







LinguaLearn – Language Learning Application

A full-featured, modern web application for mastering new languages using advanced NLP, AI-powered translation, and interactive learning tools.

Features

-  **Text Translation:** Seamlessly translate any text across multiple languages using Lecto AI API
-  **Part-of-Speech Tagging (POS):** NLP-driven linguistic analysis to identify grammar roles in your sentences using spaCy
-  **Interactive Flashcards:** Auto-generate vocabulary flashcards from translated content with a clean UI
-  **Spaced Repetition System:** Practice flashcards based on learning science to improve retention
-  **Inspirational Quotes:** Pull in daily motivational quotes using the ZenQuotes API to keep you inspired
-  **Multi-language Support:** Support for English, Spanish, French, German, Chinese, and more

NLP & AI Techniques Used

spaCy-based POS Tagging

We use spaCy for efficient linguistic processing and universal part-of-speech tagging across languages.

Key techniques:

- Tokenization
- POS tagging (`token.pos_`)
- Language model switching (`en_core_web_sm`, `fr_core_news_sm`, etc.)
- Universal POS mapping to maintain consistent tag representation across languages

Lecto AI Translation

Lecto AI provides fast and accurate multilingual translation, used to power real-time translation and vocabulary extraction.

Languages supported: 20+ including English, Spanish, French, German, Chinese, Japanese, Hindi, etc.

Google Gemini AI (Fallback)

In cases where POS analysis isn't available via spaCy (e.g., unsupported languages), we optionally use Gemini AI (Google's Generative AI) to extract linguistic and semantic data using prompt engineering.

Technologies Used

Frontend	Backend	AI & NLP	Styling
React + TypeScript	Flask (Python)	spaCy, Gemini AI	Tailwind CSS
React Router	Flask-CORS	Lecto AI API	CSS Modules
LocalStorage	REST APIs	ZenQuotes API	

Project Structure

```
language-learning-app/
├── public/                # Static assets
├── src/
│   ├── components/       # React UI components
│   ├── services/         # API interaction handlers
│   ├── hooks/            # Custom React hooks
│   ├── utils/            # Utility functions
│   ├── types/            # TypeScript types and interfaces
│   ├── App.tsx           # Main App component
│   └── index.tsx         # React entry point
├── backend/
│   ├── app.py            # Flask app with POS tagging logic
│   ├── setup.py          # Virtual environment & model setup
│   └── requirements.txt  # Python dependencies
└── README.md
```

What We Learned

Implementing POS Tagging

- Learned how to handle multilingual NLP with spaCy and lazy loading language-specific models
- Mapped spaCy's native POS tags to a universal standard across different languages for UI consistency
- Built fallback mechanisms to gracefully handle missing models

Working with Translation APIs

- Used Lecto AI API to build translation pipelines integrated with user input
- Designed a frontend flow that integrates translated content with entity tagging and flashcard generation
- Handled language detection, selection, and translation feedback to users

Integrating External AI Services

- Integrated Google Gemini AI for prompt-based language analysis as a backup for spaCy's capabilities
- Used structured prompt templates to extract linguistic features like parts of speech, root verbs, and more

Vocabulary Training Design

- Applied learning psychology (Spaced Repetition) to the flashcard design
- Allowed users to curate their own learning decks directly from analysis results

Getting Started

Prerequisites

- Node.js (v14+)
- Python 3.8+
- Lecto AI API key (Free tier works)
- Google Gemini AI API key (optional for fallback)

Installation & Setup

1. Clone the repository:

```
bash

git clone <repository-url>
cd language-learning-app
```

2. Install frontend dependencies:

```
bash

npm install
```

3. Configure environment variables: Create a `.env` file:

```
env

REACT_APP_LECTO_API_KEY=your_lecto_api_key_here
REACT_APP_GEMINI_API_KEY=your_gemini_api_key_here
REACT_APP_BACKEND_URL=http://localhost:5000/api
```

4. Set up backend:

```
bash
```






```
cd backend  
python setup.py # Installs venv, dependencies, spaCy models  
source venv/bin/activate # On Windows: venv\Scripts\activate  
python app.py
```

5. Start frontend:

```
bash
```

```
cd ..  
npm start
```

Usage Guide

1.  Enter a phrase or sentence in your native language
2.  Choose a target language and click "Translate & Analyze POS"
3.  View tagged words with grammar roles like NOUN, VERB, ADJ, etc.
4.  Click on interesting words to turn them into flashcards
5.  Go to "Practice" to test yourself using spaced repetition

Documentation & References

- [spaCy Official Docs](#)
- [Lecto AI Docs](#)
- [ZenQuotes API](#)
- [Google Generative AI](#)
- [Tailwind CSS](#)
- [Flask](#)
- [React Docs](#)

License

MIT License – Free to use, modify, and distribute.

Acknowledgments

- NLP engineers and linguists who build open-source tools like spaCy
- Open API providers who enable learning tools for everyone
- The language learning community for continuous feedback and inspiration

