goobergang : Ziying Jian, Gabriel Thompson, Joshua Liu, Jun Hong Wang

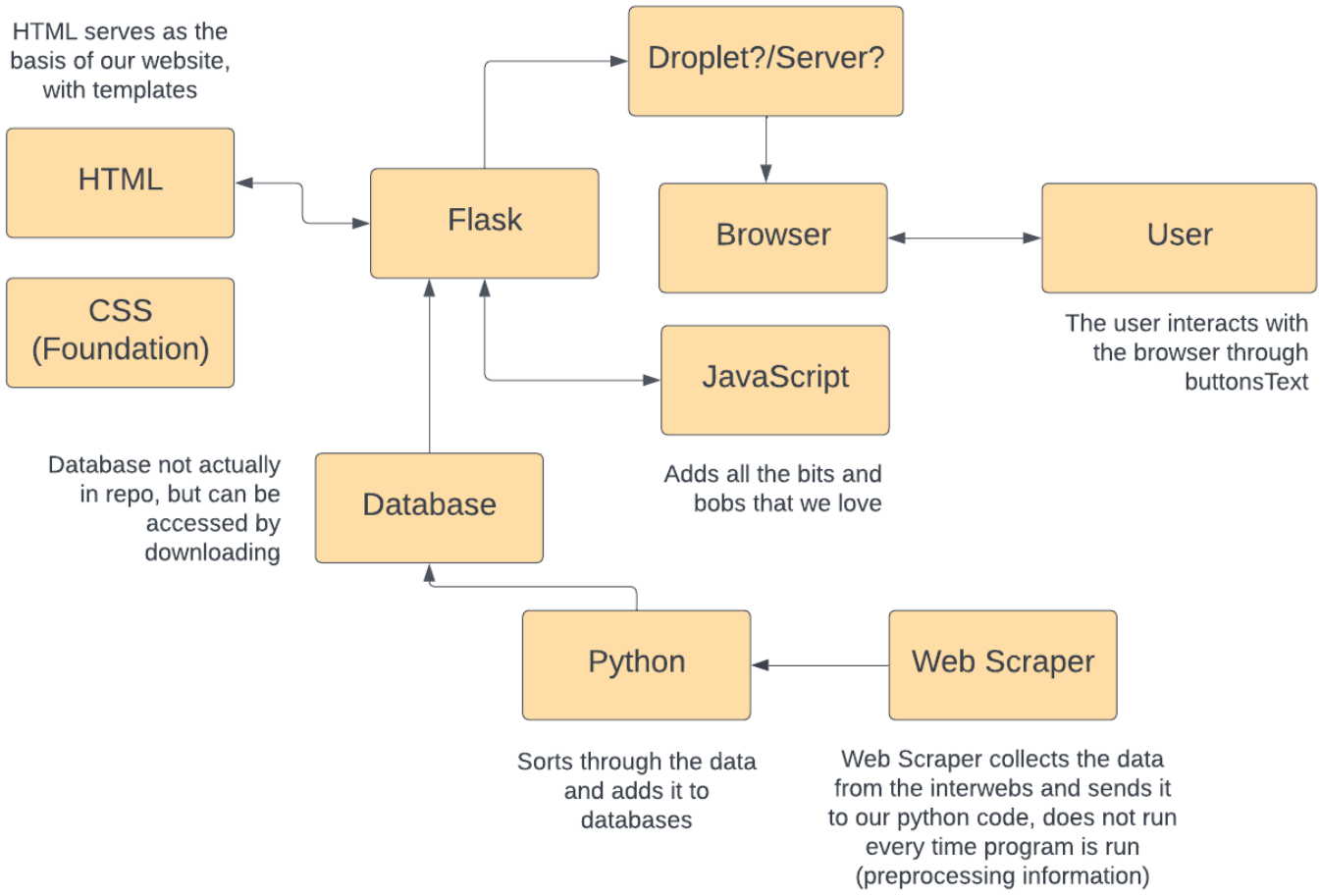
softdev pd8

p04 :: design doc

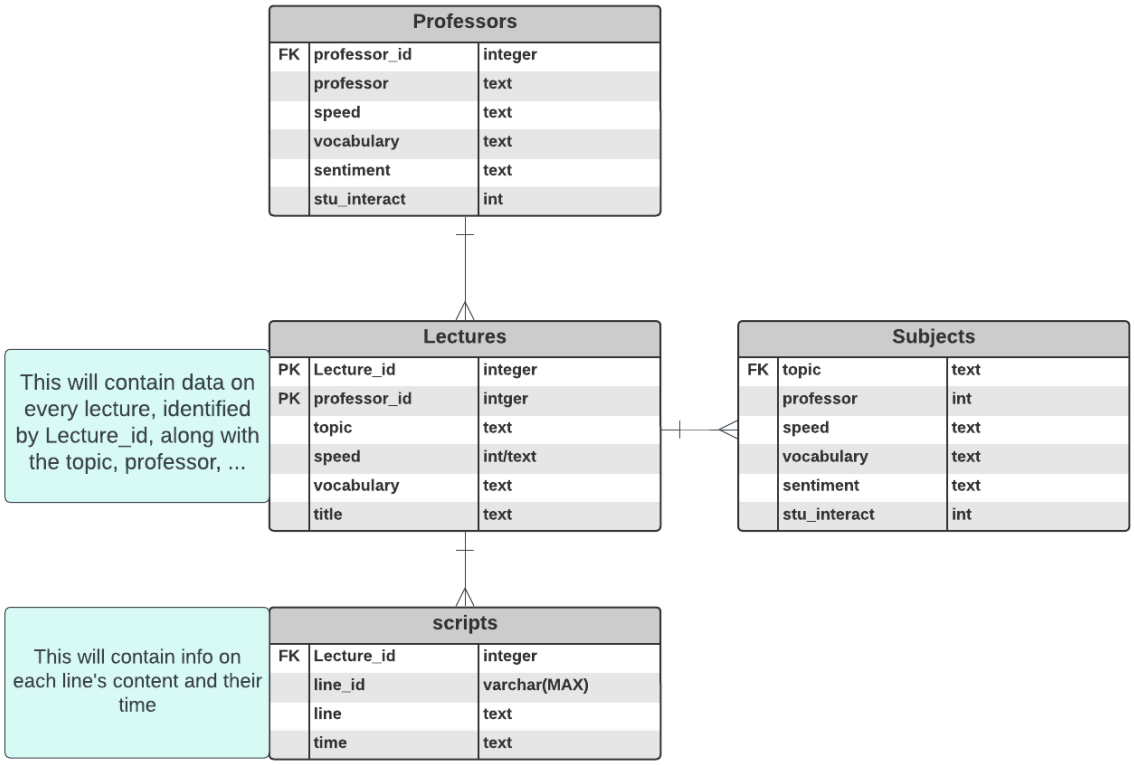
MIT Professor-o-meter

Conducting data analysis on MIT OpenCourseWare video lectures based on professors' speech patterns, such as verbosity, sentiment, speed, and student interactivity.

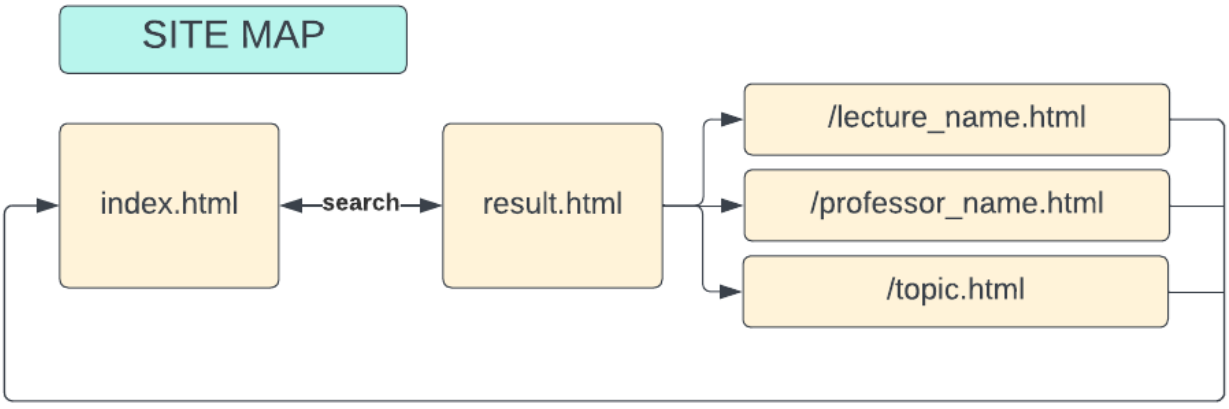
Component Map



Database Organization



Sitemap



Tasks Breakdown

Ziying

- Integrating Foundations
- Building HTML pages
- Routing linkages in Javascript
- Assist with search bar
- ~~Being a PM of three and growing gray hairs~~

Gabriel

- ~~Jaishanker test living rent free in my head~~
- Database population
- Database creation
- Javascript

Joshua

- ~~Laying back and catching up on sleep~~
- Constructing Flask app; template rendering
- Assisting in the creation of HTML templates
- Database manipulation

Jun Hong

- Frontend
- Javascript
- Search bar (unfortunately regex)
- ~~Liking typescript more than javascript~~

APIs Used

No public APIs used. We'll be creating our own Python script to scrape through the MIT OpenCourseWare site, lifting the transcript from the HTML, and storing it in a SQLITE3 database.

Frontend Framework

Foundations because goobergang doesn't have much past experience using it, so we goobers wanted to try it out.

Target Ship Date *05-24-23*

Training data

- Lecture transcripts parsed from MIT OCW, then packaged into a kaggle dataset. This dataset will be a really big csv file.
- In this csv file, each row will represent one lecture. This row will contain:
    - A numerical ID of the lecture
    - The name of the professor
    - The title of the course (we can use this to derive the faculty that is in because of MIT's course numbering conventions)
    - The date of the lecture
    - The text transcript of the lecture
    - Some text indicators of tone (use some NLP library, maybe NLTK?)
    - Also maybe average sentence length

Parts:

- Web scraper to find transcripts of lectures and other information from OCW (will not be visible to page visitors, but useful for getting the csv and uploading to Kaggle, therefore making this data publicly available)?
- Python file for Flask and serving app
- Database (csv or SQL)
- Html file for web page
- Javascript to bring data to life
- Foundation as FEF because PM hasn't used it yet
- CSS?

Data we want to present:

- Verbosity – how wordy the speaker is
- Sentiment – how emotional the speaker is
- Speed – which lecturers talk the fastest
- Student interactivity – how many times audience appears on transcript (even if unintelligible)
- Data by professor
- Data by lecture
- Data by subject/department