
│ │ └─ register/page.tsx  
│ └─ dashboard/  
│ │ ├─ layout.tsx  
│ │ └─ page.tsx  
│ └─ projects/[id]/page.tsx  
└─ api/  
    └─ auth/[...nextauth]/route.ts  
        └─ projects/route.ts  
            └─ tasks/route.ts  
├─ components/  
│ ├─ TaskCard.tsx  
│ └─ ProjectList.tsx  
│   └─ ThemeToggle.tsx  
├─ lib/  
│ └─ db.ts (Prisma client)  
│   └─ auth.ts (NextAuth config)  
├─ prisma/  
│ └─ schema.prisma  
└─ tests/
```

```
| └─ api.test.ts
| └─ components.test.tsx
└─ package.json
```

## 6. Mẹo để `/create` tạo ra kết quả tốt nhất

### Mô tả cụ thể và đầy đủ

# Thiếu thông tin:

`/create` ứng dụng bán hàng

# Đầy đủ thông tin:

`/create` ứng dụng thương mại điện tử bán quần áo thời trang.

Người dùng có thể duyệt sản phẩm, thêm vào giỏ hàng,

thanh toán qua VNPay. Admin quản lý sản phẩm và đơn hàng.

Tech stack: Next.js + PostgreSQL + Stripe (nếu cần quốc tế).

MVP không cần: review, wishlist, loyalty points.

## Bắt đầu từ MVP

Tập trung vào tính năng cốt lõi. Sau khi có MVP chạy được, dùng `/enhance` để thêm dần.

1. `/create` [core features only] → MVP
2. `/enhance` [feature A] → Version 1.1
3. `/enhance` [feature B] → Version 1.2

## Phê duyệt lần lượt từng thành phần

Nếu muốn kiểm soát chặt chẽ hơn, sau khi nhận kế hoạch:

"Hãy tạo database schema trước để tôi review,  
sau đó mới tiếp tục với backend."

## 7. Sau khi `/create` hoàn thành

Bước tiếp theo	Lệnh
Kiểm tra ứng dụng	<code>/preview start</code>
Chạy tests	<code>/test</code>
Thêm tính năng mới	<code>/enhance [tính năng]</code>
Kiểm tra bảo mật	Hỏi <code>@security-auditor review code này</code>
Triển khai	<code>/deploy</code>

Phiên bản #2

Được tạo 2026-03-04 06:11:55 UTC bởi Nam Đặng

Được cập nhật 2026-03-04 09:28:48 UTC bởi Nam Đặng