Talk-a-Palooza

An AI-powered app to boost children's language acquisition

Featuring our amazing mentors:

Bogdan Christoph

The Problem

5-10% of preschool-aged children experiencing speech delays

\+

Overburdened Speech Therapy Waitlists

=

Significant impact on a child's language progress during crucial developmental stages

The Solution

Combine the power of **machine learning, convolutional neural networks** and **pediatric speech acquisition research** to quickly **identify areas** where a child needs help, and create c**ustom language acquisition exercises and predictions**

Challenges

Balancing expectations

Working with audio data

Reduced team

Finding audio

Real Life Issues

We're late

Wins

Finding Lingualibre database

Established clear project scope

Improved communication

Progress on MVP for all tracks

Strong and fun team

Track Contributions

UX	WD	DS	DL
Personas and research	Product backlog, sprint planning, kanban board	Understand the research that resulted in the dataset	Align with DS and clearly separate features
Design ideation	Frontend platform (React Native with Expo Web)	Work on an Exploratory Data Analysis	Research of training models with audio data
Wireframes for dashboard, statistics, exercises, initial assessment (in progress).	Backend: system design, database setup, authentication, child assessment	Research for training models with the data. Options: Collaborative Filtering; LibRemmender Work on pre-processing data to apply to an ML model	Work on first feature, recognising single word audio

ux

Laura and Maria

"Mother and daughter"

- 34 and 3
- Dentist
- Europe

Bio

Laura is a first time mother of a three years old child, Maria. She is about to start going to the kindergarten and the teachers in the first meeting session pointed out that Maria would need help with the language development as she is having trouble pronouncing some sounds. Laura is very involved in the wellness of her child and went automatically to the fonoaudiologyst, who recommended to do also some exercises at home with the help of an app.

Maria´s personality

Introvert	Extrovert
Analytical	Creative
Busy	Time rich
Messy	Organized
Independent	Team player
Passive	Active

Martin and Jim

"Father and son"

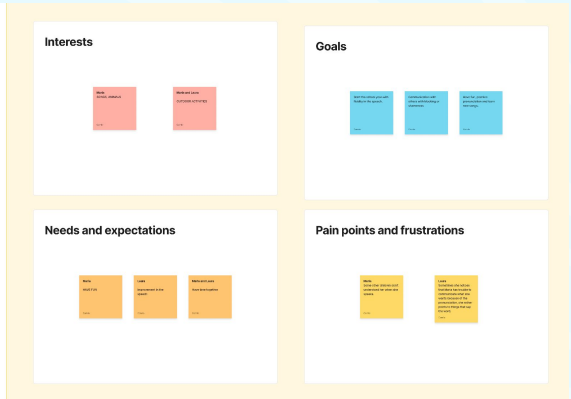
- 28 and 3
- Salesman
- Europe

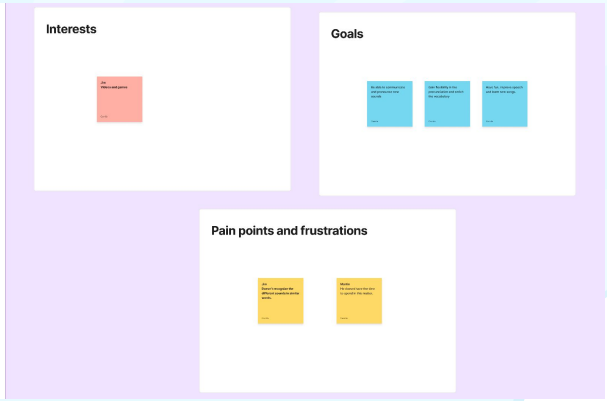
Bio

Martin works full time and is doing also a lot of over hours at work every week. He doesn't have too much time to practice with Jim, who was diagnosed with speech delay. He can comprehend and nonverbally communicate but can't say many words, he can't put them into understandable phrases. Martin needs an app that help Jim with the speech development and that he can easily track. Jim should be able to use the tablet by himself.

Jim´s personality

Introvert	Extrovert
Analytical	Creative
Busy	Time rich
Messy	Organized
Independent	Team player
Passive	Active

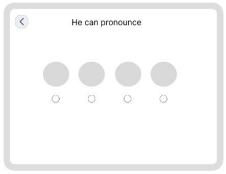


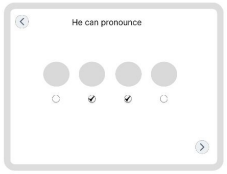


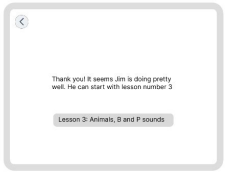
ux

Parent Area

Initial Assessment

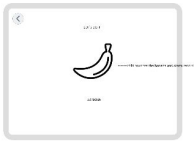


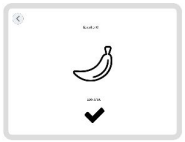


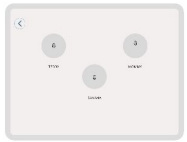


Child Area

Start Exercise

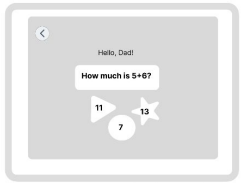




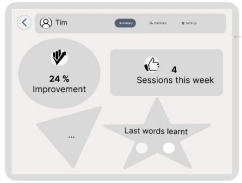


Parent Area

Dashboard

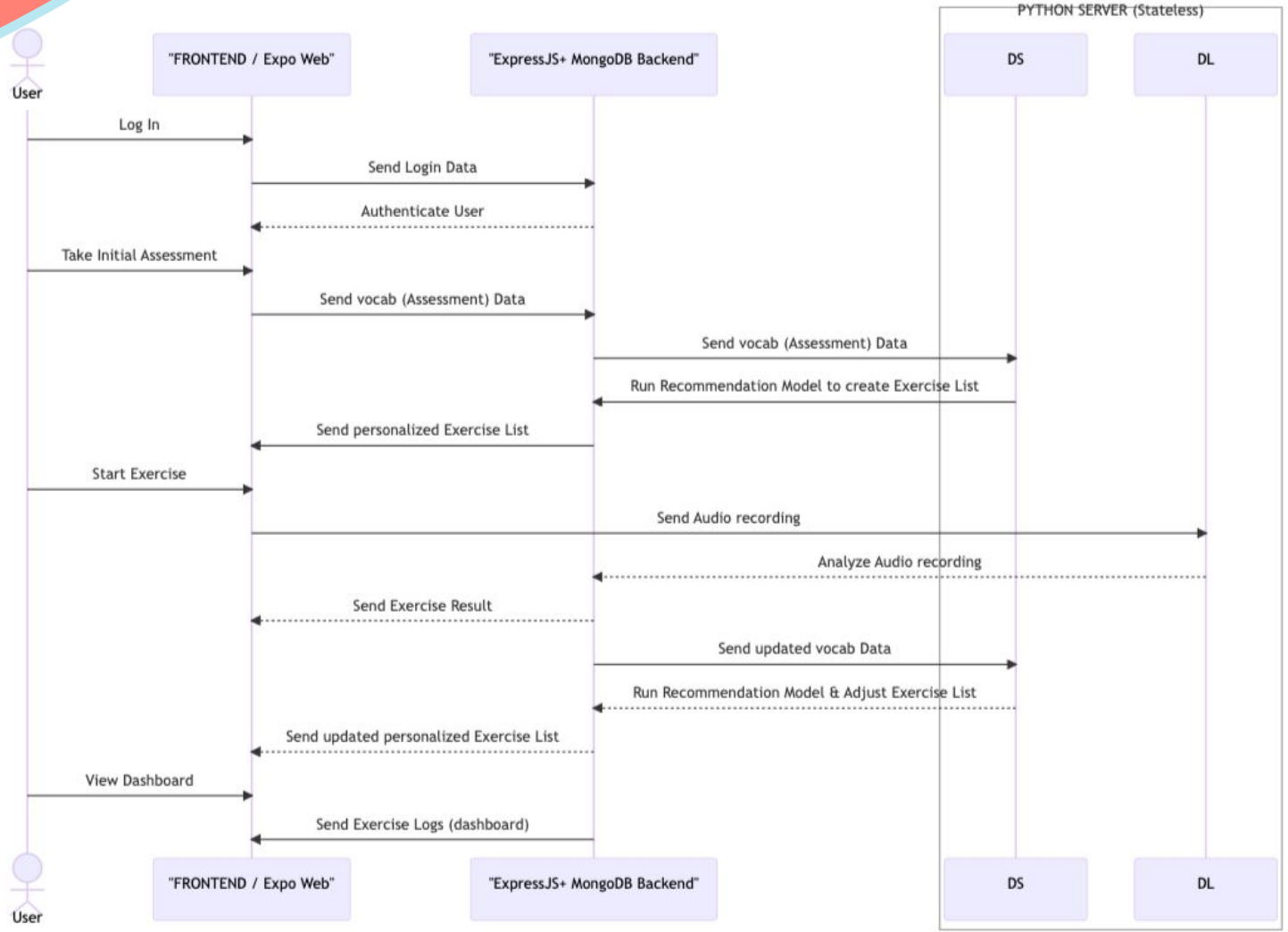


If correct, it sows the dot to the Dashboard

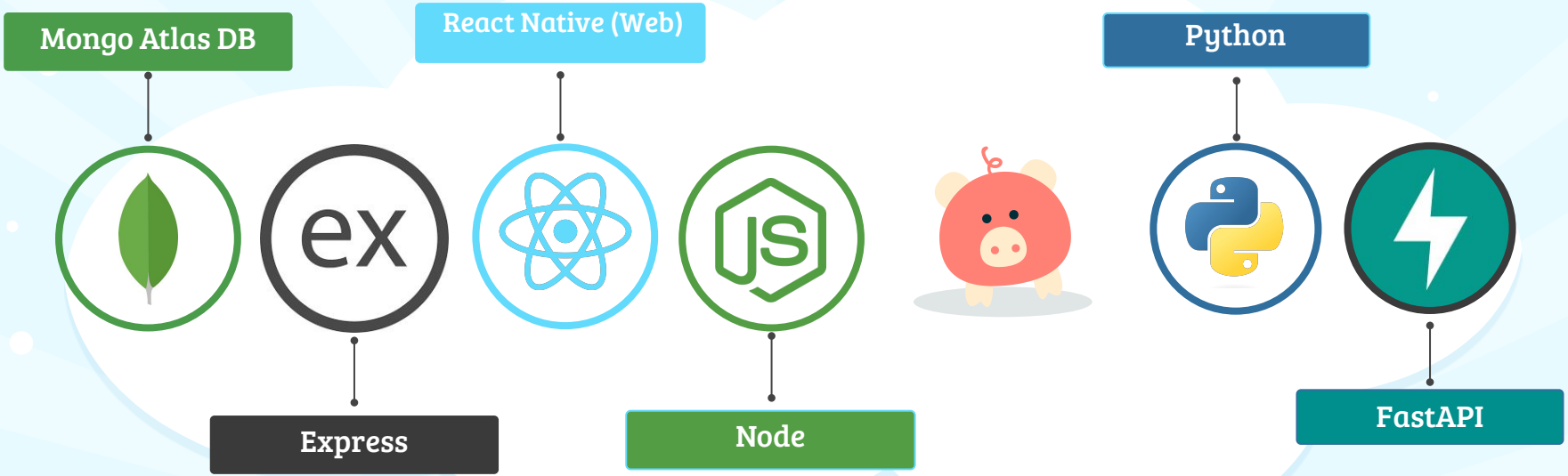


On the summary screen

comparision to other learners?


PYTHON SERVER (Stateless)
User
"FRONTEND / Expo Web"
"ExpressJS+ MongoDB Backend"
DS
DL
Log In
Send Login Data
Authenticate User
Take Initial Assessment
Send vocab (Assessment) Data
Send vocab (Assessment) Data
Run Recommendation Model to create Exercise List
Send personalized Exercise List
Start Exercise
Send Audio recording
Analyze Audio recording
Send Exercise Result
Send updated vocab Data
Run Recommendation Model & Adjust Exercise List
Send updated personalized Exercise List
View Dashboard
Send Exercise Logs (dashboard)
User
"FRONTEND / Expo Web"
"ExpressJS+ MongoDB Backend"
DS
DL

WD


Mongo Atlas DB
ex
Express
React Native (Web)
JS
Node
Python
FastAPI

DL

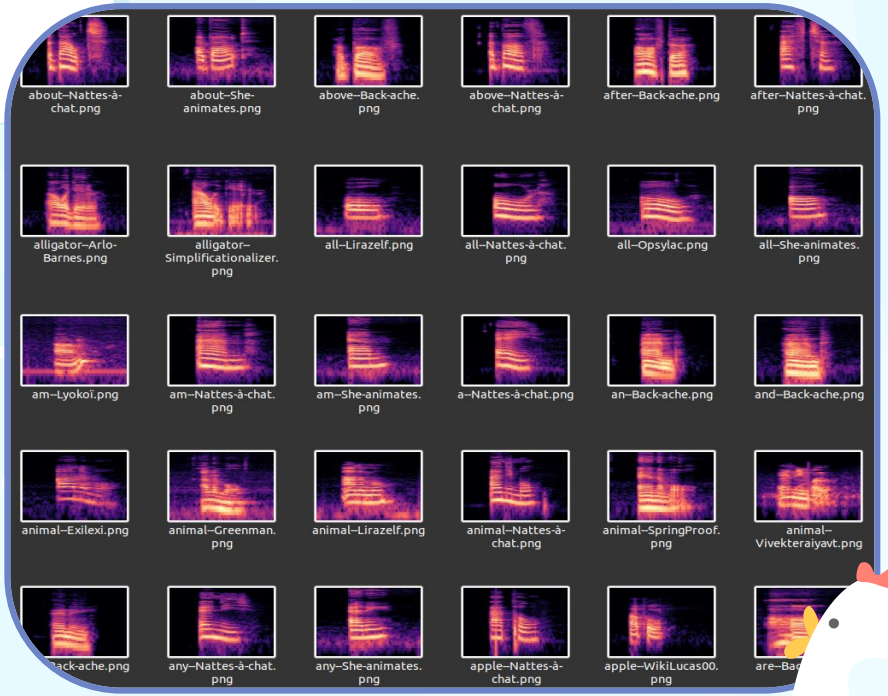
Lingua Libre

Audio dataset: LinguaLibre English dataset

Our sample: a total of 29.1k unique words totalling 33.2k recordings by 109 contributors

Focus: out of the 680 words from Wordbank, this dataset has 477 words in common

DL



Feature: Recognize single word audio and return intelligibility score

Approach: Using FastAi and FastAudio, train models with the 1200+ recordings available for the 477 words in common with Wordbank dataset

Audio to spectrogram

Image processing (CNN)

API: Receive audio, transform to spectrogram and query trained model to see if it's a match

DL

```
teError                                 Traceback (most recent call last)
thon-input-20-90634fcc3c9e> in <cell line: 1>()
-> 1 dls.show_batch()

/usr/local/lib/python3.10/dist-packages/fastai/data/core.py in show_batch(self, b, max_n, ctxs, show, uniqu
rgs)
     98             old_get_idxs = self.get_idxs
     99             self.get_idxs = lambda: Inf.zeros
--> 100         if b is None: b = self.one_batch()
    101         if not show: return self._pre_show_batch(b, max_n=max_n)
    102         show_batch(*self._pre_show_batch(b, max_n=max_n), ctxs=ctxs, max_n=max_n, **kwargs)

/usr/local/lib/python3.10/dist-packages/fastai/data/load.py in one_batch(self)
    135     def one_batch(self):
    136         if self.n is not None and len(self)==0: raise ValueError(f'This DataLoader does not contain
ches')
--> 137         with self.fake_l.no_multiproc(): res = first(self)
    138         if hasattr(self, 'it'): delattr(self, 'it')
    139         return res

/usr/local/lib/python3.10/dist-packages/fastcore/basics.py in first(x, f, negate, **kwargs)
    474     x = iter(x)
    475     if f: x = filter_ex(x, f=f, negate=negate, gen=True, **kwargs)
--> 476     return next(x, None)
    477
    478 # Cell

/usr/local/lib/python3.10/dist-packages/fastai/data/load.py in __iter__(self)
     99         self.before_iter()
    100         self.__idxs=self.get_idxs() # called in context of main process (not workers/subprocesses)
--> 101         for b in _loaders[self.fake_l.num_workers==0](self.fake_l):
    102             if self.device is not None: b = to_device(b, self.device)
    103             yield self.after_batch(b)

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py in __init__(self, loader)
    655 class _SingleProcessDataLoaderIter(_BaseDataLoaderIter):
    656     def __init__(self, loader):
--> 657         super(_SingleProcessDataLoaderIter, self).__init__(loader)
    658         assert self._timeout == 0
    659         assert self._num_workers == 0

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py in __init__(self, loader)
    583         # default behaviour is CUDA device. if pin_memory_device is selected
    584         # and pin_memory is not set, the default behaviour false.
> 585         if (len(loader.pin_memory_device) == 0):
    586             self._pin_memory = loader.pin_memory and torch.cuda.is_available()
    587             self._pin_memory_device = None

rror: '_FakeLoader' object has no attribute 'pin_memory_device'
```

- **Challenges:**
  - How to approach DL with audio?
  - FastAudio causes many errors, both on Colab as well as locally
    - Locked requirements erroneous/incomplete
- **Lookahead:**
  - Replace FastAudio with custom work (transforming to spectrograms, mainly)
  - Provide API
  - Continue to improve model
- **Doubts:** Some words only have a single recording, which calls the validity of testing/training into question

DS

Wordbank

An open database of children's vocabulary development

Our dataset: Wordbank

Our sample: monolingual and English (American) speakers (n = 7601)

Additional information: e.g. age of the child, sex, typically developing or not, birth order, ...

DS

Challenges:

Working with data not personally collected
Wordbank does not provide any documentation on the data
Wordbank has multiple contributors, data not completely standardized across them
Handling such a large dataset/datasets

Achievements:

Cleaning/preprocessing data
Detected quality issues/ discrepancies in the data
Performed initial statistical tests in order to guide further decisions

DS

Models of ML

Collaborative filtering: is a recommendation algorithm that suggests items based on the preferences and behaviors of similar users.
Approaches:

- User-Based Collaborative Filtering;
- Item-Based Collaborative Filtering.

LibRecommender: is an easy-to-use recommender library focused on end-to-end recommendation process. It contains a training(libreco) and serving(libserving) module to let users quickly train and deploy different kinds of recommendation models.

- A hybrid recommender system, which allows users to use either collaborative-filtering or content-based features. New features can be added. Implements several models, like YouTubeRanking,
- Supports dynamic feature and sequence recommendation,
- Supports cold-start prediction and recommendation,
- Easy to retrain model with new users/items from new data.

Next Steps?

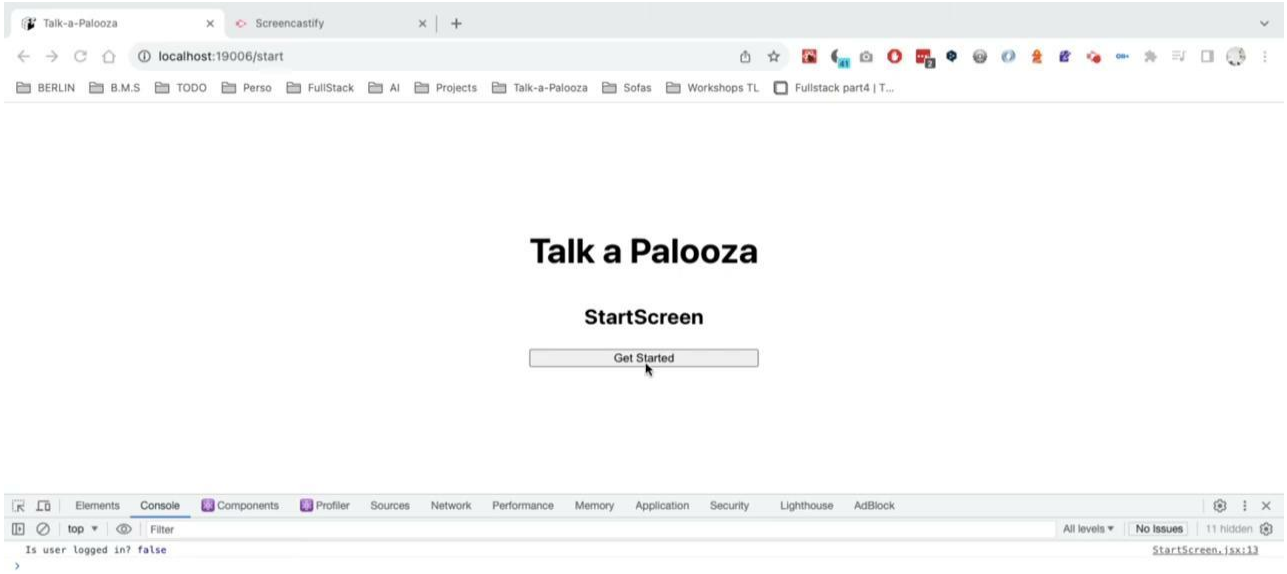
	01	02	03	04	05
 UX	Detailed functional specifications	High Fidelity wireframes	Prototype Figma	User testing to gather feedback for improvements.	**(All)** Final presentation, Testing Fine-tuning, Documenting
 WD	Finish API implementation	Frontend components	UI design Integration	Final adjustments with DS / DL	
 DS	Test models	Process data to improve model performance	Work on API	Final adjustments with WD	
 DL	Replace FastAudio	Improve model by transforming data	Work on API	Final adjustments with WD	

Thank you!

Please feel free to ask any questions. 😄

By Talk a Palooza team, for Techlabs Berlin, SS23

Bonus


Talk-a-Palooza
Screencastify
localhost:19006/start
BERLIN
B.M.S
TODO
Perso
FullStack
AI
Projects
Talk-a-Palooza
Sofas
Workshops TL
Fullstack part4 | T...
Talk a Palooza
StartScreen
Get Started
Elements
Console
Components
Profiler
Sources
Network
Performance
Memory
Application
Security
Lighthouse
AdBlock
top
Filter
All levels
No Issues
11 hidden
Is user logged in? false
StartScreen.jsx:13
Screencastify - Screen Video Recorder is sharing your screen.
Stop sharing
Hide