ফাইনাল ব্লুপ্রিন্ট · V2

বাংলা ইশারা ভাষা → টেক্সট ও কণ্ঠ

অক্ষর · শব্দ · বাক্য — সব চিনে রিয়েল-টাইমে বলবে। On-device recognition + Gemini ভাষা স্তর, এক ওয়েব ইঞ্জিন PC/iOS/Android সবখানে। ২ জনের টিমের জন্য সাজানো।

On-device	~৫০ms	২ জন	~$0	৪
রিয়েল-টাইম চেনা	লেটেন্সি	ডেভেলপার	ডেমো খরচ	প্ল্যাটফর্ম

BdSL · MediaPipe + TensorFlow.js + Gemini + TTS | স্কোপ: অক্ষর + শব্দ + বাক্য | জুন ২০২৬

এটা তোমার পছন্দ অনুযায়ী সাজানো ফাইনাল প্ল্যান — **full স্কোপ** (অক্ষর, শব্দ ও বাক্য), **সবচেয়ে fast রিয়েল-টাইম** পথ কিন্তু **কম হ্যাসেল**, এবং **ডেমোর জন্য একদম ফ্রি**। ভেতরে আছে — চূড়ান্ত আর্কিটেকচার ও কেন এটাই সেরা, পুরো data flow, cross-platform ডেলিভারি, ফিচার, ধাপে-ধাপে রোডম্যাপ, কমিট প্ল্যান, ২ জনের কাজ ভাগ ও খরচের হিসাব।

১ কী বানাচ্ছি — চূড়ান্ত স্কোপ

SCOPE

তুমি ফাইনাল স্টেজ চাও — সিস্টেমটা **অক্ষর, শব্দ ও পূর্ণ বাক্য** সব চিনে কণ্ঠে বলবে। কাজটা তিন স্তরে, একটার ওপর আরেকটা পড়ে তোলা হবে (একসাথে নয়):

স্তর ১
অক্ষর finger-spelling, static — সহজ ভিত্তি

স্তর ২
শব্দ dynamic gesture, sequence মডেল

স্তর ৩
বাক্য + কণ্ঠ শব্দ জুড়ে স্বাভাবিক বাংলা

২ চূড়ান্ত আর্কিটেকচার — ও কেন এটাই সেরা

THE DECISION

তোমার শর্ত ছিল — **একদম রিয়েল-টাইম, ভালো স্পিড, কিন্তু বেশি হাসেল নয়**। দুটো একসাথে পেতে এই ডিজাইন:

⚙️ সিদ্ধান্ত

চেনা (VISION + MODEL)
On-device ব্রাউজারেই — MediaPipe JS + TF.js। Network hop নেই, তাই সবচেয়ে fast (~৫০ms)।

ভাষা (শব্দ → বাক্য)
Gemini API শুধু বাক্য সুন্দর করতে, async — রিয়েল-টাইম আটকায় না।

ডেলিভারি
এক ওয়েব ইঞ্জিন PC/iOS/Android সবখানে। আলাদা plugin/সার্ভার লাগে না।

কেন backend সার্ভার নয়?

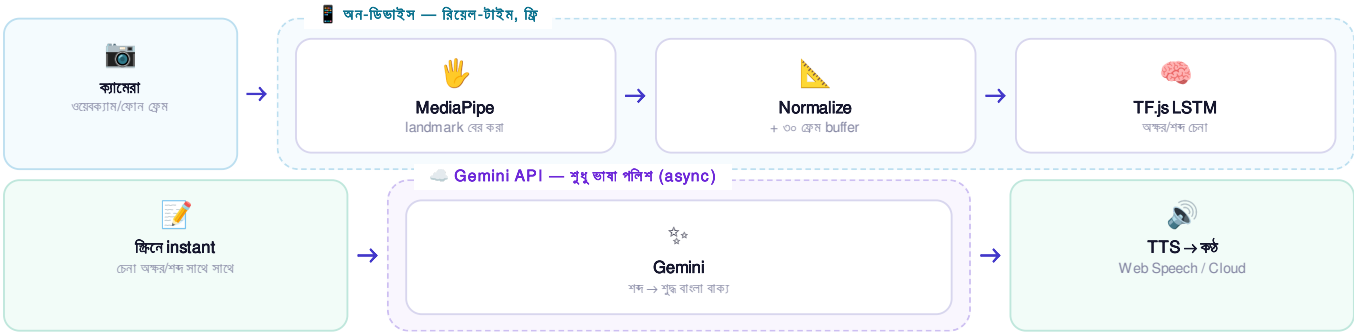
সার্ভারে প্রতি ফ্রেম পাঠানো-আনায় network delay বাড়ে — on-device তার চেয়ে fast। আর সার্ভার চালু রাখা, স্কেল করা ২ জনের জন্য extra হাসেল।

কেন native Flutter plugin নয়?

MediaPipe-এর Flutter সাপোর্ট খণ্ডিত — Android/iOS/web-এ আলাদা প্যাকেজ। ওয়েব ইঞ্জিন একবার লিখলেই সব জায়গায় চলে, কাজ অর্ধেক।

৩ সিস্টেম কীভাবে চলে — Data Flow

ভারী ও গতির অংশটা (নীল) ফোন/পিসিতেই চলে; শুধু ভাষা পলিশ (বেগুনি) API-তে যায়:



✓ দ্রুত আর নির্ভুল — দুটোই

চেনা অক্ষর/শব্দ স্ক্রিনে সাথে সাথে দেখাবে (on-device, ~৫০ms)। সুন্দর পূর্ণ বাক্য ও কণ্ঠ আসবে এক beat পরে — অনেকটা লাইভ ইন্টারপ্রেটারের মতো।

এতে রিয়েল-টাইম ফিল নষ্ট হয় না, কারণ ব্যবহারকারী চেনা শব্দ তখনই দেখছে; Gemini-র polish ব্যাকগ্রাউন্ডে চলে। Internet না থাকলেও অন্তত অক্ষর/শব্দ চেনা ও Web Speech কণ্ঠ চলবে।

৪ এক ইঞ্জিন, সব প্ল্যাটফর্ম

CROSS-PLATFORM

কঠিন অংশটা (ওয়েব ইঞ্জিন) একবার বানাও — তারপর তিনভাবে সবার কাছে পৌঁছাও:

ওয়েব ইঞ্জিন
MediaPipe JS + TF.js + Gemini + TTS
(একবার বানানো)

- **PC ব্রাউজার**
  তোমার "PC website" — Chrome/Edge/Safari-তেই চলে
- **PWA (iOS + Android)**
  ফোনে install হয়, অ্যাপের মতো, অফলাইনেও খোলে
- **Flutter shell (optional)**
  App Store / Play-তে দিতে চাইলে WebView-এ ইঞ্জিন

ছোট সতর্কতা — iOS

iOS-এ in-app WebView-এ ক্যামেরা মাঝে মাঝে ঝামেলা করে। সহজ সমাধান: iOS-এ Safari-তে **PWA** হিসেবে চালাও (ক্যামেরা ঠিক কাজ করে), Flutter shell পরে যোগ করো।

৫ চূড়ান্ত টেক স্ট্যাক

TECH STACK

চেনা (ON-DEVICE) **MediaPipe Tasks (JS/WASM) + TensorFlow.js**	ভাষা স্তর **Gemini API** — Flash-Lite, ফ্রি tier
কণ্ঠ (TTS) **Web Speech API** (ফ্রি) → Cloud/Gemini TTS (upgrade)	অ্যাপ / UI **Next.js + TS + Tailwind** + PWA + Flutter shell
মডেল ট্রেনিং **Python + Kaggle/Colab ফ্রি GPU** → export to TF.js	হোস্টিং + কোড **Vercel** (ফ্রি) · **Git + GitHub**

৬ চেনার পদ্ধতি ও ডেটাসেট

MODEL & DATA

রেডিমেড BdSL মডেল নেই — তাই একটা **হালকা মডেল train** করতে হবে। ভয়ের কিছু নেই: MediaPipe প্রতি ফ্রেমে হাতের ছোট **landmark vector** দেয় (raw pixel নয়), তার ওপর ছোট LSTM train হয় — Kaggle/Colab-এর ফ্রি GPU-তে কয়েক ঘণ্টায়।

"Dictionary" = ডেটাসেটগুলোই (label already বাংলায়, তাই বাংলা কনভার্সন ফ্রি-ইন):

- **Ishara-Lipi** — প্রথম ওপেন BdSL ক্যারেক্টার সেট
- **BAUST Lipi** — ৩৬ অক্ষর · ১৮,০০০ ছবি
- **BdSL36** — ৩৬ বাংলা অক্ষর/সংখ্যা
- **BdSL Words** — ১০টি দৈনন্দিন শব্দ

বেশি শব্দের জন্য

স্তর ২–৩ এ vocabulary বাড়াতে তোমরা ২ জন নিজেরা ২০–৩০টা common শব্দ (যেমন: পানি, খাবার, সাহায্য, ধন্যবাদ) রেকর্ড করো ভিন্ন আলো/ব্যাকগ্রাউন্ডে — এতে নির্ভুলতা ও প্রতিযোগিতার স্কোর দুটোই বাড়বে।

৭ ভাষা স্তর (Gemini) ও কণ্ঠ (TTS)

LANGUAGE & VOICE

Gemini-র কাজ: চেনা শব্দগুলোকে স্বাভাবিক বাংলায় গাঁথা।

```
[আমি, স্কুল, যাওয়া]
→ "আমি স্কুলে যাচ্ছি"
```

prompt: "এই শব্দগুলো দিয়ে একটি শুদ্ধ বাংলা বাক্য বানাও।" async চলে, ফ্রি tier যথেষ্ট।

কণ্ঠ (TTS) — দুই স্তরে:

ডিফল্ট: Web Speech API — ব্রাউজারেই, ফ্রি, instant (Android-এ বাংলা কণ্ঠ আছে)।

Upgrade: যেখানে কণ্ঠ নেই বা আরও ভালো চাও — Google Cloud TTS বাংলা / Gemini TTS / AI4Bharat Indic-TTS।

৮ ফিচার রোডম্যাপ

FEATURES

তিন স্তরে গড়ে তোলো — আগেরটা দাঁড়ালে পরেরটা:

স্তর ১ — ভিত্তি
আগে

- রিয়েল-টাইম অক্ষর চেনা (finger-spelling)
- লাইভ ক্যামেরা + হাতের landmark ওভারলে
- চেনা অক্ষর স্ক্রিনে instant
- Web Speech দিয়ে অক্ষর/শব্দ উচ্চারণ
- কনফিডেন্স স্কোর

স্তর ২ — শব্দ
এরপর

- dynamic শব্দ চেনা (sequence)
- শব্দ জমিয়ে রাখা + hold-to-confirm
- নিজেদের রেকর্ড করা শব্দ যোগ
- সংখ্যা ও quick phrases
- ট্রান্সক্রিপ্ট সেভ/এক্সপোর্ট

স্তর ৩ — বাক্য + Wow
শেষে

- Gemini দিয়ে স্বাভাবিক বাংলা বাক্য
- পূর্ণ বাক্য কণ্ঠে বলা (ভালো TTS)
- কণ্ঠ/স্পিড নির্বাচন, ডার্ক মোড
- প্র্যাকটিস মোড (BdSL শেখানো)
- PWA install + অফলাইন মোড

৯ স্টেপ-বাই-স্টেপ রোডম্যাপ

কোনটার পরে কোনটা

0 ফেজ ০ — সেটআপ
Git রিপো, README, ফোল্ডার, ২ জনের কাজ ভাগ, Gemini key সেট।

1 ফেজ ১ — ডেটা ও landmark
BdSL ডেটাসেট → MediaPipe দিয়ে landmark বের করা → train/test।
▸ মডেল-রেডি ডেটা

2 ফেজ ২ — মডেল ট্রেনিং (অক্ষর)
LSTM train (ফ্রি GPU) → ইভ্যালুয়েট → TF.js-এ export।
▸ অক্ষর চেনা মডেল

3 ফেজ ৩ — ওয়েব রিয়েল-টাইম ইঞ্জিন
ব্রাউজারে MediaPipe JS + TF.js → লাইভ ক্যামেরা → instant অক্ষর + smoothing।
▸ লাইভ চেনা ব্রাউজারে

4 ফেজ ৪ — শব্দ ও বাক্য
dynamic শব্দ মডেল → শব্দ জমা → Gemini দিয়ে বাংলা বাক্য।
▸ শব্দ + বাক্য

5 ফেজ ৫ — কণ্ঠ (TTS)
Web Speech যুক্ত → async প্রেব্যাক → Cloud TTS upgrade।
▸ ইশারা → কণ্ঠ

6 ফেজ ৬ — UI + PWA
ক্যামেরা ফিড, লাইভ ক্যাপশন, সেটিংস, রেসপন্সিভ, PWA install।
▸ সব প্ল্যাটফর্মে অ্যাপ

7 ফেজ ৭ — পলিশ, টেস্ট ও ডিপ্লয়
লেটেন্সি টিউন, এরর হ্যান্ডলিং, ডেমো ভিডিও, Vercel ডিপ্লয় (+ Flutter shell)।
▸ প্রতিযোগিতা-রেডি, লাইভ লিংক

১০ কমিট প্ল্যান — কয়টা কমিট?

২ জনের এই আর্কিটেকচারে ছোট atomic কমিট ধরে আনুমানিক **৮০–১০০টি কমিট**:

#	ফেজ	কমিট	নমুনা কমিট মেসেজ
০	সেটআপ	৭	`chore: init repo + gitignore` `docs: add README`
১	ডেটা ও landmark	১১	`feat: mediapipe landmark extraction` `feat: train/test split`
২	মডেল ট্রেনিং (অক্ষর)	১৩	`feat: LSTM training` `feat: export to tfjs`
৩	ওয়েব রিয়েল-টাইম ইঞ্জিন	১৫	`feat: mediapipe js in browser` `feat: live prediction` `fix: smoothing`
৪	শব্দ ও বাক্য	১৪	`feat: dynamic word model` `feat: gemini sentence builder`
৫	কণ্ঠ (TTS)	৮	`feat: web speech tts` `feat: cloud tts fallback`
৬	UI + PWA	১৪	`feat: caption panel` `feat: pwa manifest` `style: responsive`
৭	পলিশ, টেস্ট ও ডিপ্লয়	১০	`perf: reduce latency` `docs: demo` `chore: deploy vercel`
	মোট আনুমানিক	**~৯০**	পরিসর ৮০–১০০

প্রতিটা কমিট সুন্দর রাখার নিয়ম:

ছোট, atomic কমিট
একটাই যৌক্তিক পরিবর্তন প্রতি কমিটে।

Conventional Commits
`feat: fix: docs: refactor: test: chore:`

ফিচার শেষেই কমিট
শেষে জমিয়ে নয়, তখনই।

Feature branch + PR
main সবসময় কাজ করা অবস্থায়।

.gitignore ঠিকঠাক
node_modules, মডেল weight, venv বাদ; বড় ফাইল LFS।

মাইলস্টোন ট্যাগ
`v0.1-letters · v0.2-words · v1.0-demo`

১১ ২ জনের কাজ ভাগ

TEAM OF 2

১

Vision / ML Lead

ডেটা + MediaPipe + মডেল ট্রেনিং + on-device ইঞ্জিন ইন্টিগ্রেশন (ফেজ ১–৩, শব্দ মডেল)

২

App Lead

Web/PWA/Flutter UI + Gemini ভাষা + TTS + ডিপ্লয় + ডেমো (ফেজ ৩–৭)

দুজনে নিজ নিজ feature branch + PR। ছোট টিম, তাই কঠিন অংশে (যেমন রিয়েল-টাইম টিউনিং) pair programming করো। ফেজ ৩-এ দুজনের কাজ মেলে — সেখানে একসাথে।

১২ খরচের হিসাব — ডেমো ~$0

COST

অংশ	টুল	খরচ
Vision / চেনা	MediaPipe JS (on-device)	ফ্রি
মডেল ট্রেনিং	Kaggle / Colab GPU	ফ্রি
ভাষা স্তর	Gemini free tier (Flash-Lite)	ফ্রি · ~১৫০০/দিন
কণ্ঠ (TTS)	Web Speech API	ফ্রি
হোস্টিং	Vercel	ফ্রি
ডেটাসেট	ওপেন BdSL ডেটাসেট	ফ্রি
মোট (ডেমো/প্রতিযোগিতা)		**~$0**

পরে scale করলে অল্প খরচ: Gemini paid (সস্তা Flash-Lite), Cloud TTS, ভালো হোস্টিং — মাসে $৫–২০ এর মধ্যে।

১৩ ক্যামেরা ও টেস্টিং

CAMERA

কোন ক্যামেরা?

Dev/দ্রুত টেস্টে **ল্যাপটপ ক্যামেরা**। ডেটা কালেকশন ও ফাইনাল টেস্ট **ফোনে** (অ্যাপ ফোনেই চলবে, ক্যামেরাও ভালো)।

Accuracy-র আসল চাবি

ভালো আলো · হাত+মুখ+শরীর ফ্রেমে · 720p+ · স্থির অবস্থান। এগুলো ক্যামেরা ব্র্যান্ডের চেয়ে বেশি জরুরি।

১৪ চ্যালেঞ্জ ও সমাধান

WATCH OUT

আলো/ব্যাকগ্রাউন্ড ভেদে নির্ভুলতা — সমাধান: বৈচিত্র্যময় ডেটা + augmentation।

দুই হাত + মুখভঙ্গি (BdSL ব্যাকরণ) — সমাধান: শুধু হাত নয়, MediaPipe Holistic ব্যবহার।

একই শব্দ বারবার ধরা — সমাধান: debounce / hold-to-confirm।

iOS WebView ক্যামেরা — সমাধান: iOS-এ Safari PWA হিসেবে চালানো।

Web Speech-এ বাংলা কণ্ঠ না থাকা (কিছু ডিভাইসে) — সমাধান: Cloud/Gemini TTS fallback।

মডেল ব্রাউজারে ভারী হওয়া — সমাধান: ছোট LSTM, quantize, GPU delegate।

শুরু করো ফেজ ০ → ১ → ২ দিয়ে — অক্ষর আগে

স্তর ১ (অক্ষর) ব্রাউজারে চললেই বাকিটা — শব্দ, বাক্য, কণ্ঠ — একটার ওপর আরেকটা দাঁড়িয়ে যাবে। প্রতিটা ছোট জয় পরিষ্কার মেসেজে কমিট করো। সব সম্ভব। 🚀

তথ্যসূত্র: BAUST Lipi (arXiv:2408.10518) · BdSL36 (arXiv:2110.00869) · Ishara-Lipi (ICBSLP 2018) · MediaPipe Holistic + LSTM real-time SLR (~৯২–৯৭%, ~৫০ms/frame; IJRASET 2025) · MediaPipe web/JS ও Flutter সাপোর্ট (pub.dev, MediaPipe docs) · Gemini API ফ্রি tier — Flash/Flash-Lite, ~১৫০০ RPD (ai.google.dev, ২০২৬) · বাংলা TTS: Web Speech API, AI4Bharat Indic-TTS, bangla-tts।