

Vibe Coding with Nyaya

REAL Institute / A. Chawla

January 28, 2026

Introduction

Coding in Sukanita is not merely a technical act—it is a philosophical alignment. As the world's first Nyaya-compatible programming language, Sukanita reimagines computation through the lens of Indian intellectual traditions. It is written entirely in Devanagari script and designed for microcontroller platforms like the Arduino Uno. This document explores the vibe of coding in Sukanita, using the secure email sender app as a case study.

Philosophical Foundations

Sukanita draws from six Indian traditions:

- Nyaya Logic: Guides control flow with tokens like यदि (if), अन्यथा (else), हेतु (cause), and उदाहरण (example).
- Vedic Mathematics: Shapes arithmetic semantics with tokens like गुणन (multiply), भागहार (divide), and शेष (modulo).
- Paninian Grammar: Informs lexical discipline and token structure.
- Sankhya and Yoga: Influence execution state and halting semantics, e.g., निरोध (halt).
- Vākya Tradition: Ensures surface clarity with tokens like कार्य (function), मुख्य (main), वाक्य (string).
- Embedded ML Praxis: Enables lightweight inference with tokens like मॉडल (model), पूर्वानुमान (predict).

Hardware Discipline

Suganita targets the ATmega328P microcontroller:

- 8-bit AVR core at 16 MHz
- 2 KB SRAM, 32 KB flash
- No floating point unit

These constraints enforce compact, semantically rich tokens and a stack-based VM using 16-bit logical words.

Lexical Layer and Tokens

Part I of the Suganita blueprint defines the token system for versions v0-v2. Highlights include:

- Core Keywords: कार्य (function), मुख्य (main), वापस (return)
- Control Flow: यदि, अन्यथा, क्रम (for), जबतक (while)
- Operators: तुल्य (equal), अतुल्य (not equal), अन्वय (and), व्यतिरेक (or)
- VM Opcodes: शु (NOP), प्रवेश (push), विसर्जन (pop), निरोध (halt)
- Hardware Functions: लिखो (print), स्पर्श_पढ़ो (digital read), रुको (delay)

Case Study: Secure Email Sender

The secure email sender app demonstrates Suganita's expressive power. Its source code:

- लिखो REALInstitute
- लिखो Post-QuantumEmail(Simulation)
- लिखो MessageBody
- प्रवेश Type_your_secure_message_here
- प्रवेश Type_your_secure_message_here_too_additional_space
- लिखो Status_Log

This code uses लिखो to emit UI labels and प्रवेश to define input fields. It is parsed by the SuganitaTranslator and compiled into frontend HTML and backend Flask code.

Frontend Output

The generated HTML includes:

- Title: Sughanita Generated Mail Sender
- Labels: REALInstitute, Post-QuantumEmail(Simulation), MessageBody, Status_Log
- Textareas with placeholders from प्रवेश tokens
- Button: Encrypt & Send

Backend Output

The Flask backend:

- Logs each लिखो instruction using Aryabhata consonant enumeration
- Accepts POST requests at `/send` and echoes the message

This modular design allows secure message simulation with Nyaya-aligned semantics.

Conclusion

Sughanita is more than a language—it is a philosophical and cultural reorientation of programming. By embedding Nyaya logic and Vedic clarity into the token system, it offers a new vibe for coding: one that is grounded, expressive, and spiritually resonant. The secure email sender app is just the beginning. As the language evolves through its phases—VM prototyping, compiler development, ML augmentation—it will continue to honor its roots while expanding its reach.

Acknowledgments

This work was produced with the assistance of LLMs.