

ACMAS: The Automatic Course Material Archiving System

Jacob Weber, Justin Banzon, Cherry Bommu, Jing Cui, Noah Cussatti, Matthew Glanz, Jamarri Green, Enver Kuli-Zade, Ruikang Lin, Konain Qureshi



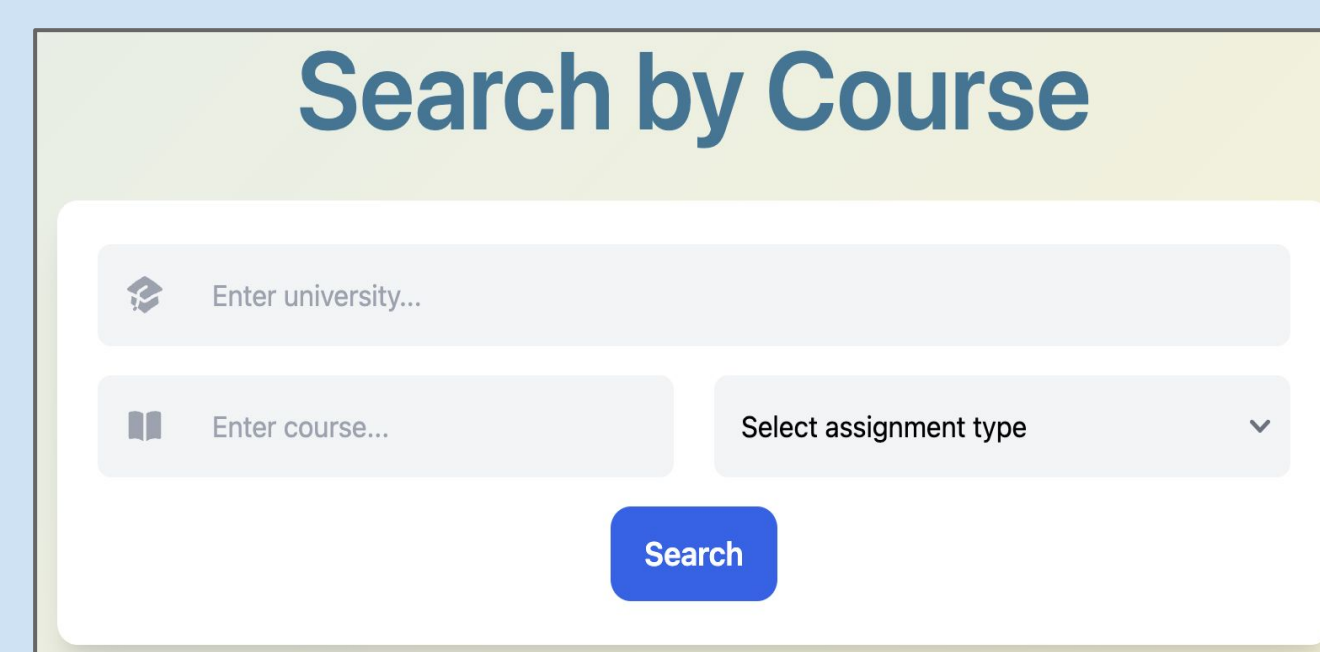
User Interface

Our accomplishments this semester were:

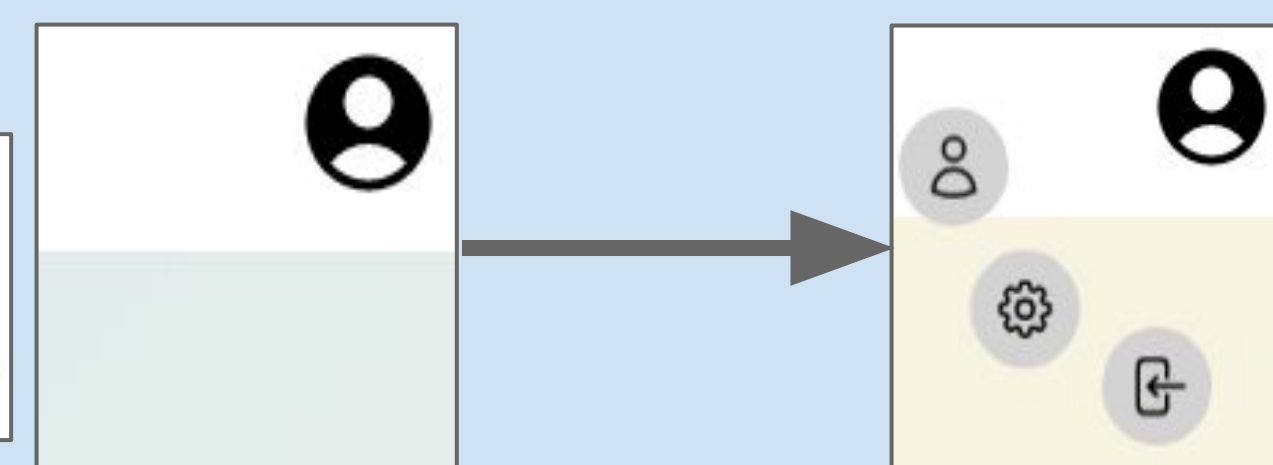
- Replaced *Bootstrap* with *tailwindcss*
- Redesigned the structure and look of the website using *tailwindcss*



- Implemented component based design in Django:
 - Object-oriented
 - Nested components
 - Minimizes grunt work
 - Easy to use
 - Documented for future development



```
# (% component "icon" icon="" icon_type="" icon_class="" %)  
# icon: "information-circle", ... (same naming as the website: https://heroicons.com/)  
# icon_type: "outline", "solid", "mini"  
# icon_class: defaults with "w-5 h-5", you can add other classes for tailwind "..."  
  
# NOTE need to load following line at the top of component's html to do nested components  
# (% load component_tags %)
```

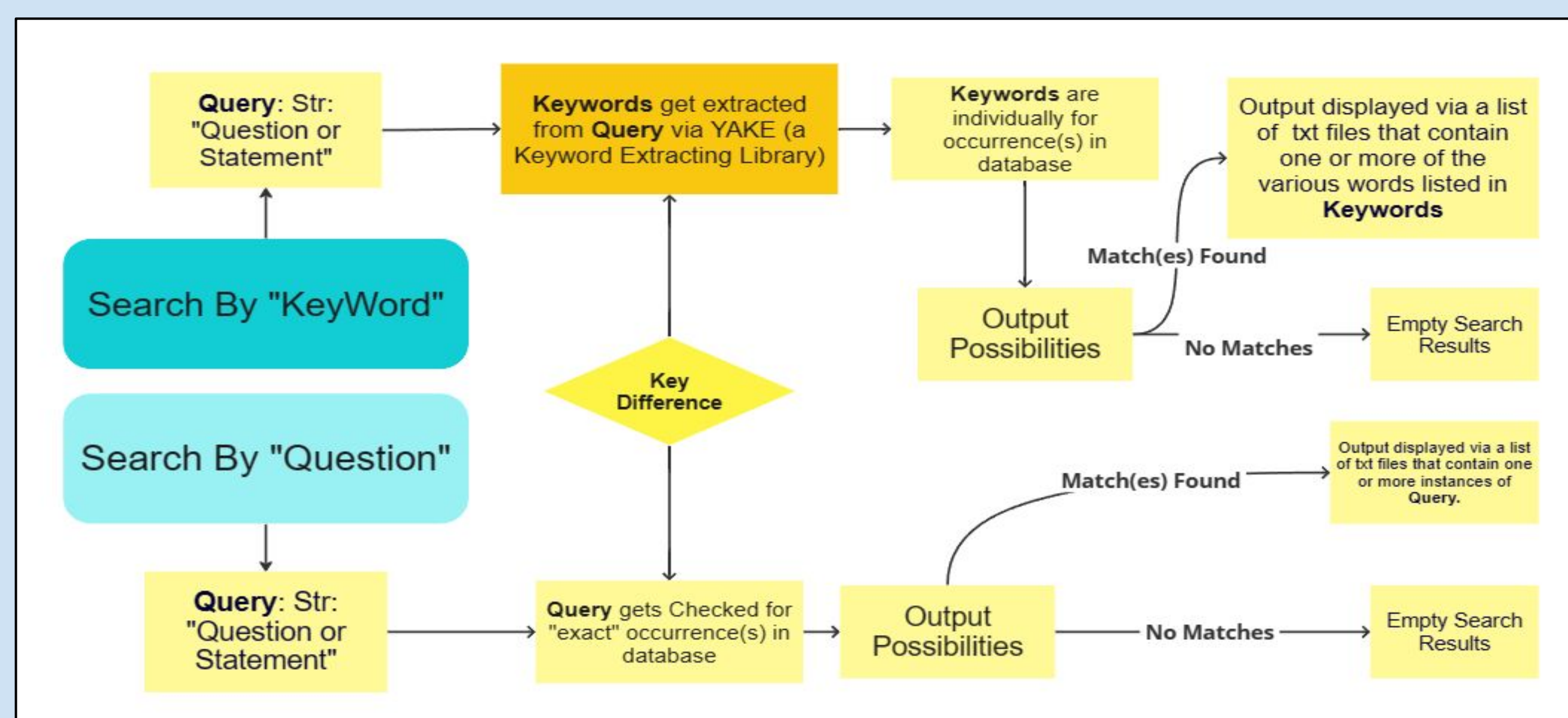


Search engine

Our Goals/Work This Semester:

- Implementing a new search algorithm that utilizes extracted keywords to match queries with the database
- Working with APO to gain access to their back work database to upload onto our backend
 - Which now entails the need for us to add zip file upload functionality to ACMAS
- Automate new content updates to the database with user uploads through the UI

Algorithmic Overview

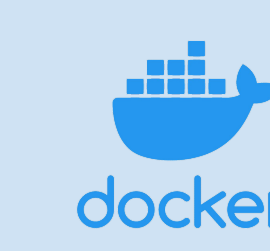
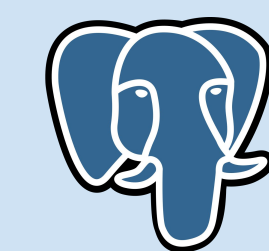
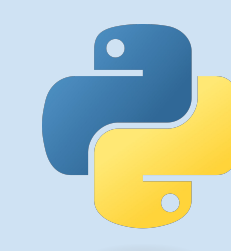


"A free to use back-work/back-test database website for students and alumni to upload and view coursework and tests from any school."



Semester Goals

- User accounts page and UI/UX functionality
- Dark mode UI and standardize colors across the website
- Implement OCR that works with all file types and converts files to text questions
- Create user account functionality
 - Allow users to edit and delete their uploaded files
 - Allow users to change/add new "tags" to said files
 - Enable easier content moderation by moderators/admin users
 - Add "verified uploader" functionality which won't require manual review by moderators
 - An example for this would be APO uploading from their existing database
- Improve the content search engine
- Enable HTTPS support, securing our website (currently only using HTTP)
- Investigate and possibly transition to a different web hosting solution if the memory requirements of the search engine prove to be too intensive



Implementation

Completely Dockerized

Application / Nginx reverse proxy / Postgres DB

Server Specifications

AWS Lightsail: 1 vCPU, 1 GB RAM, 40 GB SSD
Specs subject to change with OCR and search engine needs!

Team Structure

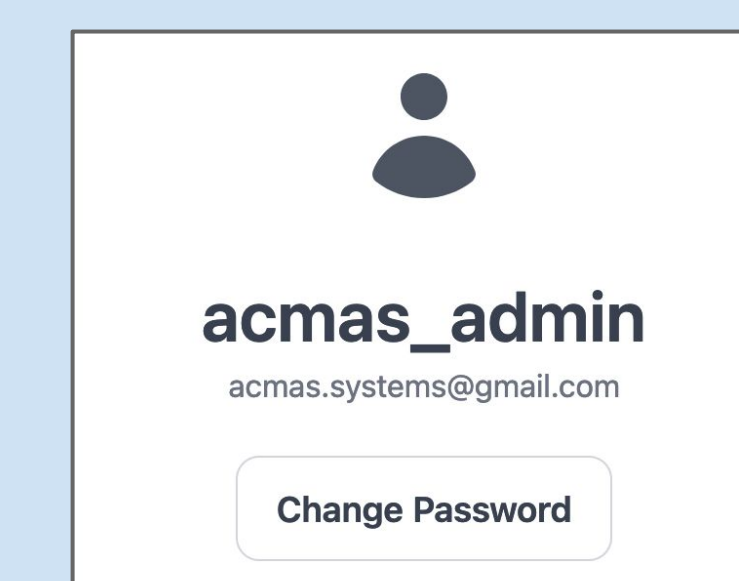
Major features are handles by sub-teams:

**OCR, User Interface, Search Engine,
User Accounts, Server/Project
Management**

User Accounts & Moderation

This semester our team focused on three primary goals:

- Implement a user account system using Django authentication into the existing ACMAS framework
- Create a moderation system for uploaded files so that they can be approved or rejected by moderators prior to upload to the database
- Create user groups so that we can assign certain administrative permissions automatically to different groups



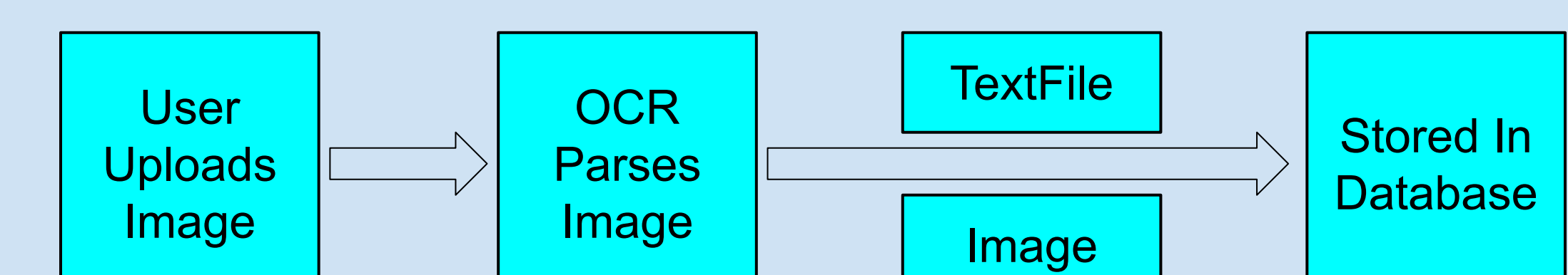
All of these goals were accomplished and implemented into ACMAS with a functional registration, login/logout system, administration portal, automated group creation script, as well as custom pages based on authentication state.

OCR

OCR is a character recognition model used to parse text from imagery inputs. In ACMAS, OCR is used to parse questions and answers from imagery uploads (PDF, PNG, JPG, etc.)

Our goal this semester:

- Have a functional OCR parser that identifies and extracts questions from uploads and store them in a text file
 - Implementation 1: Use external python modules like Pytesseract to extract text from images with a near 99% accuracy
 - Implementation 2: Utilize a KNN algorithm to train a model to classify characters from a given image (MNIST for training data)
- Create a new pipeline for storing these text files in the database so it can be utilized for search engine



Join ACMAS!

