



# Lanex

---

Juraj B.

Morad E.

Zhaohan M.

# The design of the application

- Foundational motive from the start was improving interconnectivity between language learners to improve the language learning experience.
- Based on the concept of language requests being created by users.
- Almost any request activity seeking language interactions is fair game (as long as it's feasible and sensible).
- And if users see a request they like, they can post comments or message original creators of the requests to participate.
- As a way to help users better find requests that might interest them, languages are filtered under categories but a direct search functionality is also provided,

# Inspiration for the application

---

The world is big with many people in it, containing people who try to pursue to learn languages and others who try to teach what language knowledge they possess.

---

Generally simple finding others who speak the same tongue where one lives, but finding individuals who speak foreign languages to practice with can be a challenge.

---

Language learning , in terms of finding individuals to connect and share language knowledge with, should be more accessible.

---

All in all, people passionate about languages deserve an easier way to meet others like them and to experience together their language journeys.

---

Thus we came to make Lanex - to make this possible.

# Rundown of how the application works

Guest/unregistered users can view requests, though they can't comment or send messages to request owners.

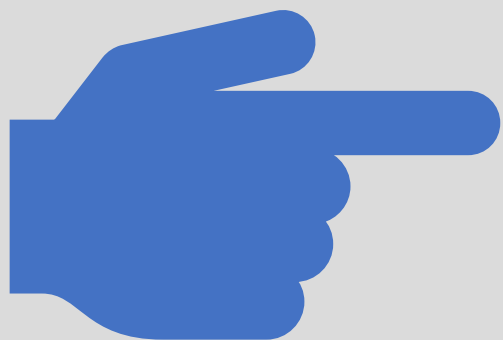
Registered users can comment on requests. Likewise, they can message request owners by clicking on the Accept Request option, followed by an email message to the owner of the request, mentioning their interest (user's default email client used for this)

A request form is provided for creating language requests, with fields to fill in relating to the specific request.

A default location is provided on the map: so for online-related language requests, users can keep this default or set their home location. (The application encourages in-person language events in any case.)

# Technologies used in building the web application

- \* [Django] –Python web framework used for building the application
- \* [Pillow] - For opening, manipulating, and saving images.
- \* [Django-Extensions] – Extending the original Django framework by providing more features.
- \* [Django-Location-Field] – For integrating map functionality from data.
- \* [OpenStreetMaps+Leaflet.js-library] – API and libraries helping implement the map feature in requests.
- \* [Django-Registration] – For user registration functionality
- \* [jQuery] – JS library helping to get more done faster.
- \* [Bootstrap, Bootdev, Bootstrap-examples] – Game changing CSS framework
- \* [Bootsnipp] – Providing an element gallery.
- \* [BIng-Search] – API used for the search results



# When Lanex meets PythonAnywhere

- Lanex is available on Github:  
<https://github.com/MoradEnCours/Lanex>
- But, we will run our demonstration on PythonAnywhere:  
<http://lanex.pythonanywhere.com/>
- Let's get started.

# Team member contributions

- Juraj: Involved in the front-end work.  
[AKA - “The Bootstrap Boss, The Separator of Concerns”]
- Morad: Involved in the back-end work.  
[AKA – Me, “The dude at the back presenting”]
- Zhaohan : Involved in a mix of front and back-end work  
[AKA – “The Thinker and Syncer”]

# What might a Lanex 2.0 look like?

---

If we were to redesign Lanex, these are top three features we would add:

---

An intrinsic messaging system – to allow for better greater application independence and to give added flexibility in communication.

---

A friend-followship system – to simplify users needing to follow other users' profiles who they like manually just to check their requests.

---

The ability to modify created requests –providing a means for users to go back and revise errors or update information in light of circumstances.

---





**Lanex**

End of the  
presentation