**Profoliofy — Step-by-Step MVP Completion Plan (Template-Based Model)**

**Goal:**

Build and launch a functional MVP of Profoliofy — a template-based AI-integrated portfolio generator for two primary audiences: tech professionals and creative artists/writers. The MVP should support user registration, portfolio generation via AI prompt, template rendering, basic analytics, and monetization.

**Phase 1: UI/UX Design & Frontend Setup (Week 1–3)**

**Tasks:**

**1. Frontend Boilerplate Setup (Day 1–2)**

* Initialize React project with Vite
* Install Tailwind CSS
* Install core libraries:
  + react-router-dom (routing)
  + axios (API requests)
  + zustand (state management)
  + framer-motion (animations)
* Setup basic project structure:
  + /pages
  + /components
  + /templates
  + /services

**2. Routing & Navigation (Day 3)**

* Create pages:
  + Home
  + Dashboard
  + Login
  + Register
  + Portfolio Preview
  + NotFound
* Add protected routing logic

**3. Authentication UI (Day 4–5)**

* Build Login and Register pages
* Connect to Supabase Auth for real user signup
* Handle basic error and success messages

**4. Dashboard Skeleton (Week 2)**

* Left-side vertical navbar
* Sections:
  + Profile Settings
  + Prompt Generator
  + Template Selector
  + Preview + Publish
* Mobile responsiveness from start

**5. Prompt Input UI (Week 2)**

* Fields:
  + Full Name, Profession, Short Bio, Skills
  + Location, Email, Phone, LinkedIn, GitHub, Other Socials
  + Education, Work Experience, Certifications, Personal Motto
* Validation for required fields

**6. Template Gallery UI (Week 3)**

* Display 10 static template thumbnails
* Allow selection via radio/active class
* Preview selected template before publishing

**7. Template Component Setup (Week 3)**

* Build 10 templates in /templates
* Use Tailwind CSS + JSX slots for dynamic injection
* Add fake content for now to simulate structure

**8. Live Preview UI (End of Week 3)**

* Side-by-side display of selected template with prompt values filled in
* Implement loading indicators and transition effects

**Phase 2: Backend Development & AI Integration (Week 4–6)**

**Tasks:**

**1. Backend API Setup (Week 4, Day 1–3)**

* Initialize Node.js + Express project
* Setup CORS, error handling, body parser
* Create API endpoints:
  + POST /api/generate — receives prompt, calls Groq, returns JSON
  + GET /api/user/:id/portfolio — fetch user portfolio
  + POST /api/user/:id/portfolio — save portfolio to Supabase

**2. Connect to Supabase (Week 4)**

* Setup Supabase client
* Create tables:
  + users, portfolios, images
* Write helper functions to fetch/save data

**3. Groq Integration (Week 4–5)**

* Setup prompt wrapper for AI request
* Structure system prompt for JSON output:
* {
* "name": "Ali",
* "profession": "Frontend Developer",
* "bio": "...",
* "skills": [...]
* }
* Handle token count, error fallback, retries
* Connect backend endpoint to Groq

**4. Image Upload + Storage (Week 5)**

* Use Supabase buckets
* Implement /api/upload-image endpoint
* Add preview + drag-drop upload on frontend

**5. Publishing + Portfolio URL Routing (Week 5)**

* Render portfolio at domain.com/view/:username
* Pull template ID + content from DB
* Inject data and render static page

**6. Stripe Integration (Week 6)**

* Setup Stripe dashboard + API keys
* Add plans: $10/month, free trial optional
* Implement /api/subscribe
* Add premium template access checks

**Phase 3: Finalization, QA & Launch (Week 7)**

**Tasks:**

1. **Self-hosted Analytics** (Plausible/Umami)
2. **Email setup** (Mailersend/Resend)
3. **Mobile & Browser Testing**
4. **404 handling + error pages**
5. **Affiliate Ad placement for free users**
6. **Deploy Frontend (Vercel)** + Backend (Render/Railway)
7. **Close group feedback loop before public launch**

**Final Output Features (MVP Scope)**

* 10 Templates fully integrated with AI output
* Groq-powered prompt-to-portfolio conversion
* Portfolio live URL preview
* User accounts (Supabase)
* Premium tier w/ Stripe
* Image uploads & public rendering
* Analytics + email integration

**Estimated Timeline: 6–7 weeks**

* **Week 1–3**: UI, UX, and frontend logic
* **Week 4–6**: Backend + AI + integrations
* **Week 7**: QA, polish, deploy

This flow puts frontend first for early design validation, then plugs in backend logic once UI is testable — ideal for MVP velocity and early user feedback.