**Chestnut Hill Engineering Summary and Phase 2 Technology Overview**

## Overview

Chestnut Hill is a modern, full-stack social media prototype built with Laravel (PHP) and React (Vite + Tailwind CSS). The project emphasizes clean architecture, scalable UI, responsive design, and future-ready API extensibility. Phase 1 and Phase 2 covered core user experience and foundational features.

This document outlines the **engineering principles** applied and the **full technology stack** used up to the end of Phase 2.

## Engineering Principles Demonstrated

### 1. **Modularity**

* UI is built with reusable React components (Sidebar, PostItem, MobileNav, etc.)
* Backend Laravel controllers are RESTful and isolated by responsibility

### 2. **Single Responsibility Principle (SRP)**

* Each component handles a specific concern: login form, post display, mobile nav, etc.
* Pages and components do not mix unrelated responsibilities

### 3. **DRY (Don’t Repeat Yourself)**

* Shared behaviors abstracted via props and utility functions
* UI behaviors (e.g., button styles, conditional rendering) reused

### 4. **Responsive Design**

* Fully mobile-optimized via Tailwind utility classes (md:hidden, w-full, mx-auto)
* Bottom nav and sticky sidebars follow industry-standard UX patterns

### 5. **Progressive Enhancement**

* Built and launched features iteratively
* Focused on MVP flow first, then added visual polish and enhancements (like sticky behavior)

### 6. **Environment Awareness**

* API URLs and auth tokens are managed in a way that supports future environment switching
* Laravel Sanctum is used for secure token-based authentication

### 7. **User-Centered UX**

* Feed, login, profile, and post interaction flow are polished with attention to spacing, alignment, and device-specific behavior
* Buttons, forms, and feedback states are intuitive and familiar to end-users

### 8. **Separation of Concerns**

* Clear separation between backend (Laravel) and frontend (React)
* Auth logic, API data fetching, and UI rendering are not tangled together

### 9. **Clean API Design**

* RESTful Laravel API with endpoints for login, register, post creation, likes, saves, follows, hides
* Postman collection prepared for all endpoints

## Full Technology Stack

### Frontend:

* **React (Vite)** for fast, modern JS app bootstrapping
* **Tailwind CSS** for utility-first, responsive design
* **Heroicons** for consistent, accessible icons
* **React Router** for SPA routing
* **Postman** for API testing and token authentication

### Backend:

* **Laravel** (PHP) with **Sanctum** for API auth
* **MySQL** or SQLite for database during dev
* **Eloquent ORM** for querying and relationships
* **Storage disk** for image uploads (to be upgraded to S3 in Phase 3)

### Tooling:

* **Postman** for API testing
* **Git + GitHub** for version control and source management
* **Local Laravel development server** and Vite for live reload

## End-to-End Features Implemented

| Feature | Status |
| --- | --- |
| User Login / Logout | ✅ |
| Signup | ✅ |
| Token-based auth | ✅ (Sanctum) |
| Post creation (Phase 2 final task) | ✅ Text-only |
| Like / Save system | ✅ |
| Mobile bottom nav | ✅ |
| Sticky sidebars (desktop only) | ✅ |
| Responsive feed, profile, saved pages | ✅ |
| Image upload (render only, Phase 3 upgrade planned) | ✅ |
| Smart DB seeding | ⏳ Deferred to Phase 3 |
| AI/ML tagging or feed ranking | ⏳ Phase 3 |

## Prepared for Phase 3 Expansion

Phase 3 will extend Chestnut Hill with more advanced tooling, AI features, smarter seeding, and cloud deployment strategies. See “Phase 3 Roadmap” document for more.