Vibe Coding: Coding in the Age of AI

Unlocking Creativity and Speed with AI-Assisted Development

A presentation by Solomon Tsao

Solomon Tsao

9+ years experience in Software Development

10 years in circuit design & testing in audio field

Founder of ResumeGuru.IO , Yottatrend.com , orionconnect.io

Master in Software Engineering from UTD

PADI SCUBA instructor

Today's Vibe

What is Vibe Coding?

A revolutionary approach to software development that leverages AI

Why it Matters (for everyone!)

The benefits and implications for developers and non-developers alike

Tools of the Trade

Overview of Lovable, Bolt, Cursor, Windsurf, GitHub Copilot, Gemini Code Assist, Claude CLI, Gemini CLI

LIVE Demo: See it in Action!

Watch Vibe Coding transform ideas into working software

Workshop Time

Try on your own to build an app or modify from your existing code

Q&A

Your questions answered

Coding: Then vs. Now (The "Old Way")

- **Manual, line-by-line**
- **Requires deep technical knowledge**
- **Can be slow & prone to human error**

While we acknowledge the incredible work of traditional developers, it's important to understand how things are changing in the AI era.

So, What Exactly IS Vibe Coding?

- "Talking" to AI to build software.
- You describe what you want (in plain English!).
- AI generates the code for you.
- It's like having a super-smart assistant!

Why Vibe Coding?

For Everyone: Unleashing Your Inner Creator

Lower Barrier to Entry: Anyone can build!
Rapid Prototyping: Turn ideas into working apps quickly.
Automate Repetitive Tasks: Free up time for creativity.
Learn Faster: See how code works in real-time.

Why Vibe Coding?

For Developers: Supercharging Your Workflow

Boost Productivity: Generate boilerplate, complex logic.
Reduce Debugging Time: AI helps identify and fix errors.
Explore New Technologies: Experiment without deep dives.
Focus on Complex Problems: Delegate the mundane.

Lovable & Bolt & base44 & replit (Focus on "full app from prompt")

Key Feature: Great for non-technical users to get started.

Tools	Primary Focus	Output Quality (Backend)	Known Limitations
Lovable.dev	Rapid "Minimum Lovable Product" (MVP) UI generation, beginner-friendly.	Limited backend functionality, better for mockups/prototypes.	Can be slower for updates; AI-generated code may need optimization for scale.
Bolt.new	Full-stack applications, more control, backend focus.	More robust backend functionality, user authentication, API integration.	While fast, complex custom features might still require significant manual intervention.
base44.com	All-in-one platform for no-code app building with built-in backend features.	Built-in database management, authentication, email system, analytics, storage – all native.	Less flexibility for highly custom UIs or integrations outside its built-in stack.
replit.com	Cloud-based IDE for coding, collaboration, and deployment, with strong AI assistance. https://replit.com/news/microsoft-partnership	Supports various backend languages/frameworks (Node.js, Python/Flask, etc.), offers built-in database and auth.	Can be less optimized for full app generation from scratch compared to dedicated gen tools; occasional "error loops."

Cursor vs. Windsurf vs. GitHub Copilot vs. Gemini Code Assist

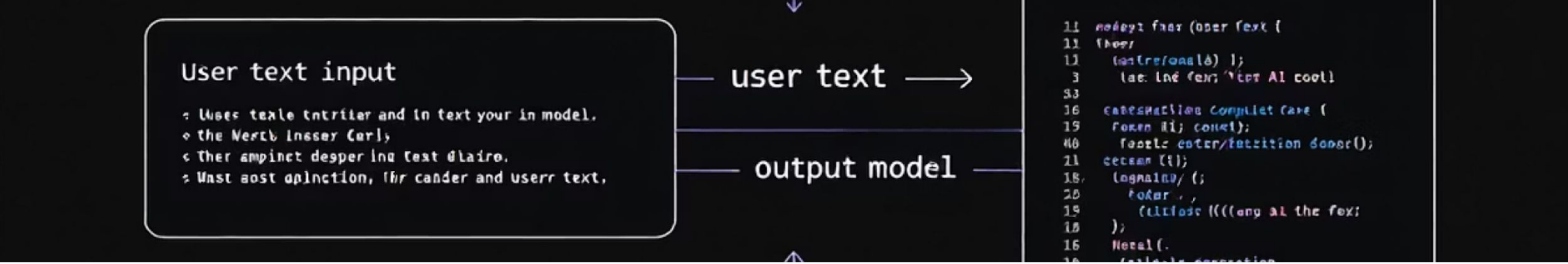
Key Feature: Enhance existing developer workflows.

Tools	Primary Focus	Context Awareness	Integration/Ecosystem
Cursor	Granular AI control, codebase understanding, autonomous "Agent Mode," multi-line edits.	Excellent; uses custom retrieval models, @ references for files/web/docs, image input.	Standalone editor (VS Code fork), but can integrate with GitHub.
Windsurf (Codeium)	Context-aware suggestions, "Supercomplete," "Cascade" for multi-file edits, real-time sync.	"Flow" technology for real-time workspace sync, Riptide for large codebase scanning.	Standalone editor, JetBrains IDEs. Strong context awareness with Software Configuration Management.
GitHub Copilot	Real-time inline code completion, boilerplate generation, GitHub ecosystem integration.	Good; understands current file context, some project-wide context via chat/Edits.	Deeply integrated with GitHub and Microsoft ecosystem; available across popular IDEs.
Gemini Code Assist	Context-aware code generation, multi-file editing, integrated with Google Cloud/Firebase.	Very strong; 1M token context window, deep understanding of local codebase, supports private repos (Enterprise).	Integrated with Google Cloud, Firebase, and Google Workspace; available as IDE plugins.

Claude CLI & Gemini CLI (Focus on "conversational coding")

Key Feature: Text-based interaction, good for quick questions, generating small scripts, understanding errors.

Tools	Primary Focus	Code Quality	Key Use Cases
Claude CLI	Agentic coding tool, focused on precise code generation, ethical AI, and autonomous task execution.	Often produces higher quality, more structured, and production-ready code. Excels in complex reasoning.	Building mission-critical commercial applications, high-quality code generation, automated refactoring, ethical AI-driven development.
Gemini CLI	Utility-focused, rapid prototyping, large context understanding, multimodal capabilities, deep Google ecosystem integration.	Good for quick fixes and utility scripts. Can sometimes be less structured or overly minimal for complex projects.	Rapid prototyping, understanding large codebases, quick fixes, generating code from visual inputs (images/PDFs), automating general system tasks.



The "Magic" Behind the Vibe

User Input

Natural language description of what you want to build with your **Domain Knowledge**

AI Processing

Large Language Models (LLMs) understand your intent

Code Generation

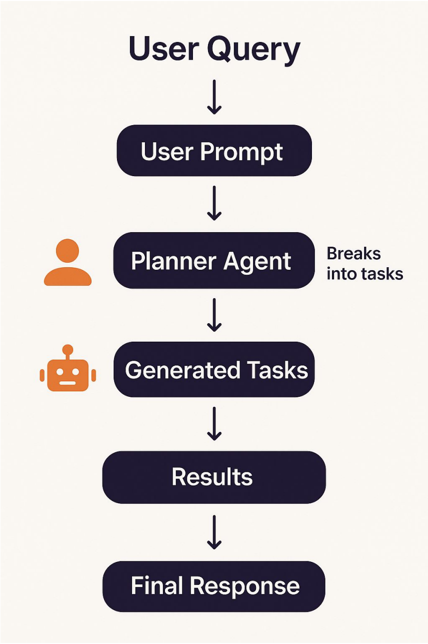
AI produces code snippets, functions, or full applications

Iteration

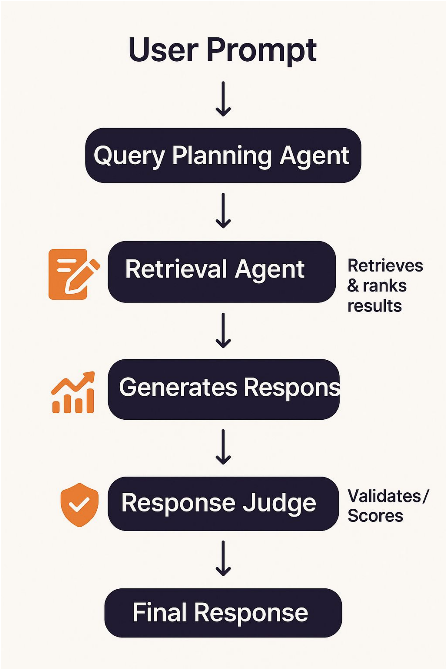
Describe, generate, refine, fix errors in a conversational flow

Agentic AI Patterns

Planning Pattern



Multi-Agent Collaboration



Start with a structured prompt :

Project Overview

Technology Stack

Architecture Design

Core Components

AI Agent Design

Agent Personality & Strategy

Prompt Engineering Strategy
```
System: Your system prompt
```

User Interface Design

Data Flow Architecture

AI Integration Flow

Demo : for non developer

"Describe your app idea with your **Domain Knowledge**" + “design prompt template”
= Your tailored design prompt for vibe coding

Design_prompt_template => https://yottatrend.com/prompt-template
Design_prompt_example => https://yottatrend.com/prompt-example

Demo : for developer, cli coding

Lab VM Information

lab1.solomontsao.com ~ lab30.solomontsao.com
Each VM is for two users and please try your project in either folder A or B based on the little note you got.

You need a Google Gmail account. Get your free api key from Google AI Studio.

You can use either the free VM (4CPU/16G Ram/40GB disk) to try or use github codespace. (120 core hours / month)

Login to the VM :
ssh root@lab1.solomontsao.com / Ubuntu@0715
Change to your folder, either cd a / cd b
Type “Gemini” and enter to start

Q&A

Thank you.

LinkedIn : https://www.linkedin.com/in/solomon-tsao/