Research and development

Objective

Develop a lightweight, intuitive mobile language learning application tailored to the needs of refugees living in the accommodation of IMA in the South of Limburg. The aim is to provide accessible, visual, and structured Dutch learning through technology, compensating for the lack of available in-person language education.

Target Audience

This project has a very clear target audience, Refugees residing at IMA Projects in the South of Limburg. They have some characteristics that have a big influence on the project.

These characteristics include:

- Limited literacy or digital literacy
- Minimal access to formal education
- Use older phones with limited resources
- Limited WiFi access
- Motivated to learn Dutch to integrate and participate in society

Insights from user testing:

Observation	**Insight**	**Action**
Pronunciation is hardest	Focus on "NG", "NK", "G", "SCH", "U"	Dedicated pronunciation module
People need real-life sentences	Add common-use sentences (e.g., “How much does it cost?”)	Prioritize conversational modules
Illiterate users rely on visuals	Use image + audio initially, remove images in exams	Layered learning
Too many features	Some users overwhelmed	Narrow down MVP: Word, Sentence, Pronunciation
Limburg accent is hard	Accent is difficult to understand	Use standard Dutch audio
QR code idea popular	Users want learning tied to real places	Add location-based QR scanning
Motivation drops fast	Users lose interest quickly	Add game elements, mascot encouragement
Interface looks like Duolingo	Feels unoriginal	Redesign UI with Dutch theme (e.g., tulips, paths)
Image-size imbalance	Users pick images instead of reading words	Make words larger than pictures
Verb explanation unclear	Confusion about grammar	Add tooltips with translation and grammar info

Want challenge at end of level	Need to test knowledge without visual aids	Add exams with no images
Flags issue (Kurdish)	Multiple Kurdish flags exist	Use neutral or selectable flag options

Tech stack recommendation (updated)

Layer	Tech	Why
Frontend	Next.js + TailwindCSS	Fast, responsive, scalable; works well with old phones
State Mgmt	Zustand	Simple and lightweight
Backend	Fastify (Node.js)	Fast, efficient, good for mobile API
Database	PostgreSQL (Prisma ORM)	Reliable tracking, user data, lessons
AI Models	- GPT-4-turbo (text AI) - Whisper (speech-to-text) - ElevenLabs (optional TTS) - Ollama (offline fallback)	Covers pronunciation, translation, and interaction
Deployment	Hera	Light deployment for low WiFi environments
Other Tools	Redis for caching, QR code generator	Faster load and real-world interaction

Validation Through Prototyping

Showcase Results:

The Figma prototype was well received by the Stake holder (Cloé).

Key validation points:

- Visual flow is intuitive
- Navigation works well without needing reading
- Users liked mascot idea and game path
- Too many features = feedback to cut down for MVP
- QR code idea strongly validated
- Users appreciated local context (common words, phrases)

User Flow Summary

1. App opens → auto account created
2. Choose native language
3. Choose module: Game or Learn
4. Choose Word, Sentence, or Pronunciation

5. Select Level (1–5)
6. Learn with visual & audio
7. End of level exam (no images)
8. Mascot gives feedback
9. Return or proceed

Sprint Plan

Sprint 1	Build core MVP: Word/Sentence/Pronunciation modules, QR code support, Visual UI
Sprint 2	Polish design, integrate mascot, add levels/exams, user testing
Sprint 3	Apply feedback, refine UI/UX, finalize deployment

Conclusion

Your app concept is validated. By focusing on pronunciation, daily-use Dutch, and visual learning, the app is solving a real problem in an effective way. The feedback confirms that your direction is strong but I have to narrow the scope for the MVP, and focus on depth. Once the core is strong, expansion will be easier.