CLST Claude Project - Prompting Strategy & Implementation Plan

Part 1: Claude Project Prompting Strategy

Project Setup Instructions for Claude

```
# CLST Digital Closet Application - Project Context

## Project Overview
You are helping build CLST, a modern fashion/closet management platform with:
- Digital closet management
- Fashion marketplace
- Style challenges
- Trend tracking
- Social features

## Technical Stack
- React 18+ with TypeScript
- Vite for build tooling
- Zustand for state management
- React Query for server state
- React Router v6 for routing
- Tailwind CSS for styling
- Vitest for testing
- i18n for internationalization

## Architecture Requirements
- Feature-based folder structure
- Zero technical debt approach
- Mobile-first responsive design
- Offline-first with advanced caching
- Virtual scrolling for large lists
- Comprehensive analytics
- WCAG AAA accessibility
- Sub-second load times

## Code Standards
- 100% TypeScript coverage
- Functional components with hooks
- Custom hooks for business logic
- Comprehensive error boundaries
- Optimistic UI updates
- Progressive enhancement
```

Systematic Prompting Strategy

Prompt 1: Initial Architecture Setup

```
I need help building a modern React TypeScript application called CLST - a digital closet and fashion
community platform.

Key features:
1. Digital closet management (add/edit/delete clothing items)
2. Outfit creation and planning
3. Fashion marketplace (buy/sell)
4. Style challenges
5. Trend tracking

Technical requirements:
- Feature-based architecture
- TypeScript with strict mode
```

```
- Zustand for state management
- Virtual scrolling for large lists
- Offline-first with caching
- i18n support
- Advanced analytics

Please provide:
1. Complete project structure
2. Core type definitions
3. Base component architecture
4. State management setup

Focus on scalability and zero technical debt.
```

Prompt 2: Core Feature Implementation

```
Now let's implement the core features for CLST. I need:

1. Authentication module with:
   - Email/password login
   - Email verification flow
   - Password reset
   - Secure token management

2. Digital closet module with:
   - CRUD operations for clothing items
   - Categories: tops, bottoms, shoes, accessories, outerwear
   - Image management
   - Tagging system
   - Search and filtering
   - Virtual scrolling for large collections

3. Outfit creation:
   - Combine items into outfits
   - Save outfit combinations
   - Weekly planning feature
   - Weather integration

Please provide complete implementations with TypeScript types, Zustand stores, and React components.
```

Prompt 3: Advanced Features

```
Let's add advanced features to CLST:

1. Marketplace functionality:
   - List items for sale
   - Browse and search listings
   - Secure transactions
   - Seller ratings
   - Favorites system

2. Style challenges:
   - 30-day challenges
   - Progress tracking
   - Community participation
   - Rewards system

3. Performance optimizations:
   - Implement virtual scrolling with react-window
   - Advanced caching with React Query and IndexedDB
   - Progressive image loading
   - Code splitting by route
   - Bundle size optimization

Include analytics tracking and error handling.
```

Prompt 4: Internationalization & Analytics

```
Add comprehensive i18n and analytics to CLST:

1. Internationalization:
   - Support for EN, ES, FR, JA
   - Dynamic language switching
   - Localized formatting (dates, currency)
   - Translation management system
   - RTL support preparation

2. Analytics implementation:
   - Multi-provider setup (Mixpanel, PostHog, Sentry)
   - Event tracking system
   - User behavior analytics
   - Performance monitoring
   - Error tracking with context
   - Revenue tracking

3. Testing strategy:
   - Unit tests for components
   - Integration tests for features
   - E2E test setup
   - Accessibility testing

Provide complete implementations with examples.
```

Prompt 5: Production Readiness

```
Make CLST production-ready:

1. Security implementations:
   - Input sanitization
   - CSRF protection
   - Secure authentication
   - Data encryption
   - Rate limiting

2. CI/CD pipeline:
   - GitHub Actions workflow
   - Automated testing
   - Code quality checks
   - Performance budgets
   - Deployment scripts

3. Monitoring & Observability:
   - Error tracking setup
   - Performance monitoring
   - User analytics dashboard
   - Health checks
   - Alerting system

4. Documentation:
   - API documentation
   - Component library docs
   - Deployment guide
   - Contributing guidelines
```

Advanced Development Prompts

Prompt 6: AI Features

```
Add AI-powered features to CLST:

1. Outfit recommendations:
   - ML-based style matching
```

```
   - Personalized suggestions
   - Trend prediction
   - Color coordination

2. Image recognition:
   - Auto-categorize clothing
   - Extract colors/patterns
   - Brand detection
   - Similar item search

3. Virtual try-on:
   - AR implementation
   - Size recommendations
   - Fit prediction

Integrate with existing architecture.
```

Prompt 7: Trophy Case & Achievements

```
Create a comprehensive trophy case system for CLST challenges:

1. Trophy Case Page:
   - Accessible from Challenges page
   - Visual trophy/badge display
   - Achievement categories
   - Progress indicators
   - Sharing capabilities

2. Achievement Types:
   - Challenge completion badges
   - Streak achievements
   - Milestone rewards
   - Seasonal trophies
   - Community recognition

3. Trophy Details:
   - Achievement name and description
   - Date earned
   - Rarity level
   - Associated challenge
   - Points/rewards earned

4. Features:
   - Animated trophy reveals
   - Progress tracking
   - Leaderboard integration
   - Social sharing
   - Export/showcase options

Please provide complete TypeScript implementation with:
- Trophy types and interfaces
- Zustand store for achievements
- Trophy case components
- Animation system
- Responsive grid layout
```

Prompt 8: Social Features

```
Implement social features for CLST:

1. User profiles:
   - Follow/follower system
   - Activity feed
   - Style inspiration boards
   - Outfit sharing

2. Community features:
   - Comments and likes
```

```
   - Direct messaging
   - Group challenges
   - Style forums

3. Influencer tools:
   - Analytics dashboard
   - Sponsored content
   - Affiliate links
   - Brand partnerships
```

Project Configuration Prompts

Environment Setup

```
Set up the development environment for CLST with:
- VS Code settings
- ESLint configuration
- Prettier setup
- Git hooks with Husky
- Environment variables structure
- Docker configuration
- Development scripts
```

Performance Optimization

```
Optimize CLST for maximum performance:
- Lighthouse score > 95
- Sub-second load times
- 60fps scrolling
- Minimal bundle size
- Efficient re-renders
- Memory leak prevention
```