

CodeMitra

System Design & Architecture

AI-Powered Multilingual Learning Platform for India

Executive Summary

What is CodeMitra?

An AI-powered multilingual learning and developer productivity platform designed specifically for the Indian market.

Mission

Democratize coding education by providing comprehensive learning resources, intelligent code assistance, and collaboration tools in 12 Indian languages.

Key Features

Real-time AI code completion • Multilingual documentation • Adaptive learning paths • Collaborative coding • Voice-to-code in regional languages

System Architecture Overview

User Layer

Web • Mobile • IDE Plugins • API

API Gateway

Load Balancer • Rate Limiting • Auth

Microservices

6 Core Services

AI/ML Layer

Code LLM • Translation • Learning

Data Layer

PostgreSQL • MongoDB • Redis • S3

Microservices Architecture

User Service

Node.js + Express

Authentication • Profiles • Subscriptions

Learning Service

Python + FastAPI

Adaptive paths • Progress tracking

Code Assistant

Python + FastAPI

AI completions • Bug detection

Translation Service

Python + FastAPI

12 languages • Tech term preservation

Collaboration Service

Node.js + Socket.io

Real-time sync • Live sessions

Analytics Service

Python + FastAPI

Usage metrics • Performance tracking

Frontend Architecture

Web Application

Tech Stack

- React 18 + TypeScript
- Next.js 14 (SSR, SSG, ISR)
- Zustand + React Query
- Tailwind CSS + Radix UI
- Monaco Editor (VS Code)
- Socket.io for real-time

Key Features

- 20+ language syntax support
- Real-time AI suggestions
- Multilingual UI & comments
- Collaborative editing

Mobile Apps

- React Native (iOS 13+, Android 8+)
- Voice-to-code in regional languages
- Offline learning modules
- Biometric authentication

IDE Plugins

- VS Code, IntelliJ, PyCharm
- Inline AI code suggestions
- Multilingual documentation
- Real-time collaboration

AI/ML Infrastructure

Code Language Model

Base: Fine-tuned CodeLlama-34B

Training: 50M+ lines of Indian code

Optimization: 4-bit quantization

Performance: <2s inference time

Translation Engine

Languages: 12 Indian languages

Model: IndicNLP transformer

Features: Tech term preservation

Quality: Context-aware translation

Adaptive Learning Algorithm

Assesses skill level → Identifies gaps → Generates personalized path → Adjusts difficulty dynamically based on performance

Database Architecture

PostgreSQL

User Data & Transactions

- ACID compliance
- Relational integrity
- Complex queries
- Multi-AZ deployment

MongoDB

Code & Content

- Flexible schema
- Hierarchical data
- Sharding enabled
- Atlas M40 cluster

Redis

Caching & Sessions

- Sub-ms latency
- Session management
- Rate limiting
- Real-time features

Elasticsearch

Search Engine

- Full-text search
- Semantic search
- Faceted filtering
- Auto-complete

Security & Compliance

Authentication

- JWT-based authentication
- Role-based access control (RBAC)
- Multi-factor authentication
- Biometric support on mobile

Encryption

- AES-256 encryption at rest
- TLS 1.3 for data in transit
- Field-level encryption for PII
- Automated key rotation

Compliance Standards

GDPR: Data minimization • Right to access • Right to deletion • Privacy by design

PCI-DSS: Secure payment processing via Razorpay/Stripe • No card data storage

Indian Data Regulations: Data localization in Mumbai region • Compliance with local laws

Cloud Infrastructure (AWS)

Primary: ap-south-1 (Mumbai) **Region Strategy**

- Main user traffic & databases
- Lowest latency for Indian users

DR: ap-southeast-1 (Singapore)

- Disaster recovery & backup

ECS Fargate

Compute Resources

- Auto-scaling 2-50 instances

Lambda Functions

- Event-driven processing
- Image processing, notifications

Content Delivery Network

CloudFront with 400+ edge locations • Static assets cached for 1 year • API response caching for 5 minutes • Automatic compression & TLS 1.3

DevOps & CI/CD Pipeline

1. Code Commit

GitHub webhook triggers

2. Build

Dependencies, linting, Docker

3. Test

Unit, integration, security

4. Staging

Deploy, smoke tests

5. Approval

Manual QA review

6. Production

Blue-green deployment

Infrastructure as Code: Terraform for AWS resources • Kubernetes manifests • Automated deployments

Monitoring & Observability

Metrics Collection

Application

- API latency (p50/p95/p99)
- Error rates by endpoint
- AI inference time

Infrastructure

- CPU & memory usage
- Network throughput
- Container health

Business

- User registrations
- Feature adoption
- Revenue metrics

Logging Strategy

CloudWatch Logs

- Structured logging
- Centralized aggregation
- Retention policies

Log Types

- Application: 30 days
- Access: 90 days
- Audit: 1 year

Distributed Tracing

- AWS X-Ray
- End-to-end visibility

Alerting

Critical

PagerDuty

- Error rate >1%
- Service down

High Priority

Slack + Email

- Latency >500ms
- Cache hit <70%

Medium

Slack

- Slow queries
- High memory

Deployment Roadmap

Phase 1: MVP

Months 1-3

- Core infrastructure setup
- AI model fine-tuning
- Code editor integration
- Hindi + English support
- Beta launch with 100 users

Phase 2: Scale

Months 4-6

- 5 additional languages
- Mobile apps launch
- IDE plugins (VS Code, IntelliJ)
- Performance optimization
- Scale to 10K users

Phase 3: Enterprise

Months 7-12

- All 12 languages
- Team collaboration features
- Advanced analytics
- Marketplace launch
- Scale to 100K users

Success Metrics & KPIs

Technical Metrics

API Response Time: <100ms (p95)

AI Inference: <2 seconds

System Uptime: 99.9%

Error Rate: <0.1%

User Metrics

Session Duration: >15 minutes

Daily Active Users: 40% of MAU

NPS Score: >70

AI Feature Adoption: 70% of users

Business Metrics

<₹500

>5%

₹50L

<5%

User Acquisition Cost

Free to Paid Conversion

MRR (Month 12)

Monthly Churn

Scalability & High Availability

Target Tracking

- CPU utilization: 70%

Auto-Scaling

- Memory utilization: 80%

- Request count: 1000/target

Predictive Scaling

- Weekday pattern: 9 AM - 11 PM

Load Balancing

AWS Application Load Balancer

- Health checks every 30s
- Sticky sessions enabled
- Path-based routing

Intelligent Routing

- Weighted traffic distribution

Disaster Recovery

Multi-Region Setup: Primary (Mumbai) + DR (Singapore) with automated failover

Recovery Objectives: RTO: 5 minutes • RPO: 1 minute (near-zero data loss)

Backup Strategy: Continuous replication • Daily snapshots (30 days) • Weekly snapshots (1 year) • Cross-region backup

Ready to Scale

CodeMitra System Architecture
✓ Scalable microservices architecture

- ✓ AI-powered multilingual platform
- ✓ Enterprise-grade security & compliance
- ✓ High availability with 99.9% uptime
- ✓ Optimized for Indian market & networks