Project: Digital library search engine with a Chatbot

Student name: Maaz Hasnain Khan

Student UIN: 01244465

All sections are mandatory. Please do not change the format of this template.

1. Overview

The website we are working on is a Digital Library search engine with a Chatbot. We are using Create React App (CRA) Framework and Node.js API. For this milestone we are implementing the document indexing and search function with the features mentioned below:

2. Milestone Accomplishments

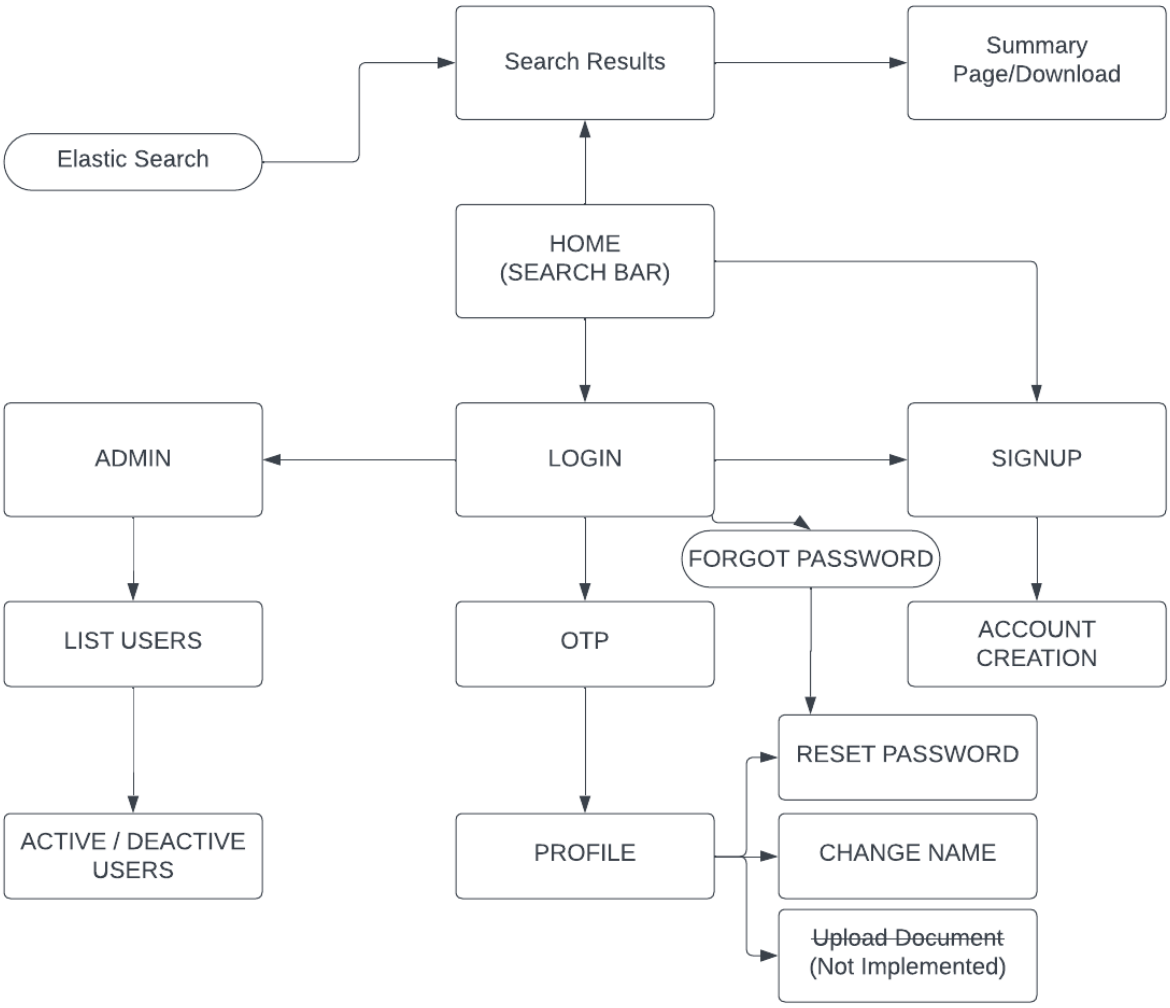
List ALL specifications of this milestone and specify whether certain specifications are fulfilled or not.

Table 1: Status of milestone specifications.

Fulfilled	Feature#	Specification
Yes	1	The website should index all documents in the ETD500 repository.
Yes	2	Users can query the search engine and obtain results without logging in.
Yes	3	The search engine accepts a text query in the search box and return results on the search engine result page (SERP).
Yes	4	Each item in SERP should link to a summary page containing metadata for a document and a link to download the PDF. The metadata of an ETD include title, author, year, university, program, degree, advisor, and abstract.
Yes	5	The SERP should display the number of returned items on top of search results.
Yes	6	The SERP should show both the queried keyword and the keywords after automatic spell correction.
Yes	7	The SERP should show the search box so users can continue searching.
Yes	8	The SERP should be paginated. The user should navigate to the next page, the previous page, a button that flips to the last page and a button that flips to the next page.
Yes	9	If the number of returned items cannot be held within 2 pages, the SERP should display at least 2-page numbers, in addition to the buttons above. Otherwise, the SERP should only display two-page numbers and the pagination buttons should be disabled. [For 518 students only] For example, if there is only one page, the SERP should display (gray color means the button is disabled, but you can use other colors you like)
Yes	10	The query terms should be highlighted in the search results. [For 518 students only]
Yes	11	Use Wikifier to obtain Wikipedia terms appearing in ETD abstracts and store the terms and the Wikipedia URLs in the JSON object above.
No	12	A regular user should be able to insert a new document after logging in, including key metadata (title, author, year, advisor, university, degree, program) and upload the PDF. Search the title of the new paper and it should show up in the top search results.

3. Architecture

The architecture follows a standard web application model with a client-server interaction pattern, where Create React App (CRA) serves as the client (frontend) and Node.js as the server (backend). For milestone 1, the database stores and retrieves user data, and various pages and components provide different functionalities as described milestone specifications. For milestone 2, the Metadata was parsed and dumped in the database. The ETD Metadata is indexed in Elastic Search and pulled into the search page to be displayed. Also, each search result leads to a summary page containing the information of the particular document along with a PDF downloading option. Other specification mentioned in the milestone 2 specification were also implemented like, search term highlighting, pagination, spell correction, total search results and wikifier. The uploading document specification was not completed and will be shown in milestone 3.



4. Data


```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 2,
      "relation" : "eq"
    },
    "max_score" : 5.6627026,
    "hits" : [
      {
        "_index" : "edt-index",
        "_type" : "_doc",
        "_id" : "xh0ky4sBKCRR3dV9UMmM",
        "_score" : 5.6627026,
        "_ignored" : [
          "wikifier_terms.keyword",
          "abstract.keyword"
        ],
        "_source" : {
          "edtid" : 344,
          "title" : "improving investment performance and the role of fixed-income securities",
          "author" : "fischer, richard gatts",
          "year" : "1973",
          "university" : "harvard university",
          "program" : "finance",
          "degree" : "d.b.a.",
          "advisor" : "",
```

"abstract" : """['Abstract The objective of the study was to determine the normative role of fixed-income securities in contributing to improved investment performance in mixed]""",

"pdf" : "344.pdf",

"wikifier_terms" : """[{"title":"Fixed income","url":"http://en.wikipedia.org/wiki/Fixed_income","lang":"en","pageRank":0.005528781017943669,"secLang":"en","secTitle"

5. Elastic Search Design

The 'edtid' column serves as the primary key, ensuring uniqueness for each record. Fields like title, author, year, university, program, degree, advisor, abstract, pdf, and wikifier_terms hold information such as the thesis title, author's name, production year, university, academic program, degree, advisor, abstract, PDF file location, and wikifier terms, respectively.

This design follows a relational database model, where data is organized into rows and columns. The primary key facilitates record uniqueness and can be used for establishing relationships with other tables. Additionally, the use of appropriate data types, such as LONGTEXT for extensive text and VARCHAR for variable character strings, contributes to storage efficiency. The overall structure adheres to normalization principles, promoting effective data manipulation and retrieval within a relational database framework.

Field	Type	Key	Example
edtid	INT	Primary Key	1
title	LONGTEXT		Ecology Side Effect Paper
author	VARCHAR (100)		Saleem Khan
year	VARCHAR (5)		1990
university	VARCHAR (255)		ODU
program	VARCHAR (255)		CS
degree	VARCHAR (255)		PhD
advisor	VARCHAR (255)		Jian Wu
abstract	LONGTEXT		Ecology affects us in….
pdf	VARCHAR (255)		1.pdf
wikifier_terms	LONGTEXT		title:aa,url:Wikipedia.com

6. Implementation

Search box and search button:

The search box and search button are at the landing page of my website. The search bar allows users to input specific queries, and upon initiating the search, the system leverages the capabilities of Elastic Search to retrieve relevant information quickly and effectively. Even after the search is completed the search bar is still there if the user wants to search something else. The code written for the elastic search in `digital-library-api/elastic-search.ts` helped in indexing the data and the query written to implement the search functionality. Each item in the search results is associated with individual links. Clicking on these links redirects users to detailed summary pages for the respective search results. The code for this page is in `digital-library-chatbot/src/pages/search-bar/search-bar.tsx`

Summary Page:

Upon clicking the “More” option on the search page, you will be directed to the summary page of the search result. It contains specific information about the document and the document can also be downloaded. The code for this page is in `digital-library-chatbot/src/pages/detail/detail.tsx`

Returned Items:

The returned items are shown below the search bar after a query is searched. The code for this feature is in `digital-library-chatbot/src/pages/search-bar/search-bar.tsx`

Spell Correction:

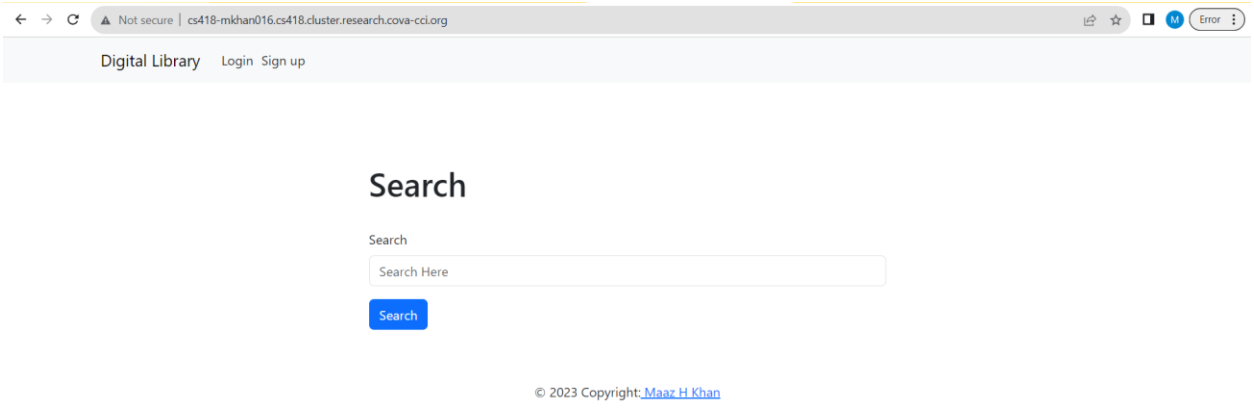
If a wrong or misspelled word is searched, the user will be given suggestions regarding the correct word. The code for this feature is in `digital-library-chatbot/src/pages/search-bar/search-bar.tsx`

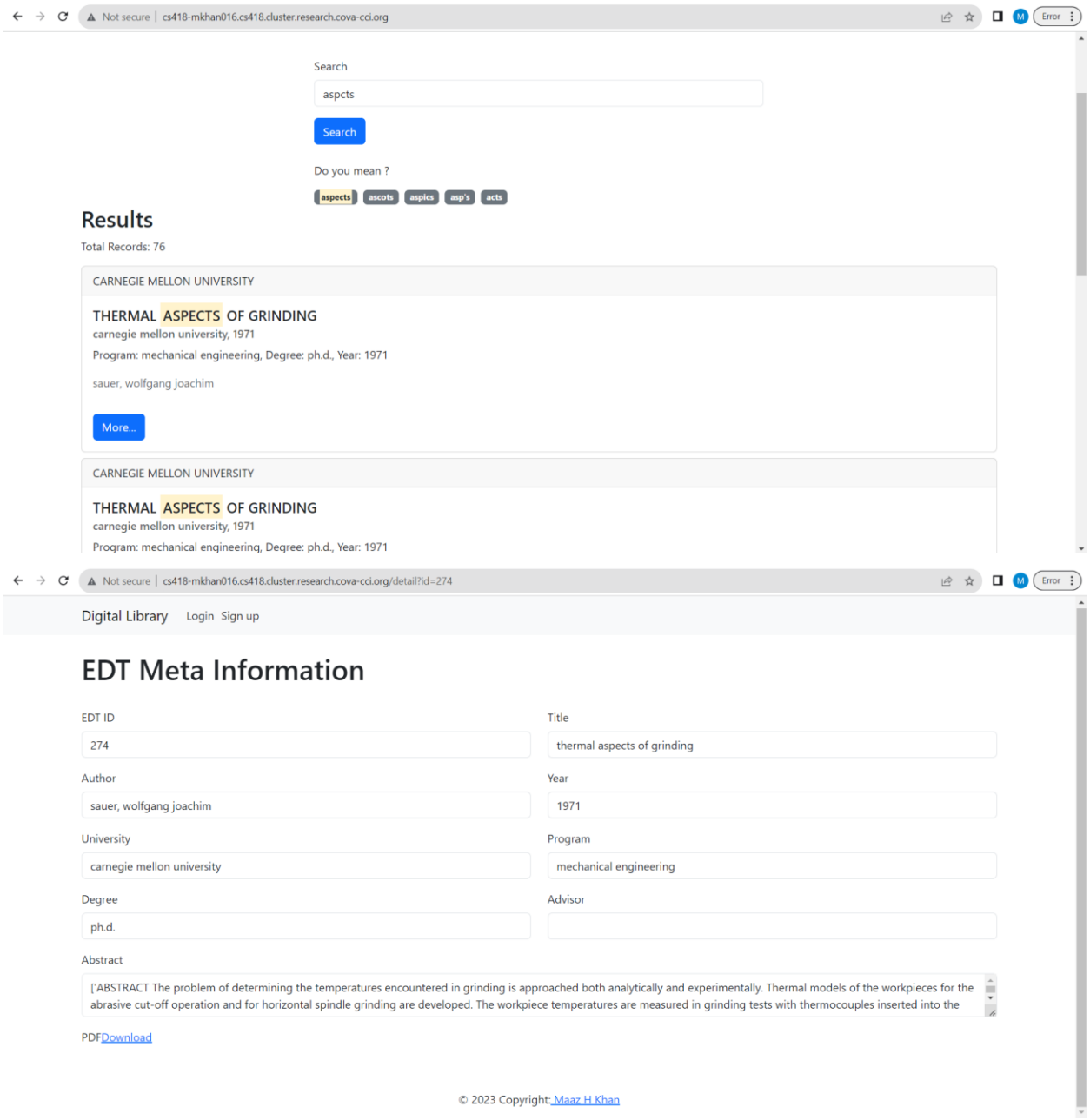
Pagination:

The search page should be paginated. The user can navigate to the next page, the previous page and last page. If the number of returned items cannot be held within 2 pages, the search page will display at least 2-page numbers, in addition to the buttons above. Otherwise, the search page will only display two-page numbers and the pagination buttons should be disabled. The code for this feature is in `digital-library-chatbot/src/pages/search-bar/search-bar.tsx`

Highlighted Search Words:

Whenever a query is searched, the searched word will be highlighted in the search page. The code for this feature is in `digital-library-chatbot/src/pages/search-bar/search-bar.tsx`




Not secure | cs418-mkhan016.cs418.cluster.research.cova-cci.org
Error
Search
aspcts
Search
Do you mean ?
aspects
ascots
aspics
asp's
acts
Results
Total Records: 76
CARNEGIE MELLON UNIVERSITY
THERMAL ASPECTS OF GRINDING
carnegie mellon university, 1971
Program: mechanical engineering, Degree: ph.d., Year: 1971
sauer, wolfgang joachim
More...
CARNEGIE MELLON UNIVERSITY
THERMAL ASPECTS OF GRINDING
carnegie mellon university, 1971
Program: mechanical engineering, Degree: ph.d., Year: 1971

Not secure | cs418-mkhan016.cs418.cluster.research.cova-cci.org/detail?id=274
Error
Digital Library
Login
Sign up
EDT Meta Information
EDT ID
274
Title
thermal aspects of grinding
Author
sauer, wolfgang joachim
Year
1971
University
carnegie mellon university
Program
mechanical engineering
Degree
ph.d.
Advisor
Abstract
['ABSTRACT The problem of determining the temperatures encountered in grinding is approached both analytically and experimentally. Thermal models of the workpieces for the abrasive cut-off operation and for horizontal spindle grinding are developed. The workpiece temperatures are measured in grinding tests with thermocouples inserted into the
PDFDownload
© 2023 Copyright: Maaz H Khan