

VIBE CODER

AI-Powered Local Coding Environment – Complete A to Z Documentation

1. Vision & Purpose

Vibe Coder is an AI-powered local-first coding environment inspired by Cursor and VS Code. It is designed to deeply understand entire codebases, analyze files and folders, detect errors, rewrite and refactor code safely, and maintain persistent AI memory inside the project itself.

2. Problems Solved

Traditional IDEs lack deep AI reasoning. Cloud AI lacks privacy and full project context. Vibe Coder solves this by running local LLMs, scanning full repositories, storing error reports, and applying controlled AI-driven edits with diff previews and rollback support.

3. Target Users

- AI Engineers
- Full Stack Developers
- Startup Founders
- Power Users
- Offline / Privacy-focused developers

4. Core Features

- Full repository scanning
- Context-aware AI chat
- Multi-file reasoning
- Safe code rewriting
- Error & bug tracking inside project
- Integrated terminal (CMD, PowerShell)
- Docker & GPU support

5. Technology Stack

Backend: Python 3.11, FastAPI, Pydantic, Uvicorn

Frontend: React + TypeScript, Vite, TailwindCSS, Monaco Editor

AI: CodeBERT, DeepSeek-Coder, LLaMA 3, FAISS, LangChain

Infra: Docker, NVIDIA Container Toolkit, WSL2, CUDA

6. System Architecture

UI Layer communicates with FastAPI via REST and WebSockets. AI Orchestrator manages prompts, tools, and LLMs. Repository Analyzer scans and indexes files. Error Engine captures runtime and static issues.

7. Repository Structure

backend/: API, AI engine, repo scanner, error handler

frontend/: UI, editor, state, services

workspace/: user projects & AI reports

docker/: containers

docker-compose.yml: full stack orchestration

8. AI Workflow

User opens project → Repo scanned → Embeddings generated → Stored in vector DB → User queries AI → Context builder selects relevant files → LLM responds → Diff preview shown → User accepts → Changes applied safely.

9. Error & Bug Intelligence

Errors are detected via terminal output, static analysis, and runtime logs. Each error is classified, explained, and stored inside /ai-reports/errors as JSON.

10. Terminal Integration

Supports npm, pip, pytest, docker, git commands. Outputs are parsed and sent to AI for explanation and fixes.

11. LLM Deployment

Local LLMs run via Docker or native CUDA. RTX 4060 supports 7B–8B models efficiently. Quantized models ensure stability and speed.

12. Security

Sandboxed execution, path protection, git snapshots before write operations, manual approval for destructive actions.

13. Development Phases

Phase 1: Editor + Repo Scan + Chat

Phase 2: Write & Diff Engine

Phase 3: Error Intelligence

Phase 4: Plugins & Scaling

14. Future Scope

Plugin marketplace, team collaboration, cloud sync (optional), enterprise mode.

15. Conclusion

Vibe Coder is a production-grade AI IDE, not a demo. Designed for real-world development, privacy, and performance.

Prepared for Implementation by Muruganandam (Master ████)

This document consolidates and formalizes all discussions, ideas, structures, and workflows from the complete chat history into a single implementation-ready reference.