🧱 DEVELOPMENT PLAN — AI-Powered Multilingual Localization Engine

1️⃣ SYSTEM OVERVIEW

Goal:
Create a scalable platform that:

- Translates, localizes, and contextualizes vocational skill content (text, audio, video)
- Supports **22 Indian languages**
- Integrates with LMS platforms (Skill India, NCVET)
- Provides **speech accessibility** (STT + TTS)
- Improves continuously via **user feedback loops**

2️⃣ CORE COMPONENTS

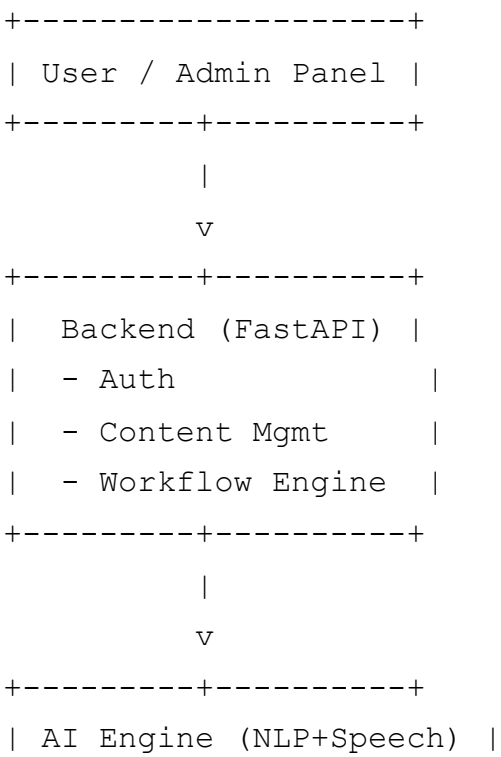
Layer	Component	Description
Frontend	Web dashboard	Content upload, translation preview, review, and feedback UI
Backend	Orchestration & APIs	Manage workflows, translation pipeline, storage, user management
AI/NLP Engine	NMT + Context Model	Multilingual neural translation and contextual adaptation
Speech Engine	STT + TTS	Converts speech-to-text (for video/audio) and text-to-speech
Data Layer	Storage + Datasets	Handles training data, metadata, and localized content
Integration Layer	API Gateway	Connects with LMS, Skill India, and external systems
Feedback System	Active Learning	Continuous model retraining based on human corrections
DevOps/Infra	Cloud Infra + MLOps	Handles deployment, CI/CD, monitoring, and scalability

---

3️⃣ TECH STACK OVERVIEW

Domain	Recommended Tech
Frontend	React.js / Next.js + TailwindCSS + TypeScript
Backend	Python (FastAPI) / Node.js (NestJS)
Database	PostgreSQL (metadata), MongoDB (unstructured content)
Storage	AWS S3 / Azure Blob (course content, media)
Auth & Security	OAuth 2.0 + JWT + HTTPS + RBAC
AI/ML Stack	PyTorch / TensorFlow + HuggingFace Transformers
Speech Models	Whisper (STT), VITS / Tacotron2 (TTS)
NLP Models	IndicBERT / mBART / NLLB / Llama 3 fine-tuned for Indic languages
MLOps	MLflow + Docker + Kubernetes + Airflow pipelines
DevOps	GitHub Actions / Jenkins, AWS EKS / Azure AKS
API Gateway	FastAPI / Kong / AWS API Gateway
Monitoring	Prometheus + Grafana
Version Control	Git + DVC (Data Version Control)

4️⃣ ARCHITECTURE FLOW



```
| - Language Detection    |
| - NMT Translation       |
| - Contextual Adaptation |
| - Cultural Localization |
| - STT / TTS Processing  |
+---------+----------+
          |
          v
+---------+----------+
| Feedback & Retraining |
+---------+----------+
          |
          v
+---------+----------+
|  Output Generator   |
|  - Text / Audio / SCORM |
|  - API Delivery         |
+-------------------------+
```

5️⃣ AI/NLP MODEL PIPELINE DETAILS

🧠 A. Language Detection

- **Model:** FastText Language Identification / LangDetect
- **Purpose:** Detect input source language (auto-detect English/Hindi/Tamil/etc.)
- **Output:** Source language code (e.g., `en`, `hi`, `ta`)

🧩 B. Neural Machine Translation (NMT)

Target	Recommended Model	Notes
General translation	**NLLB-200** (Meta AI)	200+ languages including all 22 Indian ones
Contextual/domain translation	**IndicBERT** fine-tuned	Fine-tune per domain (Healthcare, Construction, IT, etc.)
Parallel training	**mBART50 / M2M100**	For multi-language pairs
Framework	HuggingFace Transformers (PyTorch backend)	

Target	Recommended Model	Notes
Fine-tuning data	OPUS, AI4Bharat datasets, NCVET content corpus	

Training Steps:

1. Collect bilingual corpora per skill domain
2. Fine-tune NMT models (IndicBERT/mBART)
3. Validate BLEU and COMET scores
4. Optimize for inference with ONNX Runtime or TensorRT

🪶 C. Context & Cultural Localization

- Build **custom vocabulary banks** for each domain (IT, healthcare, beauty, etc.)
- Use **named entity recognition (NER)** for preserving brand/technical terms
- Add a **Cultural Adaptation Layer** that swaps examples idiomatically per region (Tamil Nadu, Assam, etc.)

🗣️ D. Speech Layer

Speech-to-Text (STT):

- Model: **Whisper Large-V3** (OpenAI)
- Use GPU inference with streaming
- Fine-tune on Indian accent data (AI4Bharat Indic-ASR dataset)

Text-to-Speech (TTS):

- Model: **VITS / Tacotron2 + HiFi-GAN**
- Use multilingual voicebanks (IndicTTS, IITM)
- Export audio in WAV/MP3 with neural prosody control

♻️ E. Feedback Loop (Active Learning)

- Capture corrections made by human validators in the web dashboard
- Use a **feedback database** (PostgreSQL)
- Automate retraining pipeline using **MLflow + Airflow**
- Periodically retrain models on corrected translations

6️⃣ DEPLOYMENT BLUEPRINT

Area	Tool	Purpose
Containerization	Docker	Package model & API
Orchestration	Kubernetes (AWS EKS)	Scaling, load balancing
CI/CD	GitHub Actions	Automate build, test, deploy
Monitoring	Prometheus + Grafana	Resource and model performance tracking
Logging	ELK Stack (Elasticsearch, Logstash, Kibana)	Centralized logs
Versioning	DVC + Git	Track model & dataset versions

7️⃣ DEVELOPMENT MILESTONES (Technical)

Phase	Duration	Technical Deliverables
Phase 1: Setup & Data	Month 1–2	Dev environment, dataset collection, model selection
Phase 2: Prototype NMT Engine	Month 3–4	Build base translation pipeline (5 languages)
Phase 3: Domain Fine-tuning	Month 5–6	Add contextual vocabularies
Phase 4: Speech Integration	Month 7–8	Implement STT/TTS layers
Phase 5: Cultural Adaptation	Month 9–10	Develop cultural localization module
Phase 6: API & LMS Integration	Month 11	Build REST APIs, connect to LMS
Phase 7: QA + Active Learning Loop	Month 12	Feedback system + retraining pipelines

---

8️⃣ RESOURCE REQUIREMENTS

Resource	Spec	Notes
Developers	6–8 total	AI, Backend, Frontend, Speech, DevOps
Compute	4× A100 GPUs (training) + 2× T4 (inference)	Hosted on AWS or Azure

Resource	Spec	Notes
Storage	10 TB	Corpus + model checkpoints + audio/video
Budget Estimate	~ ₹75–90 lakh (12 months)	Includes compute, team, infra, dataset licensing
Cloud Provider	AWS (preferred)	EC2 + S3 + EKS + SageMaker

9️⃣ FINAL OUTPUTS

✅ REST API endpoints for translation, localization, STT/TTS
✅ Web UI for content upload, review, and feedback
✅ Scalable backend with role-based access
✅ Automated retraining & deployment pipelines
✅ LMS integration-ready APIs
✅ Fully accessible (WCAG 2.1 compliant) platform