IGNITE

Carina Liu
Jacob Hebbel

Project Recap

The IGNITE project is an online repository of back exams that are accessible by all students so they can best prepare for their exams. This will be accomplished with the use of a database to store the pdfs in; we will index this database and display the requested content to the client as both html and as a downloadable document. The website will make extensive use of this database and local caching, so that costly SQL executions are minimized to reduce load.

Information Architecture

The planned information architecture for the IGNITE project starts with a landing page index.html. Users interact with clickable elements on this page to find the major they want back exams for. Clicking navigates them to the second page, classes.html. Users are directed to this page when clicking on any specific major, and classes.html loads all classes that have existing exams within the selected major. From here, users do a similar process to reach exams.html, the page that loads all exams from the selected major/class combination.

index.html

index.html is the first page users see when they access our website. It will have a banner across the top with our project name, as well as the 3 instructions it takes to see any back exam we have. Underneath this banner is a series of interactive tiles. These tiles represent the different majors that the database currently has exams for. The contents of the database are determined through a quick fetch()/$ajax() to the backend, then cached in localStorage so future endpoint calls are minimized for the client's session.

Upon clicking any tile, the client is directed to the classes.html page.

classes.html

classes.html is the page that will load all the classes associated with the selected major who also have existing back exams. Similarly to how index.html loads all majors with existing back exams, classes.html recycles that same logic and implements it just a tier lower. The logic for filtering db results by major can be handled in javascript pretty easily, and if the localStorage variable is still accessible after changing files, then another endpoint call is avoided.

Upon clicking one of these tiles, the user is directed to exams.html, the page with the good stuff.

exams.html

Like index.html and classes.html, exams.html consists of interactive tiles that users can click. These tiles now represent actual back exams that exist in the database. The tiles can be given various unique identification criteria, such as teacher, year, semester, has answers, etc.

Upon clicking a tile, you are redirected to exam.html

exam.html

This page is a simple html rendering of the pdf, fit to your computer's screen. Extendable features to consider are implementing the same pdf viewer the browser uses, and a download pdf button.

upload.html

This page is reached via index.html, and is the place where people can upload back exams they find. This is a rather delicate feature, because it introduced a whole world of security risks. Further research needs to be done to support the risk and effort incurred by this feature.

Important Tools Needed:

- A database to hold PDF's in
- A way to serve a downloadable file
- A backend with an endpoint to load PDFs
- JQuery to handle parsing logic

Significant Changes

We have thoughtfully considered our website in the shoes of the user, and modified our IA to simplify the user experience as much as possible. Our new IA has much more depth.

We also have identified the resources we need and challenges to cross. Specifically, a lot of this website depends on the database and local caching strategies to ensure speedy load times. Additionally, we will need to facilitate requests to a backend to help request this information. We will need a strong understanding of SQL and PHP for this to work.

Another significant change we are considering is designing the security layout for a feature where a user can upload a PDF. This has many concerning strings attached, but is an interesting challenge that should be considered as the project moves forward.