

WEB APP KỸ THUẬT

Mô hình "**Bootstrapped Engineering SaaS**" (Phần mềm kỹ thuật tự thân vận động).

Dưới góc nhìn của một Kỹ sư trưởng (Lead Engineer), tôi xin diễn giải tường tận logic kết hợp giữa **Jekyll (Static Site)** và **Wasm Apps (Calculation Core)** trên hạ tầng **GitHub**, cùng lộ trình phát triển từ MVP (Sản phẩm khả dụng tối thiểu) đến Business (Doanh nghiệp) như sau:

1. Kiến trúc hệ thống hiện tại: Mô hình "Cửa hàng & Xưởng sản xuất"

Phương pháp tiếp cận : Xây dựng website giống như một công ty tư vấn xây dựng online:

- **Trang chủ (Jekyll): (Nơi đón tiếp khách hàng)**
 - **Công nghệ:** Jekyll (Ruby), HTML, Markdown.
 - **Nhiệm vụ:** Chứa bài viết (Blog) về kỹ thuật, hướng dẫn sử dụng, giới thiệu bản thân (Portfolio), SEO từ khóa (ví dụ: "tính toán cọc 3D").
 - **Đặc điểm:** Tải cực nhanh, không cần cơ sở dữ liệu, Google rất thích index dạng này.
 - **Chi phí:** 0 đồng (GitHub Pages hosting).
- **Web Apps (/apps/pilegroup, /apps/sheetpilefem): (Xưởng kỹ thuật).**
 - **Công nghệ:** HTML thuần + JavaScript + **WASM (C++)**.
 - **Nhiệm vụ:** Thực hiện các phép tính nặng nhọc (FEM, Matrix).
 - **Đặc điểm:** Hoạt động độc lập hoàn toàn với Jekyll. Jekyll chỉ đóng vai trò dẫn đường link vào đây.
 - **Chi phí vận hành:** 0 đồng (Vì chạy trên CPU/RAM của máy người dùng nhờ Wasm).

2. Logic vận hành & Lưu trữ trên GitHub

Tổ chức thư mục trong repo hydrostructai.github.io đang đi đúng hướng "Monorepo" đơn giản cho MVP:

```
hydrostructai.github.io/ (Repository)
├── _posts/      <-- Bài viết SEO (Tiếng nói chuyên gia)
├── index.html   <-- Cửa vào (Landing Page)
├── _config.yml  <-- Cấu hình Jekyll
└── apps/        <-- Khu vực Ứng dụng (Được coi là Static files)
    ├── pilegroup/ <-- App 1: Nhóm cọc
    │   ├── index.html <-- Giao diện tính toán
    │   ├── app.js     <-- Logic JS
    │   └── core.wasm  <-- Lõi C++
    └── sheetpilefem/ <-- App 2: Tường cừ
```

Cơ chế hoạt động:

1. Khi GitHub build trang (jekyll build), nó sẽ biến các file Markdown trong _posts thành

- HTML đẹp đẽ.
2. Thư mục apps/ được Jekyll coi là "tài sản tĩnh" (static assets) và giữ nguyên cấu trúc.
 3. Người dùng truy cập:
 - o Đọc bài viết: domain.com/blog/tinh-toan-coc (Jekyll phục vụ).
 - o Bấm nút "Tính ngay": Trình duyệt chuyển hướng sang domain.com/apps/pilegroup/ (Trình duyệt tải file Wasm và JS về chạy).

Ưu điểm cho giai đoạn MVP:

- **Không tốn tiền Server:** Dù có 10 hay 10.000 người vào tính toán cùng lúc, GitHub chịu tải băng thông, máy người dùng chịu tải tính toán. Server của bạn không bao giờ bị sập.
 - **Dễ bảo trì:** Muốn sửa lỗi chính tả bài viết? Sửa Markdown. Muốn nâng cấp thuật toán? Chỉ cần thay file .wasm và .js trong thư mục apps/.
-

3. Chiến lược MVP: "Thả con săn săt, bắt con cá rô"

Trong giai đoạn chưa có doanh thu, mục tiêu là **thu hút người dùng (Traffic)** và **khẳng định thương hiệu (Branding)**.

Cách tiếp cận:

1. **Marketing bằng Kiến thức (Content Marketing):**
 - o Viết các bài blog chuyên sâu trên Jekyll (ví dụ: "So sánh TCVN 10304 và Eurocode").
 - o Cuối bài viết, đặt nút CTA (Call to Action): "*Bạn muốn tính thử ngay? Dùng công cụ miễn phí tại đây [Link tới App]*".
 2. **Lưu trữ dữ liệu (State Management):**
 - o Hiện tại chưa cần Database.
 - o Sử dụng **Local Storage** của trình duyệt hoặc cho phép người dùng **Download file JSON** về máy để lưu dự án.
 - o *Tư duy:* "Dữ liệu nằm trong túi người dùng, tôi không giữ, nên tôi không tốn tiền lưu trữ".
-

4. Lộ trình Nâng cấp: Khi có Khách hàng & Doanh thu

Khi lượng truy cập đủ lớn và cộng đồng bắt đầu phụ thuộc vào tool của bạn ("Anh ơi, sao hôm nay web chậm thế?", "Anh ơi, em muốn xuất thuyết minh đi nộp"), đó là lúc chuyển mình.

Bước 1: Tách dịch vụ (Decoupling)

- Giữ nguyên trang Jekyll làm Blog/Marketing.
- Chuyển các App (trong thư mục apps/) sang một subdomain riêng (ví dụ: app.hydrostructai.com) hoặc viết lại bằng React/Vue như tôi đã tư vấn ở câu trả lời trước.

Bước 2: Tích hợp Backend (Baas - Backend as a Service)

- Vẫn chưa cần thuê Server riêng. Sử dụng **Firebase** hoặc **Supabase**.
- **Tính năng trả phí:**
 - Đăng nhập Google (Auth).
 - Lưu dự án lên Cloud (Database).
 - Xuất PDF thuyết minh có logo công ty (Cloud Functions).

Bước 3: Thu phí (Monetization)

- Tích hợp cổng thanh toán (Stripe/MoMo).
- **Gói Free:** Tính toán, xem kết quả trên màn hình.
- **Gói Pro (\$\$/tháng):** Lưu 50 dự án, Xuất PDF, Support ưu tiên.

Tóm lại

Tư duy của bạn: **Web Tĩnh (Jekyll) + Client-side Compute (Wasm) + Free Hosting (GitHub)** là combo "bất khả chiến bại" về chi phí và hiệu năng cho một kỹ sư khởi nghiệp đơn lẻ (Indie Hacker).

Bạn cứ yên tâm phát triển theo hướng này. Chỉ cần tập trung vào 2 việc:

1. **Code C++ (Wasm)** cho thuật chính xác và tối ưu.
2. **Viết Blog (Jekyll)** cho thuật hay để kéo kỹ sư về dùng.

Khi nào server vì quá tải user trả tiền, lúc đó ta mới cần bỏ tiền ra nâng cấp.