

# Vibe Coding Prompt Library

---

Tech Horizon Academy

*Build Functional Apps with AI - No Coding Experience Required*

---

## Table of Contents

---

1. [App Planning & Architecture](#)
  2. [Frontend Development](#)
  3. [Backend & Database](#)
  4. [Debugging & Troubleshooting](#)
  5. [Code Optimization](#)
  6. [API Integration](#)
  7. [Testing & Quality Assurance](#)
  8. [Deployment & Maintenance](#)
- 

## App Planning & Architecture

---

### 1. App Idea Validator

**Tier:** Starter+

**Platform:** Manus, Cursor, ChatGPT

**Use Case:** Validate and refine your app concept

Act as a product strategist and technical architect. I want to build [APP IDEA]. Help me:

1. Validate if this idea solves a real problem
2. Identify the core features (MVP)
3. Suggest the simplest tech stack for a no-code/low-code approach
4. Estimate complexity (beginner/intermediate/advanced)
5. List potential challenges
6. Recommend similar apps to study
7. Suggest the best platform to build it (Manus/Replit/Bubble/etc.)

App Idea: [YOUR INPUT]

Target Users: [YOUR INPUT]

Main Problem It Solves: [YOUR INPUT]

## 2. Feature Specification Generator

**Tier:** Lite+

**Platform:** Manus, Cursor

**Use Case:** Create detailed feature requirements

You are a product manager. Create a detailed feature specification for [FEATURE] in my [APP TYPE] app:

- User story: As a [user type], I want to [action] so that [benefit]
- Acceptance criteria (what defines "done")
- User flow (step-by-step interaction)
- Data requirements (what needs to be stored)
- UI/UX considerations
- Edge cases to handle
- Success metrics

Feature: [YOUR INPUT]

App Type: [YOUR INPUT]

User Type: [YOUR INPUT]

## 3. Database Schema Designer

**Tier:** Lite+

**Platform:** Manus, Replit

**Use Case:** Design database structure for your app

Act as a database architect. Design a database schema for [APP TYPE] that needs to handle:

[LIST YOUR DATA REQUIREMENTS]

Provide:

- Table names and relationships
- Column names with data types
- Primary and foreign keys
- Indexes for performance
- Sample data structure in JSON format
- Explanation of relationships
- Best practices for this use case

App Type: [YOUR INPUT]

Data Requirements: [YOUR INPUT]

Expected Scale: [Small/Medium/Large]

## Frontend Development

---

### 4. Landing Page Builder

**Tier:** Starter+

**Platform:** Manus, Cursor

**Use Case:** Generate complete landing page code

You are a frontend developer. Create a modern, responsive landing page for [PRODUCT/SERVICE] using React and Tailwind CSS. Include:

- Hero section with headline, subheadline, and CTA button
- Features section (3-4 key benefits with icons)
- How it works (3-step process)
- Social proof section (testimonials or stats)
- Pricing cards (if applicable)
- FAQ section
- Footer with links

Requirements:

- Mobile-first responsive design
- Clean, modern aesthetic
- Fast loading (optimized images)
- Accessibility best practices

Product/Service: [YOUR INPUT]

Brand Colors: [YOUR INPUT]

Target Audience: [YOUR INPUT]

## 5. Form Component Generator

**Tier:** Starter+

**Platform:** Manus, Cursor, Replit

**Use Case:** Create functional forms with validation

Act as a React developer. Create a form component for [FORM PURPOSE] with these fields:

[LIST FIELDS: name, email, etc.]

Include:

- Input validation (email format, required fields, etc.)
- Error messages
- Loading state during submission
- Success/failure feedback
- Accessible labels and ARIA attributes
- Mobile-friendly styling with Tailwind CSS
- Submit handler that sends data to API endpoint

Form Purpose: [YOUR INPUT]

Fields Needed: [YOUR INPUT]

Validation Rules: [YOUR INPUT]

## 6. Dashboard Layout Creator

**Tier:** Lite+

**Platform:** Manus, Cursor

**Use Case:** Build admin or user dashboard interfaces

You are a UI/UX developer. Create a dashboard layout for [APP TYPE] with:

- Sidebar navigation (collapsible on mobile)
- Top header with user profile and notifications
- Main content area with cards/widgets
- Data visualization components (charts, stats)
- Responsive grid layout
- Dark mode toggle
- Loading skeletons

Use React, Tailwind CSS, and shadcn/ui components.

App Type: [YOUR INPUT]

Key Metrics to Display: [YOUR INPUT]

User Actions Needed: [YOUR INPUT]

## 7. Interactive Component Builder

**Tier:** Lite+

**Platform:** Manus, Cursor

**Use Case:** Create reusable UI components

Act as a React component library developer. Create a reusable [COMPONENT TYPE] component with:

- Props interface (TypeScript)
- State management (if needed)
- Event handlers
- Variants (sizes, colors, states)
- Accessibility features
- Usage examples
- Storybook-style documentation

Component Type: [Modal/Dropdown/Carousel/Tabs/etc.]

Required Features: [YOUR INPUT]

Styling Framework: [Tailwind/CSS/styled-components]

## Backend & Database

### 8. API Endpoint Generator

**Tier:** Lite+

**Platform:** Manus, Replit

**Use Case:** Create RESTful API endpoints

You are a backend developer. Create a tRPC/Express API endpoint for [ACTION] that:

- Accepts [INPUT PARAMETERS]
- Validates input data
- Performs [DATABASE OPERATION]
- Returns [RESPONSE FORMAT]
- Handles errors gracefully
- Includes authentication check (if needed)
- Follows REST best practices

Action: [Create/Read/Update/Delete]

Resource: [YOUR INPUT]

Input Parameters: [YOUR INPUT]

Database: [MySQL/PostgreSQL/MongoDB]

## 9. Database Query Builder

**Tier:** Lite+

**Platform:** Manus, Replit

**Use Case:** Generate optimized database queries

Act **as** a database expert. Write a [DATABASE TYPE] query **to**:  
[DESCRIBE WHAT YOU NEED TO DO]

**Include:**

- The complete query **with** proper syntax
- Explanation **of each** part
- Performance considerations
- Index recommendations
- Example result **set**
- Alternative approaches

Database Type: [MySQL/PostgreSQL/MongoDB]

Tables Involved: [YOUR INPUT]

Desired Output: [YOUR INPUT]

## 10. Authentication System

**Tier:** Pro

**Platform:** Manus, Replit

**Use Case:** Implement user authentication

You are a security-focused backend developer. Implement a complete authentication system for [APP TYPE] with:

- User registration with email verification
- Login with JWT tokens
- Password hashing (bcrypt)
- Protected routes/endpoints
- Session management
- Password reset flow
- OAuth integration (Google/Facebook) - optional
- Rate limiting for security

Provide:

- Backend code (Node.js/Express or tRPC)
- Database schema for users
- Frontend login/signup forms
- Security best practices

App Type: [YOUR INPUT]

Auth Provider: [JWT/OAuth/Both]

## Debugging & Troubleshooting

### 11. Error Debugger

**Tier:** Starter+

**Platform:** Manus, Cursor, ChatGPT

**Use Case:** Fix bugs and errors in your code

Act **as** a debugging expert. I'm getting this error in my [LANGUAGE/FRAMEWORK] code:

[PASTE ERROR MESSAGE]

Here's the relevant code:

[PASTE CODE SNIPPET]

Help **me**:

1. Explain what's causing the error
2. Provide the fix **with** corrected code
3. Explain why the fix works
4. Suggest how **to** prevent similar errors
5. Recommend debugging tools **or** techniques

Language/Framework: [YOUR INPUT]

Error Message: [YOUR INPUT]

Code Context: [YOUR INPUT]

## 12. Code Review Assistant

**Tier:** Lite+

**Platform:** Cursor, ChatGPT

**Use Case:** Get feedback on code quality

You are a senior developer conducting a code review. Review this [LANGUAGE] code for:

- Bugs and logic errors
- Performance issues
- Security vulnerabilities
- Code readability and maintainability
- Best practices violations
- Potential edge cases
- Refactoring opportunities

[PASTE CODE]

Provide:

- Issues found (with severity: critical/major/minor)
- Suggested improvements with code examples
- Explanation of why changes are needed

Language: [YOUR INPUT]

Code Purpose: [YOUR INPUT]

## 13. Performance Optimizer

**Tier:** Lite+

**Platform:** Manus, Cursor

**Use Case:** Improve app speed and efficiency

Act as a performance optimization expert. My [APP TYPE] is slow. Analyze and optimize:

- Database queries (N+1 problems, missing indexes)
- Frontend rendering (unnecessary re-renders, large bundles)
- API calls (caching, batch requests)
- Image/asset loading
- Memory leaks

Current issues:

[DESCRIBE PERFORMANCE PROBLEMS]

Provide:

- Root cause analysis
- Specific optimizations with code examples
- Expected performance improvements
- Monitoring recommendations

App Type: [YOUR INPUT]

Performance Issues: [YOUR INPUT]

Tech Stack: [YOUR INPUT]

## Code Optimization

---

### 14. Refactoring Assistant

**Tier:** Lite+

**Platform:** Cursor, ChatGPT

**Use Case:** Clean up and improve existing code

You are a code refactoring specialist. Refactor this [LANGUAGE] code to:

- Improve readability
- Reduce complexity
- Follow DRY principles
- Use modern syntax/features
- Add proper error handling
- Include helpful comments

[PASTE CODE TO REFACTOR]

Provide:

- Refactored code with explanations
- Before/after comparison
- Benefits of the refactoring

Language: [YOUR INPUT]

Code Purpose: [YOUR INPUT]

Framework: [YOUR INPUT]

## 15. Type Safety Converter

**Tier:** Lite+

**Platform:** Cursor

**Use Case:** Add TypeScript types to JavaScript code

Act as a TypeScript expert. Convert this JavaScript code to TypeScript with:

- Proper type annotations
- Interface definitions
- Generic types where appropriate
- Strict mode compliance
- Type guards for runtime safety

[PASTE JAVASCRIPT CODE]

Provide:

- Fully typed TypeScript version
- Explanation of type choices
- Benefits of type safety for this code

Original Language: JavaScript

Target: TypeScript

## API Integration

---

### 16. Third-Party API Integrator

**Tier:** Lite+

**Platform:** Manus, Replit

**Use Case:** Connect to external APIs (Stripe, OpenAI, etc.)

You are an API integration specialist. Help me integrate [API NAME] into my [APP TYPE]:

- Authentication setup (API keys, OAuth)
- Request/response handling
- Error handling and retries
- Rate limiting considerations
- Webhook setup (if applicable)
- Testing strategy

Provide:

- Complete integration code
- Environment variable setup
- Example API calls
- Error scenarios to handle

API: [Stripe/OpenAI/Google Maps/etc.]

Use Case: [YOUR INPUT]

Language: [Node.js/Python/etc.]

## 17. Webhook Handler

**Tier:** Pro

**Platform:** Manus, Replit

**Use Case:** Process incoming webhook events

Act as a backend developer. Create a webhook handler for [SERVICE] that:

- Receives webhook POST requests
- Verifies webhook signatures
- Parses event data
- Performs actions based on event type
- Logs events for debugging
- Handles failures gracefully
- Returns proper status codes

Service: [Stripe/GitHub/Shopify/etc.]

Events to Handle: [YOUR INPUT]

Actions to Take: [YOUR INPUT]

# Testing & Quality Assurance

---

## 18. Unit Test Generator

**Tier:** Pro

**Platform:** Manus, Cursor

**Use Case:** Create comprehensive test suites

You are a test-driven development expert. Write unit tests for this [LANGUAGE/FRAMEWORK] function:

[PASTE FUNCTION CODE]

Create tests for:

- Happy path scenarios
- Edge cases
- Error conditions
- Boundary values
- Mock external dependencies

Use [TESTING FRAMEWORK] and provide:

- Complete test file
- Test descriptions
- Setup/teardown if needed
- Coverage report interpretation

Language/Framework: [YOUR INPUT]

Testing Framework: [Jest/Vitest/Pytest/etc.]

Function Purpose: [YOUR INPUT]

## 19. Integration Test Builder

**Tier:** Pro

**Platform:** Manus, Replit

**Use Case:** Test API endpoints and workflows

Act as a QA engineer. Create integration tests for [FEATURE/ENDPOINT]:

- Test complete user flows
- Test API endpoints with various inputs
- Test database interactions
- Test authentication/authorization
- Test error responses

Provide:

- Test scenarios and cases
- Test code with assertions
- Mock data setup
- Expected outcomes

Feature/Endpoint: [YOUR INPUT]

Tech Stack: [YOUR INPUT]

## Deployment & Maintenance

---

### 20. Deployment Checklist Generator

**Tier:** Pro

**Platform:** Manus, Replit

**Use Case:** Prepare app for production launch

You are a DevOps engineer. Create a deployment checklist for [APP TYPE] covering:

- Environment variables configuration
- Database migration strategy
- SSL/HTTPS setup
- Domain configuration
- Performance optimization
- Security hardening
- Monitoring and logging setup
- Backup strategy
- Rollback plan
- Post-deployment testing

App Type: [YOUR INPUT]

Hosting Platform: [Vercel/Railway/AWS/etc.]

Database: [YOUR INPUT]

## Platform-Specific Tips

---

### Manus (Recommended for Beginners)

- **Best for:** Full-stack web apps with database
- **Strengths:** Built-in auth, database, deployment
- **Use when:** You want the fastest path from idea to live app
- **Prompts to use:** 1, 4, 5, 6, 8, 9, 10, 16

### Replit (Best for Learning)

- **Best for:** Prototyping, learning, mobile apps
- **Strengths:** Collaborative coding, mobile app builder
- **Use when:** You want to experiment or learn by doing
- **Prompts to use:** All prompts work, great for 11-15

## Cursor (Best for Advanced Users)

- **Best for:** Complex apps, custom code
  - **Strengths:** AI pair programming, code completion
  - **Use when:** You need more control and customization
  - **Prompts to use:** 7, 11, 12, 13, 14, 15, 18
- 

# How to Use This Library

---

## For Complete Beginners

1. Start with **App Idea Validator** (#1)
2. Use **Database Schema Designer** (#3) to plan your data
3. Build UI with **Landing Page Builder** (#4) or **Dashboard Layout** (#6)
4. Add functionality with **API Endpoint Generator** (#8)
5. Fix issues with **Error Debugger** (#11)

## For Intermediate Builders

1. Use **Feature Specification Generator** (#2) for planning
2. Create reusable components with **Interactive Component Builder** (#7)
3. Integrate services with **Third-Party API Integrator** (#16)
4. Optimize with **Performance Optimizer** (#13)
5. Test with **Unit Test Generator** (#18)

## For Advanced Developers

1. Design complex systems with **Database Schema Designer** (#3)
2. Build secure auth with **Authentication System** (#10)
3. Handle webhooks with **Webhook Handler** (#17)
4. Ensure quality with **Integration Test Builder** (#19)

## 5. Deploy confidently with **Deployment Checklist** (#20)

---

# Pro Tips

---

## Prompt Engineering for Code

- **Be specific** about your tech stack (React, Node.js, MySQL, etc.)
- **Provide context** (what you've already built, what's not working)
- **Ask for explanations** not just code (so you learn)
- **Request alternatives** (there's often more than one way)
- **Iterate** - refine the output with follow-up prompts

## Common Follow-Up Prompts

- “Explain this code line by line”
- “Make this more beginner-friendly”
- “Add error handling for [scenario]”
- “Convert this to use [different library]”
- “Show me how to test this”
- “What could go wrong with this approach?”

## Tier Access

- **Starter (\$27)**: Access to 8 basic prompts
  - **Lite (\$77/mo)**: Access to 15 prompts + monthly workshops
  - **Pro (\$247/mo)**: Full library + advanced workshops + code reviews
  - **Elite (\$797/mo)**: Everything + 1-on-1 app building sessions
-

# Need More Help?

---

## Join Vibe Coding Workshops

Build real apps with AI in live, hands-on sessions.

2-4 workshops per month (Lite: 2, Pro: 4).

## Access the Full Tools Database

1,620+ AI tools including no-code platforms, code editors, and deployment services.

## Get Your Code Reviewed

Pro and Elite members get personalized code feedback and optimization tips.

---

© 2025 Tech Horizon Labs. All rights reserved.

For support: [info@thzn.world](mailto:info@thzn.world)