Final Report
CS 5604: Information Storage and Retrieval

User Interface Team:

Mary Kate McGranahan, Sunday Ubur

Subject Matter Expert:

Sara Ahmadi

December 17, 2023

Instructed by Professor Edward A. Fox

Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

Abstract

The Electronic Thesis and Dissertation (ETD) project concept has evolved for years now and has been undergoing improvements across key themes. The Information Retrieval course at Virginia Tech led by Dr. Edward Fox has been at the forefront of implementing a robust ETD system. In this report, we document our contribution to the system as Information Retrieval course students for the Fall 2023 academic year. The overall project tasks were divided among 6 teams. Our team, Team 6, was in charge of the User Interface. The state of the system at the beginning of the semester was a fragmented interface with several independent implementations done by students who took this course in the past. Our main task was to merge these into one robust and consistent user interface. We were also tasked with upgrading the loosely implemented authentication system so that users can securely sign up for the system, with User Interface (UI) pages supporting new user roles of curator and experimenter. It was our responsibility to meet these requirements in collaboration with other teams, especially with Team 5 who provided the Application Programming Interfaces (APIs). We accomplished the main goals of one consistent user interface that supports all of the functionality of the ETD system including new user roles, and a secure system for signing up and logging in. This report details these accomplishments, as well as how to use the frontend of the ETD system. This course has taught us an important concept of hands-on practice beyond reliance on the theory of Information Retrieval, and as a result we have improved our technical skills on frontend implementation using React and dependencies, how to implement a reliable user authentication system, and more importantly how to collaborate on an industry level project as a team using the Agile software development framework, and CI-CD development with GitHub. Finally, we tested the reliability of our frontend production version using an NGINX server.

Contents

List of Figures

List of Tables

Chapter 1

Overview

1.1 Requirements

The requirements of this project for the different versions are as follows:

1. Version 1: Make the system from Fall 2022 fully operational, with pages to support the personas of end-user doing research, experimenter, curator, and administrator.
2. Version 2: Create the UI with a GUI interface. Add UI support for all new version 2 features such as topic modeling and object detection. Make sure only authenticated users can access the system.
3. Version 3: Add a chat feature to the UI: We have not implemented chat. It is reserved for future work.

1.2 Accomplishments

The goal of our work was to create a single consistent User Interface to support all of the personas a user might have, with enhanced user authentication, and new search, recommendation, curation, and experimentation features.

Our main focus in the initial stages of work was defining a design for the system. This involved creating a new user flow design to show how the user would navigate through the new authentication system and the improved User Interface. We then created wireframe designs to communicate changes to the interface with other teams and gather their feedback.

In our second stage, we worked with Team 2 to update the API calls for search so the search interface was fully functional. We then worked with Team 5 to plan the API calls and architecture for the rest of the user interface updates, with a focus on the login architecture. After discussions with Team 5, we diagrammed all the system interactions. We also began work on some of the brand new pages, such as the administrator interface. We went through a lengthy process of designing the curator and experimenter pages. After exchanging wireframes with representatives of the other teams, we devised a list of options that curators and experimenters should have, and updated the pages for these functions accordingly.

Our final task was implementing authentication and user profiles. In collaboration with Team 5, we made it possible for users to sign up and login to the system. Only users with a vt.edu or odu.edu email can sign up, and users must be approved after signup, before they can access the system. We controlled user access so that pages on the site will only load for a logged in user. There is role based access to certain pages such as the administrator, curator, and experimenter pages. A user with administrator access can see an administrator page where they have the ability to alter the roles of other users. In addition to being more secure, the new system also enhances the user experience, and aids the recommendation of ETDs, by allowing selection of preferred ProQuest categories and Topic Modeling topics. These categories and topics were determined in collaboration with Teams 2, 3, and 4 to work with their recommendation, topic modeling, and classification systems.

This paper will outline the requirements we are aiming to meet in this project in Chapter 2. In Chapter 3 we discuss the literature used to inform our development decisions and user authentication architecture. Chapter 4 will discuss our design process; this is where the wireframe designs and user flow we created for our first update can be found along with the outline of the system architecture. In Chapter 5 we discuss how this project was implemented. Chapter 6 provides a user manual to guide prospective users and developers in navigating the system. Chapter 7 contains the Developer Manual and Chapter 8 highlights our future plans for the project.

1.3 Challenges

An initial challenge was dealing with bugs in the existing user interface code. Last year's codebase could not be run without resolving a number of issues.

Once we resolved these issues, our biggest challenge was to get user authentication working. Without working user authentication, it was difficult to access and test pages. This has required a lot of communication with Team 5. It required planning and coordination on the architecture for authentication. Team 5 then had to test the API endpoint to authenticate users. Once basic signup was working, the structure of API calls had to be updated to include sending the ProQuest categories and topic modeling topics to the database. Finally, the system had to been enhanced with further security measures such as limiting access to certain emails and requiring approval.

Another challenge we faced was testing our frontend on an NGINX server locally. We are able to deploy the production version on NGINX, but had challenges communicating with the API from it.

A final persistent challenge was communication. As the User Interface team, much of our work was reliant on what functionality other teams had been able to implement and what parts of their systems required user input. This left us stalled at times waiting on information from other teams. It created an ongoing process of communicating with teams as they accomplished their tasks and gained a better understanding of what their interfaces could look like.

1.4 How We Overcame These Challenges

We dealt with our challenges fixing bugs in the older codebase by becoming familiar with the languages and dependencies that were used. We also consulted with some subject matter experts who were familiar with the old system.

The issues with authentication were solved through communication with Team 5 to define a clear system for sending signup data to the database and login data as well as returning user access tokens for a logged in user to access the various pages.

Our NGINX problems were solved by deploying code to the development version of the system on endeavour as opposed to trying to run a local NGINX server.

1.5 Future Work Needed

There were a number of things we were unable to accomplish within the constraints of this course that we believe would be beneficial work for future researchers, namely:

1. Testing Experiment and Curation Features: The new experimenter and curator features on the user interface have not undergone any testing to ensure they align with the APIs for those systems. These features should be tested and refined.
2. User Testing for All Pages: The ETD system needs user feedback from new users, people who are not in the course and do not already understand the system.
3. Topic Modeling Interface: The Topic Modeling team was able to categorize the ETDs in the system according to a collection of topics. A specific topic modeling centered search should be added to the ETD interface to utilize their work.
4. Adding Support for Administrators Analyzing Trends: The user interface should add proper logging of user actions and a page for Administrators to be able to track trends in user actions.
5. Chat Feature: In the future the interface should have an AI chat feature to assist with locating documents and help features.

Chapter 2

Literature Review

2.1 Previous Work on the ETD System

We utilized materials from the Fall 2022 Teams to assist with our initial design of the User Interface. We also utilized the codebase from Fall 2022 as our base template for this work. Although there was no User Interface team in the Fall of 2022, Team 2 created a user interface to run ElasticSearch that supported a general user. Therefore, the main report we studied was from Team 2 [1]. We also planned to use Fall 2022 Team 3's work to inform our creation of a topic modeling page, as well as the topic modeling toolkit developed by a Spring 2023 CS4624 team [2].

2.2 Frontend Development with React.js

Frontend development typically relies on the combination of HTML, CSS, and JavaScript, closely tied to the Document Object Model (DOM) to create vanilla JavaScript functions. This approach often necessitates writing lengthy and convoluted code, leading to difficulties in managing and maintaining the application. In response to these challenges, React.js [3], an open-source JavaScript library developed by Facebook (now Meta), has emerged as a powerful tool for building frontend applications. React.js offers the capability to construct high-performance frontend applications by breaking them down into manageable, reusable components. This approach aligns with the concept of micro-frontends, single-page applications (SPAs), and microservices, which were introduced to mitigate the increasing complexity of web applications developed using the traditional DOM approach, and reduce coupling [4]. React.js streamlines the process by handling the conversion of components into a virtual DOM behind the scenes. This not only ensures reusability and organization but also enhances application speed. Research indicates that React.js, alongside other widely-used frontend libraries like Angular and Vue, is a dominant force in the industry [5]. This efficiency, especially the reliance on the virtual DOM, is a key reason for adopting React.js in demanding applications such as the Electronic Thesis and Dissertation (ETD) project [6], and it is also favored in colossal web applications like Facebook, Instagram, and WhatsApp. Moreover, React.js proves invaluable in developing trading applications that must process vast amounts of data and accommodate rapid and frequent dynamic changes [7].

2.3 User Authentication

User authentication and authorization are integral components of nearly every frontend application, providing the foundation for secure application development. Web applications are inherently accessible to anyone with internet access, necessitating robust authentication and authorization mechanisms, particularly in single-page applications (SPAs) like React, where implementing these features can be challenging [8]. Authentication holds a pivotal role in ensuring the security and controlled access of digital environments. It encompasses the process of verifying a user's identity and trustworthiness, ultimately granting access to the system's resources. This verification process may involve users providing evidence of their identity, such as passwords, personal identification numbers (PINs), or biometric data [9]. Notably, authentication through visual means, such as image recognition [10], can offer a more convenient and expedited approach in this regard.

2.4 Protocols to Authenticate Users in React

There are numerous authentication protocols and methods for user verification in React applications. One widely employed protocol is the 802.1X protocol, extensively utilized in LAN environments [11]. This protocol entails the user sending an EAP-Response/Identify message to the authenticator, including the user's username. Alternatively, there is the location-based authentication protocol, which leverages user location data to grant access [12]. However, it's essential to emphasize that location-based access falls within the purview of the authorization service, distinct from the authentication process. Aside from these protocols, React applications offer a range of other authentication methods. Password authentication stands out as a secure option, even in scenarios where an intruder gains access to the system's data [13], which is the main reason we adopt it for this application. Another avenue is biometric authentication, hinging on biometric attributes like facial profiles or keystroke dynamics for user identity verification [14, 15]. While these methods offer heightened security, they may necessitate additional hardware or software investments [16]. It's noteworthy that ongoing research and development in user authentication are shaping the landscape. This includes investigations into behavioral biometrics for continuous user authentication [17]. Furthermore, research delves into password persistence and the quest for alternative authentication mechanisms [18]. Artificial intelligence and machine learning techniques are also under exploration to enhance authentication accuracy and reliability [14]. In React applications, the array of available authentication options encompasses the 802.1X protocol, location-based authentication, password authentication, and biometric authentication. Each method boasts its own merits and considerations, with the selection dependent on the specific application requirements and constraints. At Virginia Tech, two-factor (2-FA) authentication is a common practice for user authentication [19]. However, it's important to note that 2-FA might not be the ideal choice for the Information Retrieval System, especially considering its broader user base beyond the Virginia Tech community.

2.4.1 OAuth

OAuth (Open Authorization) is a widely embraced open standard protocol designed to enable users to grant third-party applications access to their resources without disclosing their credentials. This protocol serves as a secure and standardized means for users to authorize access to their data stored across various websites and services. OAuth finds extensive utility in diverse applications, including social media platforms, online banking, cloud storage services, and Information Retrieval systems. The evolution of the OAuth protocol has produced two major versions: OAuth 1.0 and OAuth 2.0. OAuth 1.0 made its debut in 2010, while OAuth 2.0 was introduced in 2012. Both versions come with their own set of specifications and have been widely implemented by various service providers. OAuth 1.0 adopts a three-legged authorization flow, involving the user, the client application, and the service provider. The flow commences with the client application seeking authorization from the user. Subsequently, the user is redirected to the service provider's authorization endpoint, where authentication occurs, and permission is granted to the client application. The service provider generates a temporary access token and secret, which the client application utilizes to access the user's resources, ensuring request integrity through the use of signatures [20]. In contrast, OAuth 2.0 simplifies the authorization process through the introduction of access tokens and a four-legged authorization flow. This flow encompasses the user, the client application, the authorization server, and the resource server. It kicks off with the client application's request for user authorization, leading the user to the authorization server. After user authentication and consent, the client application acquires an authorization code, which is subsequently exchanged for an access token from the authorization server. This access token empowers the client application to access the user's stored resources on the resource server [21]. OAuth 2.0 brings several advantages over OAuth 1.0, including enhanced security, scalability, and flexibility. It also offers support for various authorization grants, such as authorization code, implicit, client credentials, and resource owner password credentials. These grants accommodate diverse access and authentication needs, tailored to specific application requirements [22].

2.4.2 JWT

JSON Web Tokens (JWTs) represent a widely adopted approach for securely conveying JSON object-based information between frontend and backend [23]. Their primary use is in web applications, particularly for authentication and authorization purposes. This section serves to offer insights into JWTs, encompassing their composition, operational principles, and security aspects. A JWT comprises three integral components – the header, the payload, and the signature – as outlined by [24]. These segments are base64url encoded and subsequently concatenated with periods to form the complete token. The header carries details concerning the token type and the employed signing algorithm. Meanwhile, the payload incorporates claims, which encompass statements about an entity and supplementary metadata. The signature is established by signing the encoded header and payload, utilizing either a secret key or a private key. The utilization of JWTs hinges on a combined involvement of the issuer, the client, and the resource server. The issuer holds the responsibility for JWT creation and signing. Subsequently, the client receives the JWT and integrates it into subsequent requests directed at the resource server. The resource server takes on the role of verifying the signature and validating the claims within the JWT to permit access to safeguarded resources [25].

2.4.3 Conclusion

React.js is a dominant force in frontend development, offering efficiency and organization, while user authentication remains a crucial aspect in securing web applications, with various methods, including visual authentication, playing a significant role in ensuring user identity and trustworthiness using OAuth and JWT. OAuth is a widely adopted protocol with major versions, OAuth 1.0 and OAuth 2.0. OAuth 2.0, with its simplified four-legged flow and use of access tokens, offers improved security, scalability, and flexibility, making it the preferred choice for many service providers. Similarly, JSON Web Tokens (JWTs) provide a versatile and secure means of exchanging information between parties. Understanding their structure, operational principles, and security considerations is pivotal for establishing robust authentication and authorization mechanisms within web applications.

Chapter 3

Design

Our design process consisted of three main parts: user flow diagrams for the entire system; UML diagrams to show interactions between the frontend, the APIs, and the database; and wireframe designs created to get feedback from other teams on our proposed screen models. All of these were vital to communicating our ideas with other teams, and ensuring a smooth transition to the updated system.

3.1 User Flow Diagrams

The previous system had an in depth user flow diagram showcasing all the paths a user could take to navigate the system. The main focus of designing a new user flow was the authentication process. The old system allowed users to search and access the ETD without being logged in, which was apparent in their user flow diagram. We designed a new login flow that clearly blocks any access to the features of the web application until the user is logged in. By logging in, the user gets an authentication token that will be passed into all their future requests. If a user does not login, they will be unable to leave the login page.

Figure 3.1 represents the user flow from last year's system, in which the user can bypass sign up and still access the recommendation and search pages. [1]

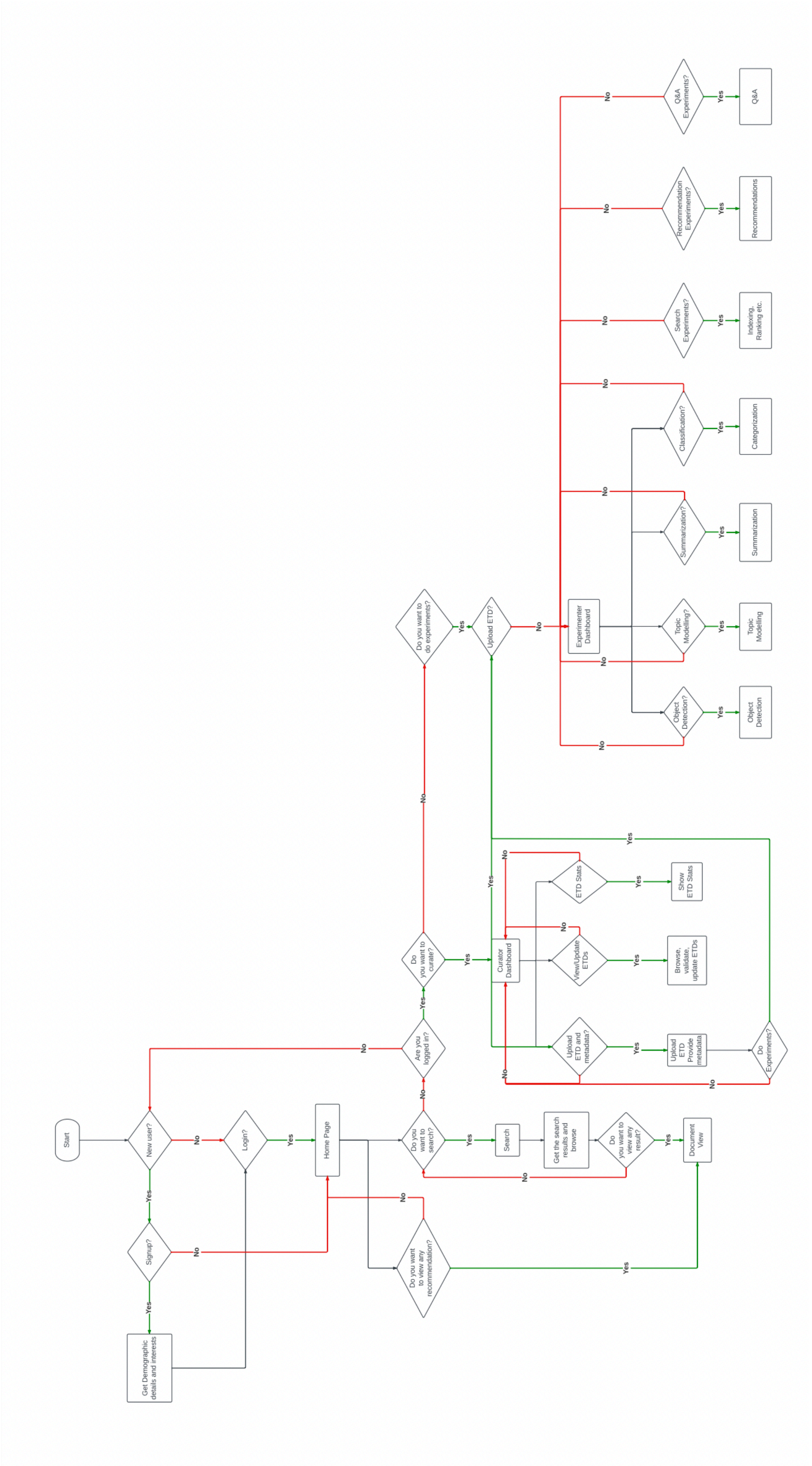


Figure 3.1: Old frontend flow diagram

Figure 3.2 represents the new user flow, showing our modification in which the user cannot access the home page without being fully logged in.

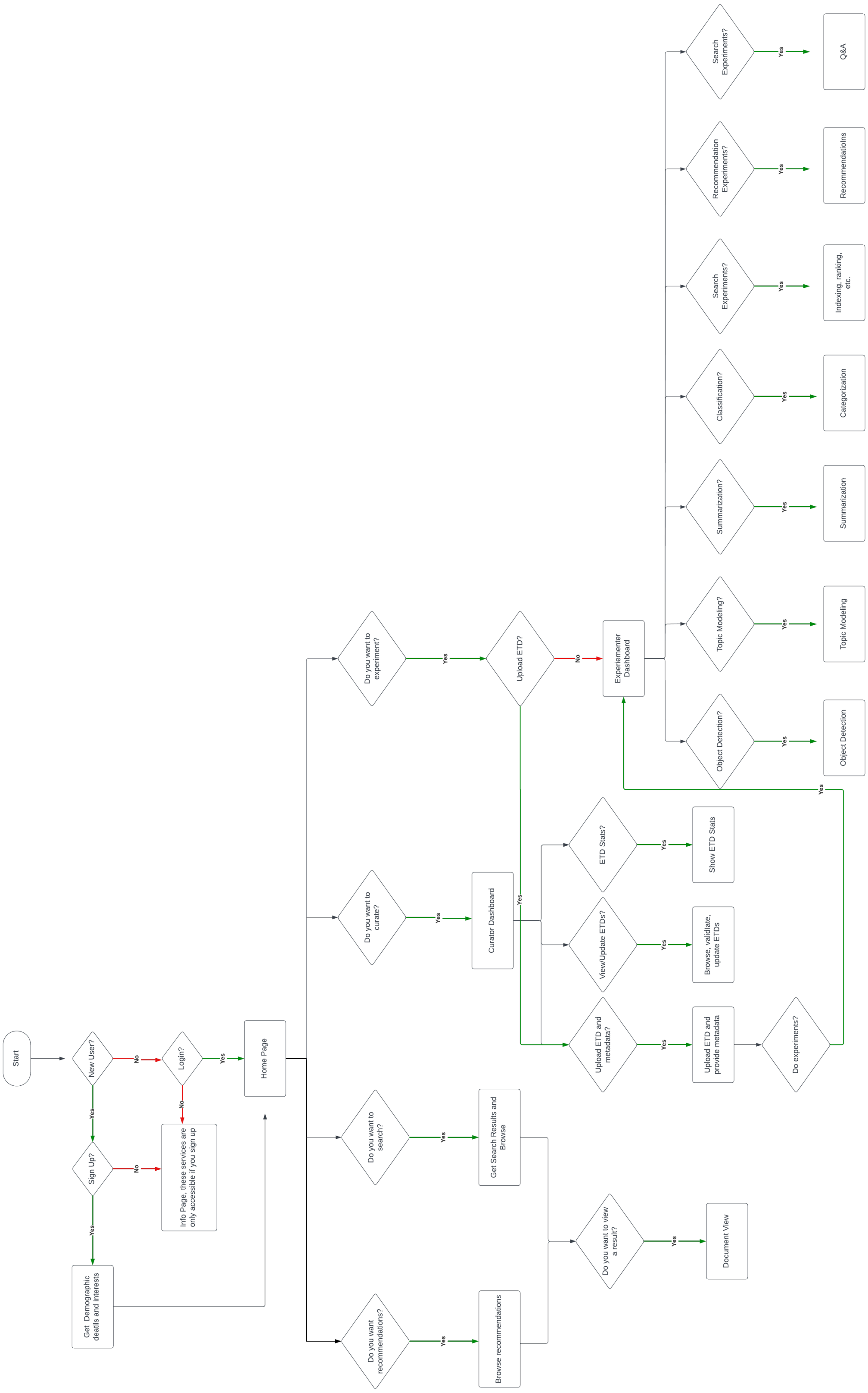


Figure 3.2: New frontend flow diagram

3.2 Wireframe Designs

Wireframes were designed to showcase edits to pages that already existed in the old interface, as well as new pages to be added to the interface. The wireframes were useful in ensuring a unified user interface that all teams were on board with. These wireframes went through many iterations and were our main tool for gathering feedback from other teams on the frontend functionality. We will go through the wireframes for different pages one by one, making note of the fact that there were previous versions of these wireframes, and the pages may have been adjusted in the final version of the system.

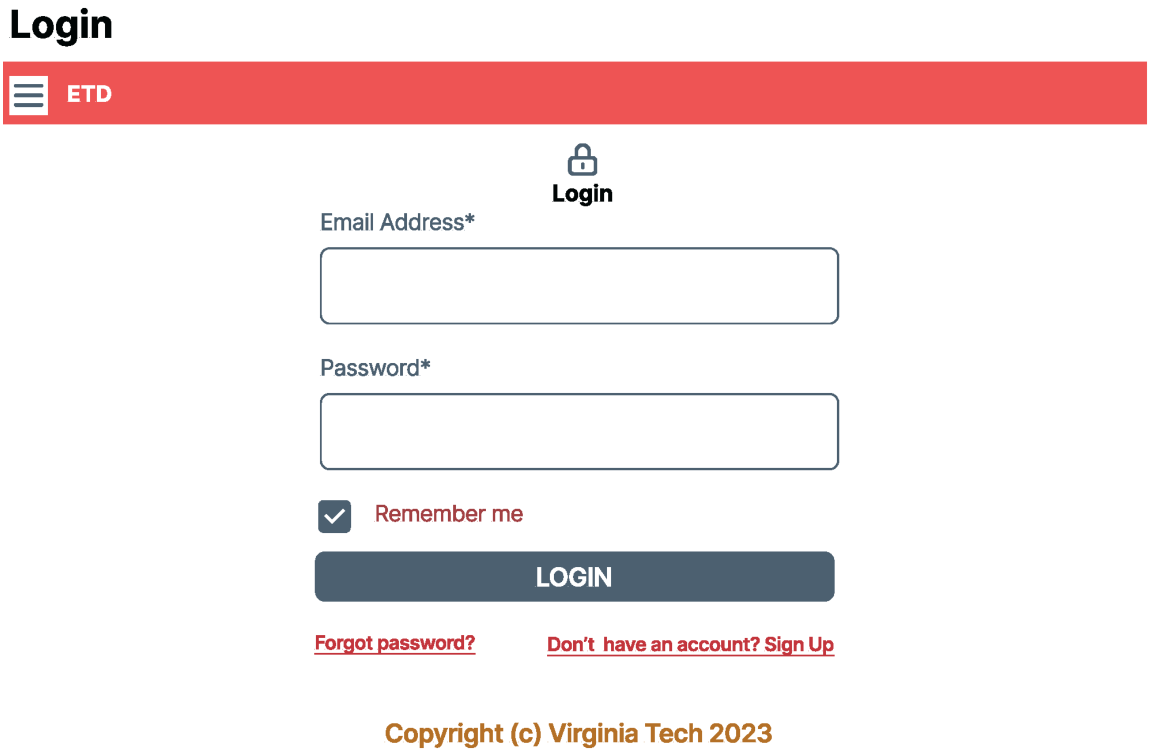


Figure 3.3: Login

The login page design shown in Fig. 3.3 has remained the same between the new and old versions. Changes to login functionality occurred on the backend.

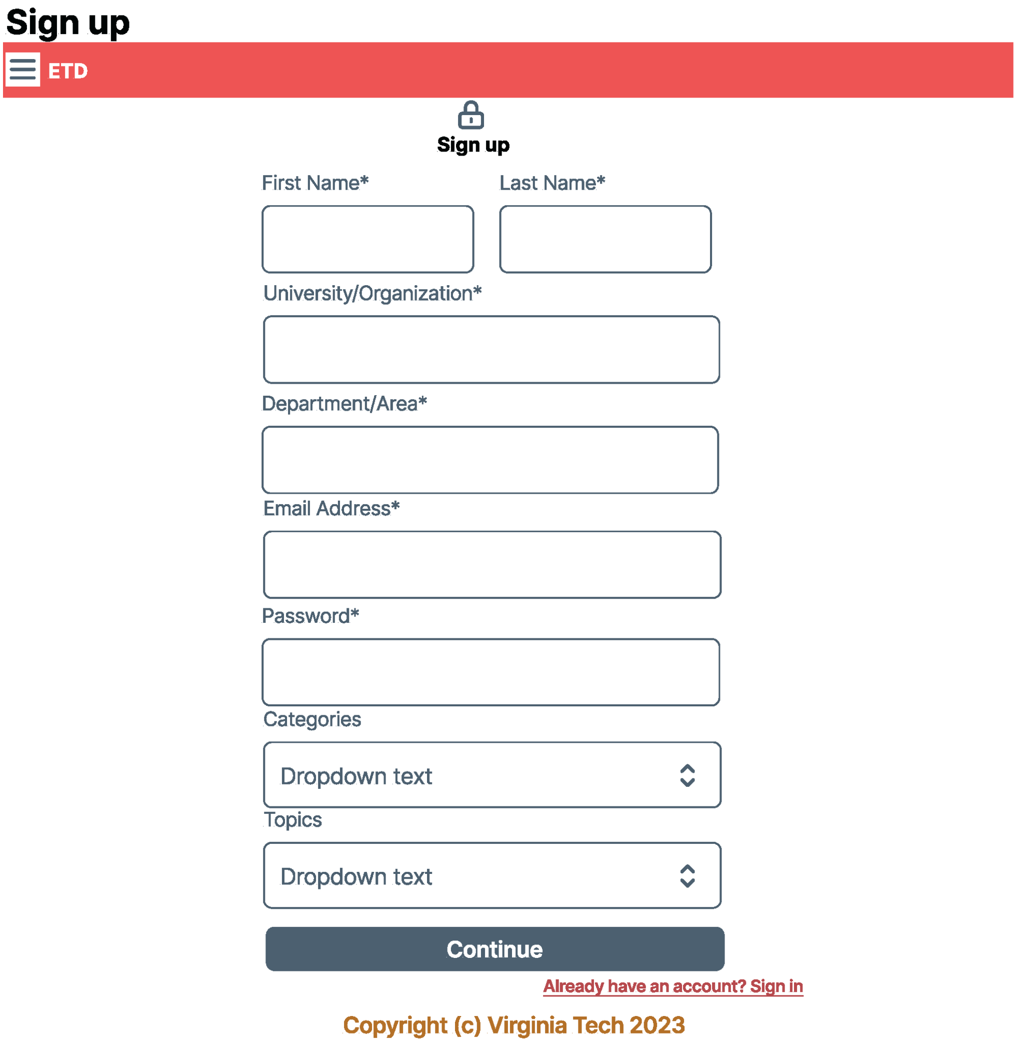
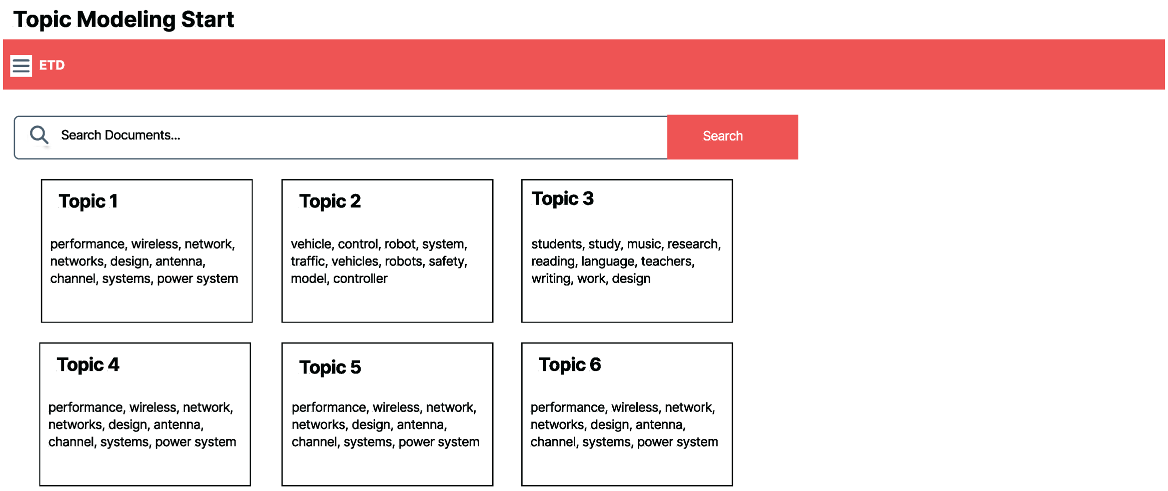


Figure 3.4: Signup

The Signup page, Fig. 3.4, was designed to gather all essential information about the user as well as their preferred topic modeling topics and ProQuest categories. The topics for topic modeling are extensive, and it was originally planned that they may need to be shown on a second screen designed in Fig. 3.5. However, in the end the topics were narrowed down to be displayed on one screen.



Topic Modeling Start

Topic 1

performance, wireless, network, networks, design, antenna, channel, systems, power system

Topic 2

vehicle, control, robot, system, traffic, vehicles, robots, safety, model, controller

Topic 3

students, study, music, research, reading, language, teachers, writing, work, design

Topic 4

performance, wireless, network, networks, design, antenna, channel, systems, power system

Topic 5

performance, wireless, network, networks, design, antenna, channel, systems, power system

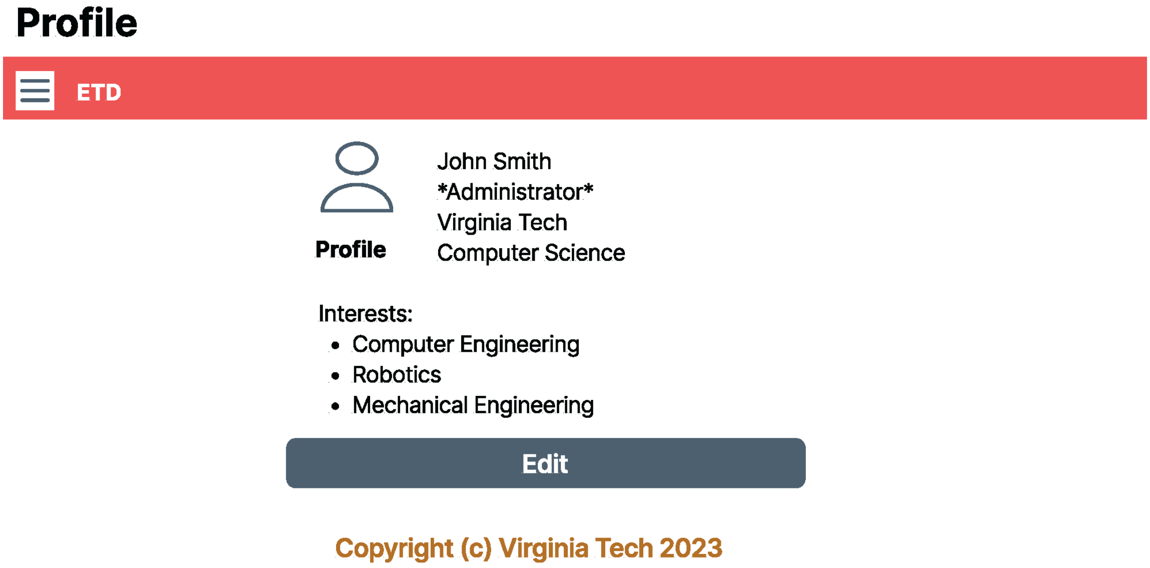
Topic 6

performance, wireless, network, networks, design, antenna, channel, systems, power system

Figure 3.5: Topics

The continue button takes the user to the more extensive topics page as shown in Fig. 3.5. This button was also discontinued in later versions of the system. These topics will be stored in a user topics table on the backend along with the user ID.

Finally the user needs to click the sign up button to complete the account creation process.



Profile

John Smith
Administrator
Virginia Tech
Computer Science

Interests:

- Computer Engineering
- Robotics
- Mechanical Engineering

Copyright (c) Virginia Tech 2023

Figure 3.6: User Profile

The user profile, Fig. 3.6, was a new addition to this version. It was meant to allow the user to see their current information and edit their interests. In this version of the system, there are different types of users: general users, curators, experimenters, and administrators. These designations will be given by matching the user ID with their designated role in the user roles table. The edit button on this page was discontinued in favor of displaying as editable data right away.

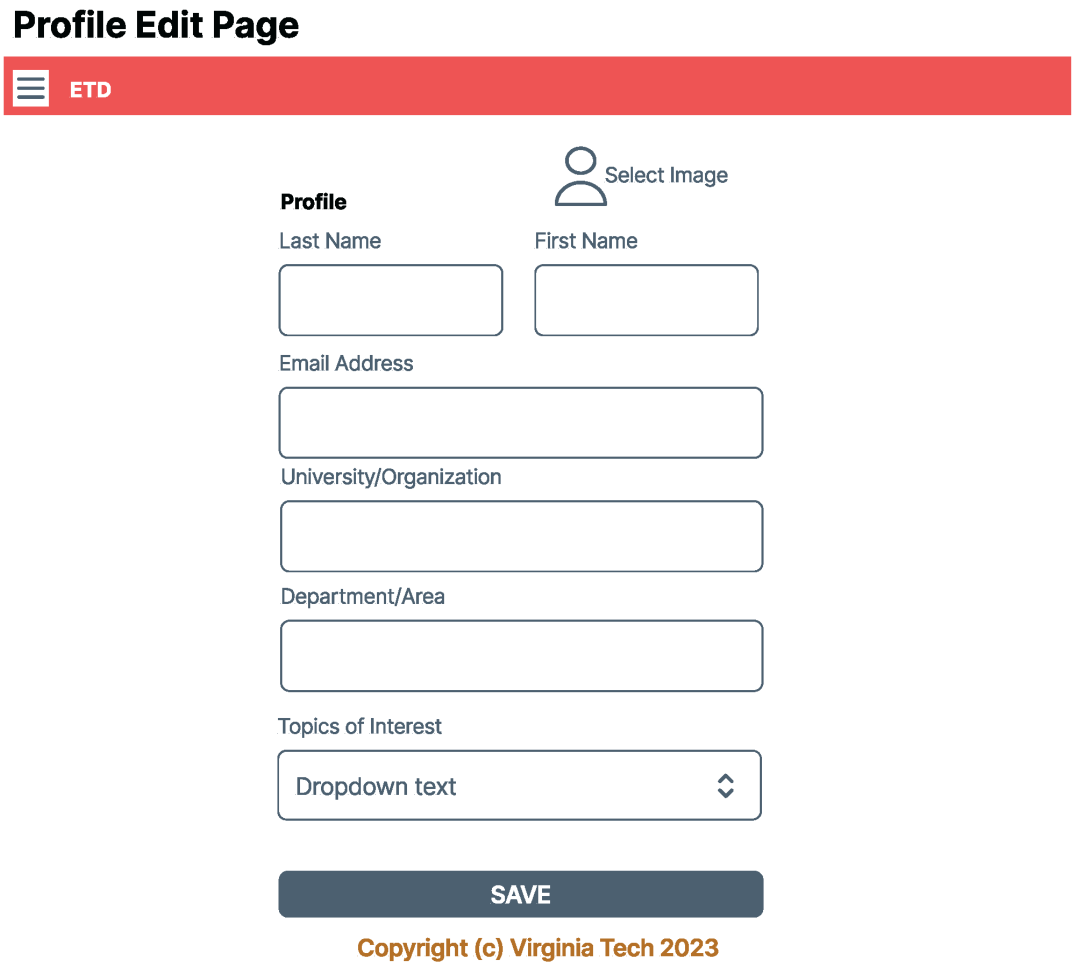


Figure 3.7: Edit Profile

The edit profile page shown in Fig. 3.7 gives users the opportunity to update their information if they were to, for example, switch institutions. It also allows them to edit their categories and topics of interest which would change their recommendations. This screen is closest to the final edit screen.

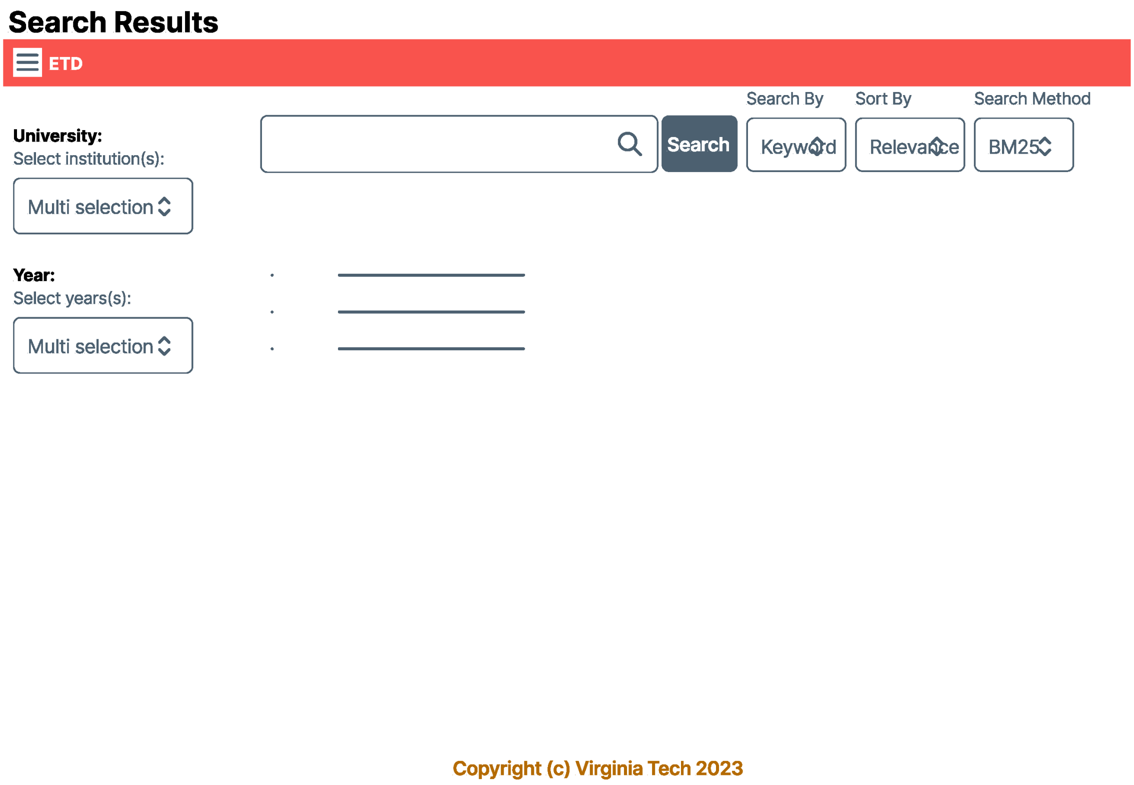
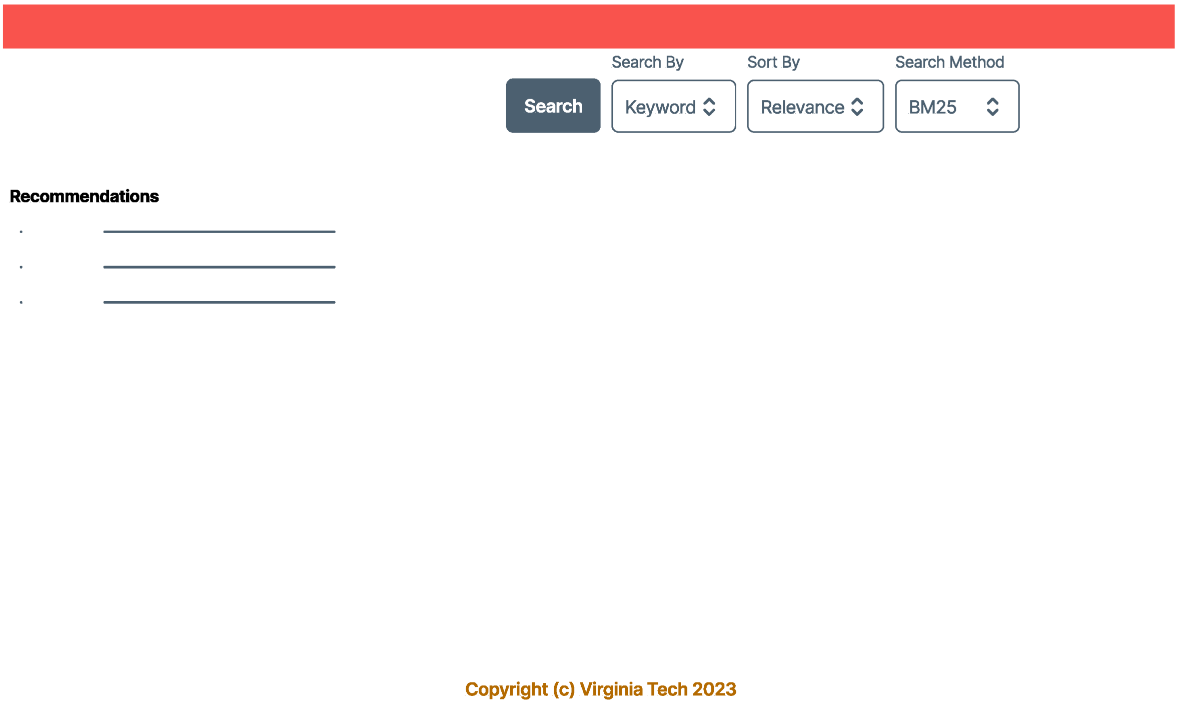


Figure 3.8: Search Results

The search page design (Fig. 3.8) shows advanced search parameters to narrow down results to more specific documents: fetch results from a specific university, year(s), keyword, relevance, etc.



Recommendations

Copyright (c) Virginia Tech 2023

Figure 3.9: Search Recommendations

The recommendations section, shown in Fig. 3.9, automatically searches the database using ElasticSearch capabilities and AI supported functions to match results with a user's interests which a user selected during signup. Recommendations will appear on the search page when a user first opens it.

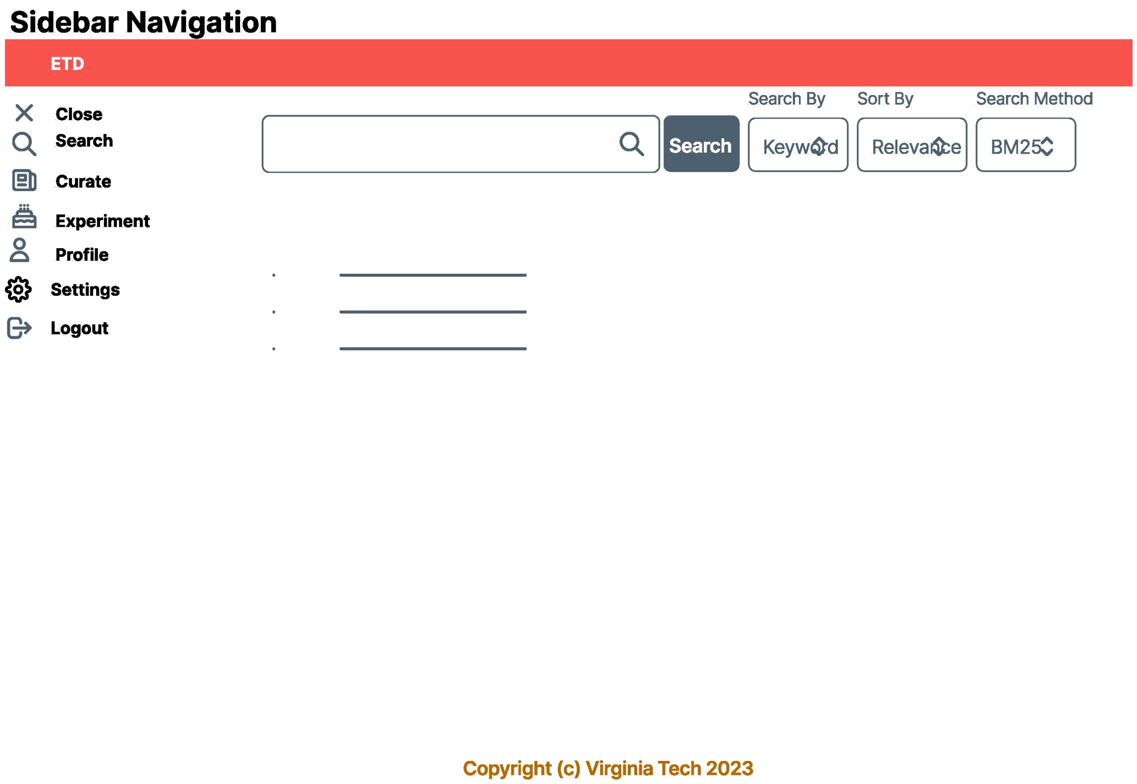


Figure 3.10: Side bar

The sidebar design shown in Fig. 3.10 showcases some of the functionality users can access through the system. They have the ability to curate and experiment with ETDs in addition to search functionality and access to their profile.

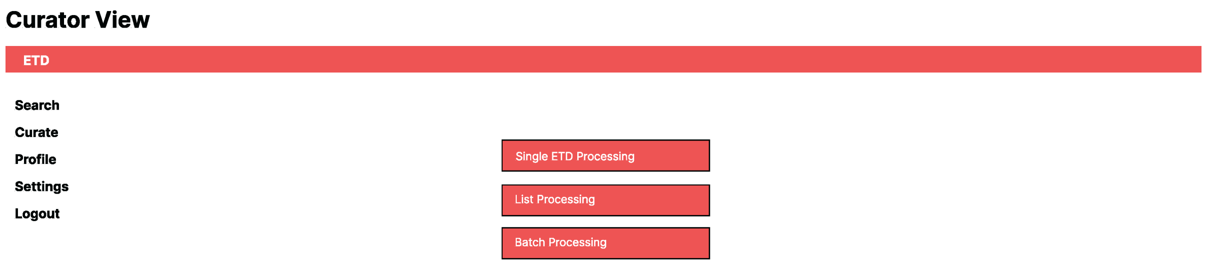


Figure 3.11: Curator Main Page

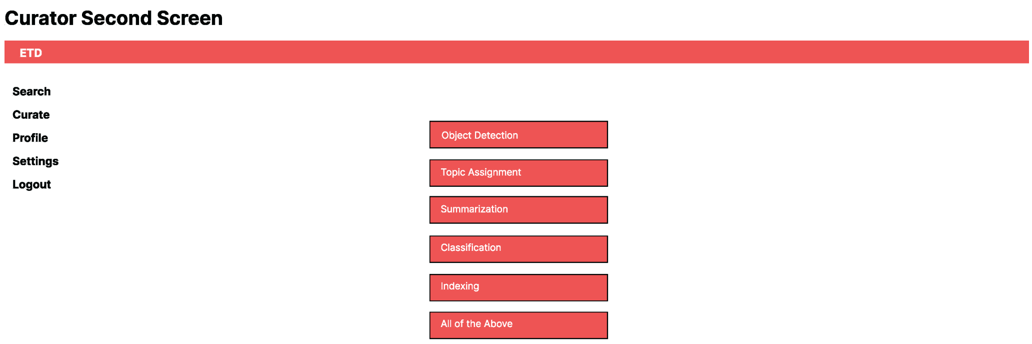


Figure 3.12: Curator Processing Choices

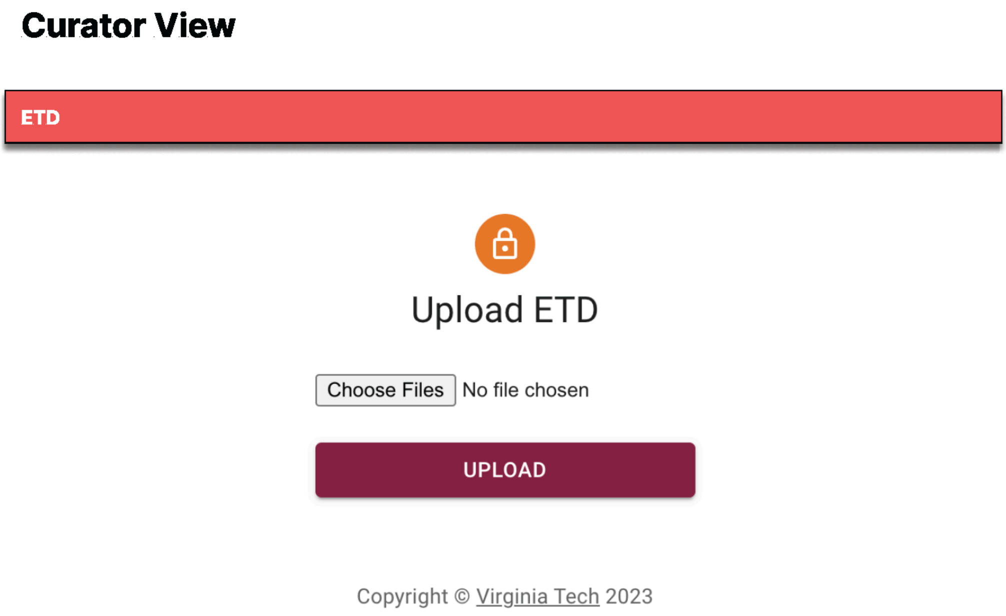


Figure 3.13: Curator Upload

The curator screen was heavily edited and iterated based on conversations with other teams. Shown in Figures 3.11, 3.12, and 3.13 is a curator design from midway through the iteration process. Users who are designated as curators will have the ability to upload and edit ETDs. This curator page was a multi-step process, allowing the user to choose how many ETDs to process and the method of processing to use, before being taken to a page to upload their ETDs.

These choices have since been narrowed down and streamlined into one screen.

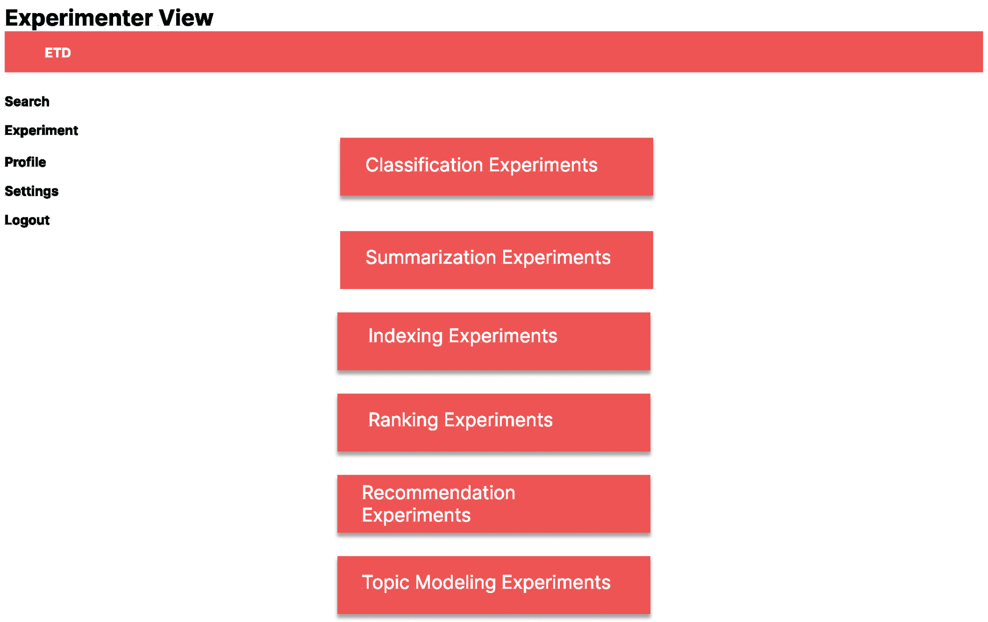

Experimenter View
ETD
Search
Experiment
Profile
Settings
Logout
Classification Experiments
Summarization Experiments
Indexing Experiments
Ranking Experiments
Recommendation Experiments
Topic Modeling Experiments

Figure 3.14: Experimenter

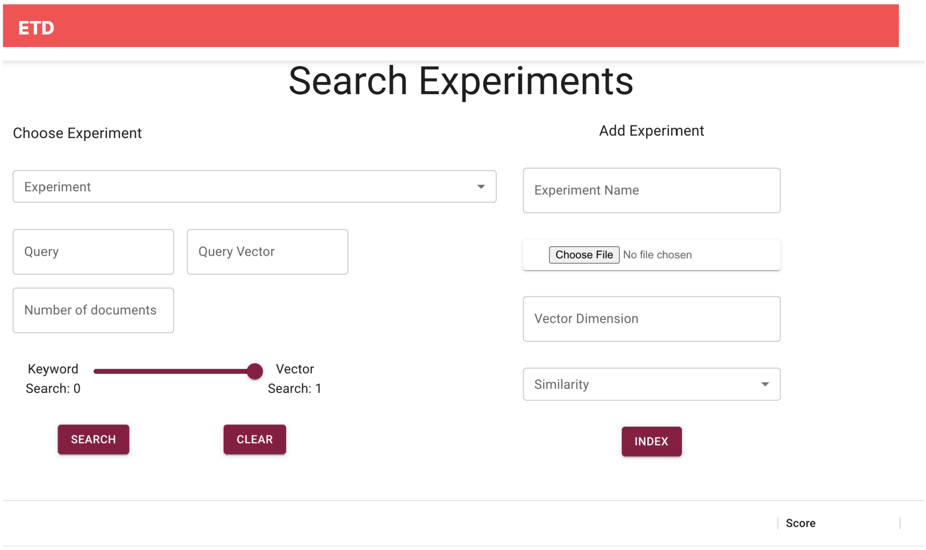


Figure 3.15: Manage Experiment

The experimenter page also went though a heavy iteration process. Figures 3.14 and 3.15 show a screen from midway through the process. This was a two screen design.

3.3 System Architecture

The frontend system requires interactions between the user, the user interface, and the backend database tables run by Team 5. The architecture was largely devised based on the previous year's work. The largest change to the architecture can be found in the login and authentication process. Now, once a user is logged in, every one of their API calls will contain a user authorization token. This token will be verified in the database tables before any results are returned, to ensure only registered users can access the system. Another addition to the architecture is the administrator user interface, which will allow people whose user role has been designated as administrator in the backend tables to edit the roles and privileges of other users.

Figure 3.16 depicts the plan for the architecture of the login and search process for a typical user. When a user logs in, an Auth token is returned that will be stored locally. The Auth token allows them to submit search queries and use the rest of the functionality of the site.

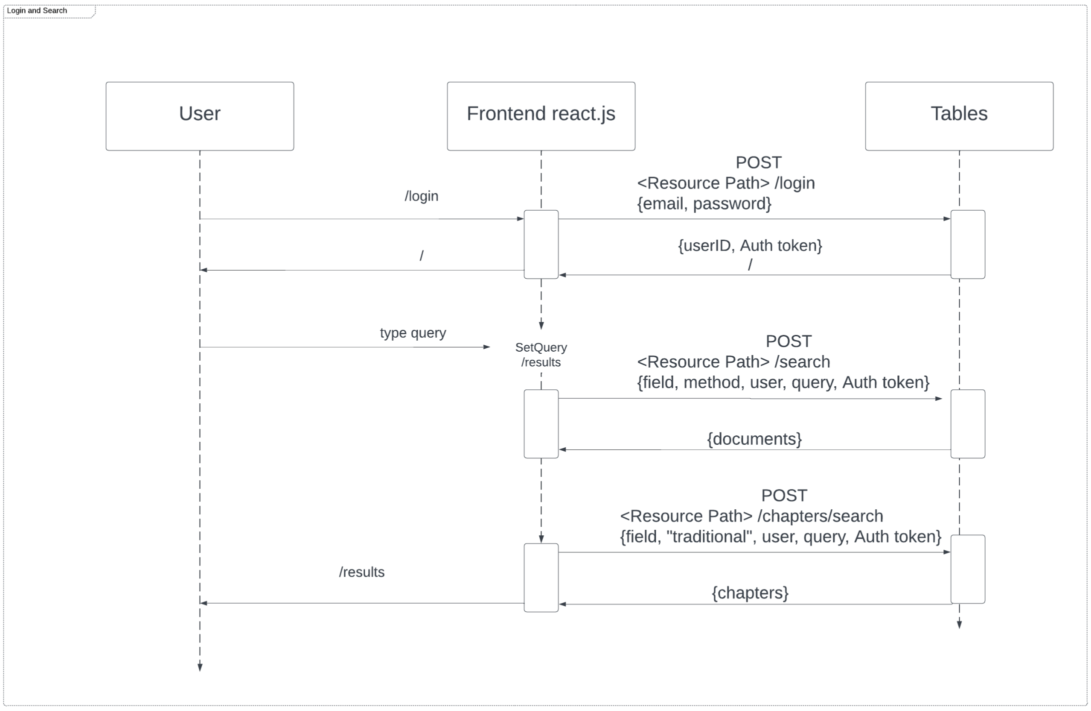


Figure 3.16: Login and Search Architecture

Figure 3.17 depicts the plan for the architecture of the signup process. When a user signs up for the site, they input their first name, last name, and password. This password is hashed and the password hash is used to verify future logins. The user also inputs a list of categories and topics they are interested in. This topic list is sent to the User_ topics table, and the selected categories go to the User_ classes table, where they can be used by the recommendations team. This architecture has since been updated to include checks for whether the email address is a vt.edu or odu.edu address, and a waiting period for newly signed up users to be approved.

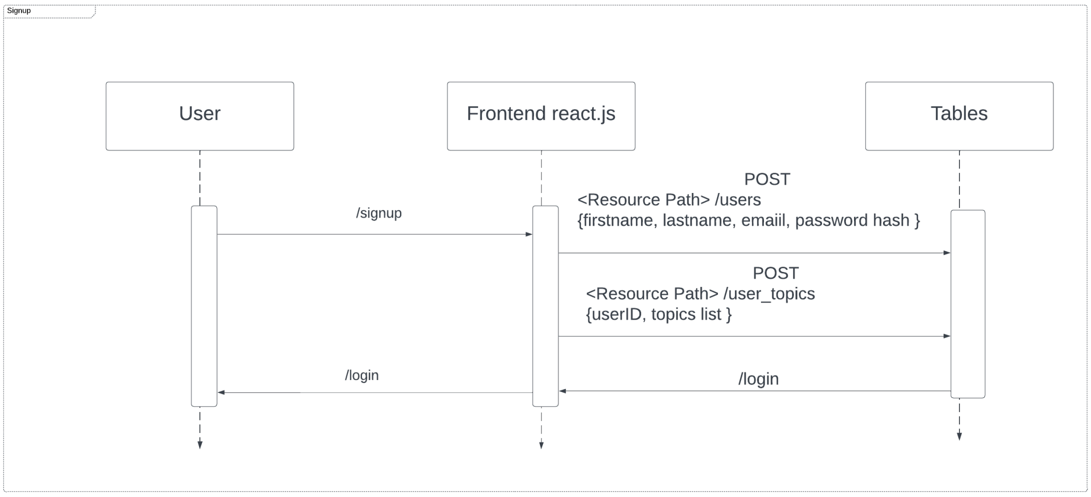


Figure 3.17: Sign Up Architecture

Figure 3.18 depicts the architecture of the administrator role assignment process.

The administrator has the ability to change roles for other users. The role options are general user, experimenter, curator, and administrator. When a user has experimenter or curator privileges, they are able to access those pages on the menu.

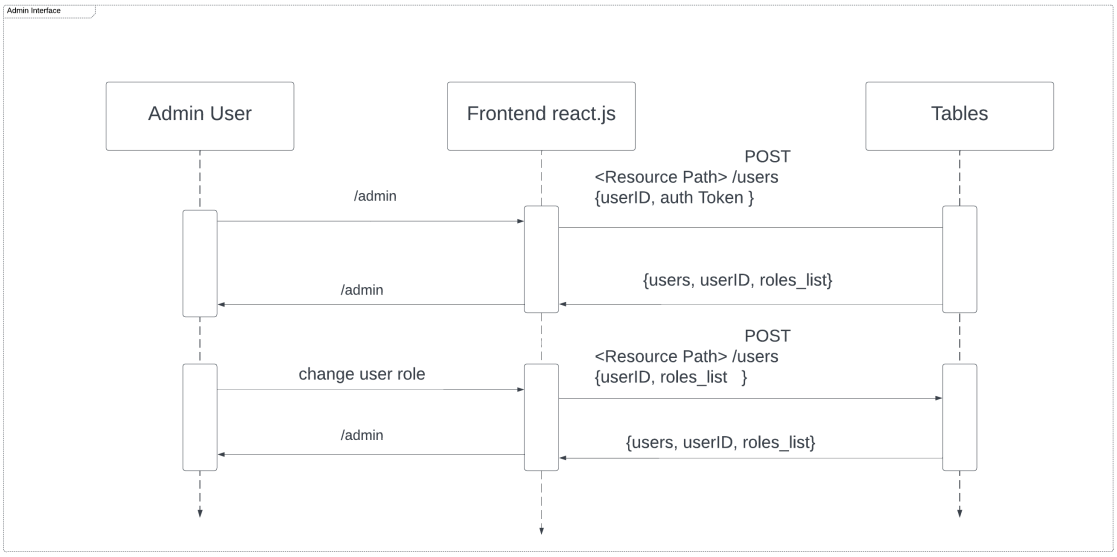


Figure 3.18: Administrator Architecture

Figure 3.19 depicts the process of a user trying to access the experimenter page. The architecture would be the same for a user trying to access the curator page or administrator page. When the user requests those pages, a request is sent to the tables to check their user roles. If their user roles show that they have access, they will be taken to the experimenter page. If not, they will be taken back to the general search page.

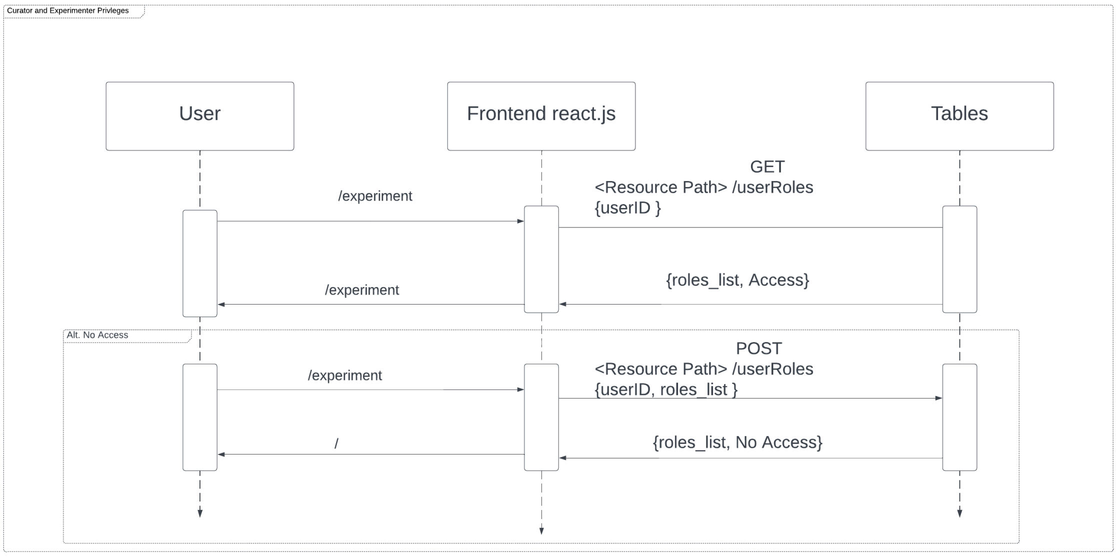


Figure 3.19: Experimenter Architecture

3.4 Tools

The main tools used by this team were:

1. Gitlab: For storing and sharing code
2. React.js: For building user interfaces
3. Figma: For designing wireframes
4. Lucidchart: For building UML and user flow diagrams
5. NGINX: This is a high-performance, open-source web server and reverse proxy server. It is known for its efficiency in handling concurrent connections and serving static content. Nginx is widely used as a load balancer, reverse proxy, and web server for deploying and scaling web applications. It's designed to be lightweight, scalable, and efficient, making it a popular choice for improving the performance and reliability of web applications.

Chapter 4

Implementation

4.1 Timeline

- 8/24-8/31: Determined what tools we would use. We examined the current codebase and decided that we would use React.js to code our interface so that we would be able to work with the current code base, which was also done in React.js. We decided that to wireframe and plan designs we would use Figma [26] because it has functionality for group editing. For any architecture charts and user flow designs we use LucidChart [27].

- 9/5-9/7: We determined what changes needed to be made between the old and the new system, such as authentication. We then examined the current user interface for changes we thought could enhance the user experience, such as an opportunity to view and edit the profile page and change their listed interests. The third set of changes to be made are additions to the user interface based on new features other teams have been working on.

- 9/12-9/14: Once we had a list of changes to start from, we began the process of creating a new user flow chart to represent how we expect the user to be able to navigate through the new authentication process and pages. We also designed wireframes to showcase changes to the pre-existing pages and shared them with other teams to receive their feedback.

- 9/19-9/21: We gained access to the existing code base and made sure we could get it to properly run locally. We were able to complete an initial setup of the project.

- 9/26-10/5: With access to this code, and collaboration with Team 5, we were able to outline what API interactions would take place between the user interface and the database tables. We made and consulted on UML diagrams for these actions.

- 10/10-10/12: While waiting for work on the APIs to be done, we began work on new pages to be added to the interface. We completed an administrator page and started work on wireframes for the curator and experimenter pages.

- 10/17-10/26: We received lots of feedback on our curator and experimenter wireframes and held in-depth conversations with all of the teams to understand what functionality needed to be supported on these pages. We continued to iterate designs based on this feedback. In addition, we continued to work with Team 5 to get login and signup API calls to work.

- 10/31-11/9: Login and signup became functional. We were able to send signup information to the database via the APIs, login users who were saved in the database, and return user tokens that limited a user's access to the system based on their role.

- 11/14-11/30: We continued to improve the signup and authentication process. Working with Team 2 we added a categories selection to signup, based on level 2 ProQuest categories, that will save a user's preference to help their recommendations. We also worked with Team 3 to add a topics selection, allowing users to select preferred topics for topic modeling.

- 11/30-12/5: Our last accomplishment was making the signup process more secure, limiting the ability to sign up to only vt.edu and odu.edu email addresses, and working with Team 5 so that all users who sign up have to be manually approved from the backend database before they can access the system. The UI was adjusted to alert users whether their signup was successful.

4.2 Future Work

We mentioned future work in Section 1.4; however, this section will provide more guidance to help extend the work.

- Testing Experiment and Curation Features: While we have implemented the curator and experimenter UIs, we have not integrated them with the relevant APIs and therefore did not test their functionality and usability. This was due to delays from Team 3 and Team 4 in implementing the curation and experimentation functions. Future extension work should consider: integrating the curator and the experimenter UIs with the APIs, testing them, and ensuring they work as intended.

- User Testing of All Pages: Another area of focus towards improving the functionality, usability, and effectiveness of the ETD system is to conduct user studies. Random users should be asked to play with the ETD system and provide feedback via a survey. The analysis of the feedback should be used to address areas of concerns.

- Topic Modeling Interface: The Topic Modeling team was able to categorize the ETDs in the system according to a collection of topics. A specific topic modeling centered search should be added to the ETD interface to utilize their work.

  4. Adding Support for Administrators Analyzing Trends: Currently the UI lacks Statistics Report forms. These forms are essential to aid administrators in analyzing and understanding logging trends, including other factors that can help assess the ETD performance across key indicators, such as most searched articles, most contributing institutions / countries, etc.

Chapter 5

User Manual

5.1 Screens and Guides

This chapter shows the React.js coded pages that are currently functional on the ETD website, and directs a user on how to utilize them.

A user can access the system by going to: https://cs5604.endeavour.cs.vt.edu/login. This screen is shown in Figure 5.1.

5.1.1 Login and Sign Up

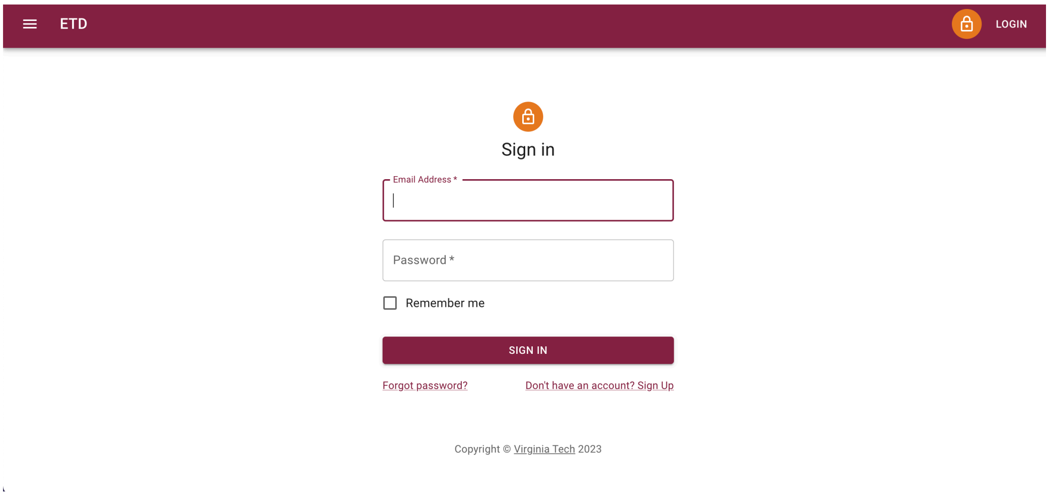
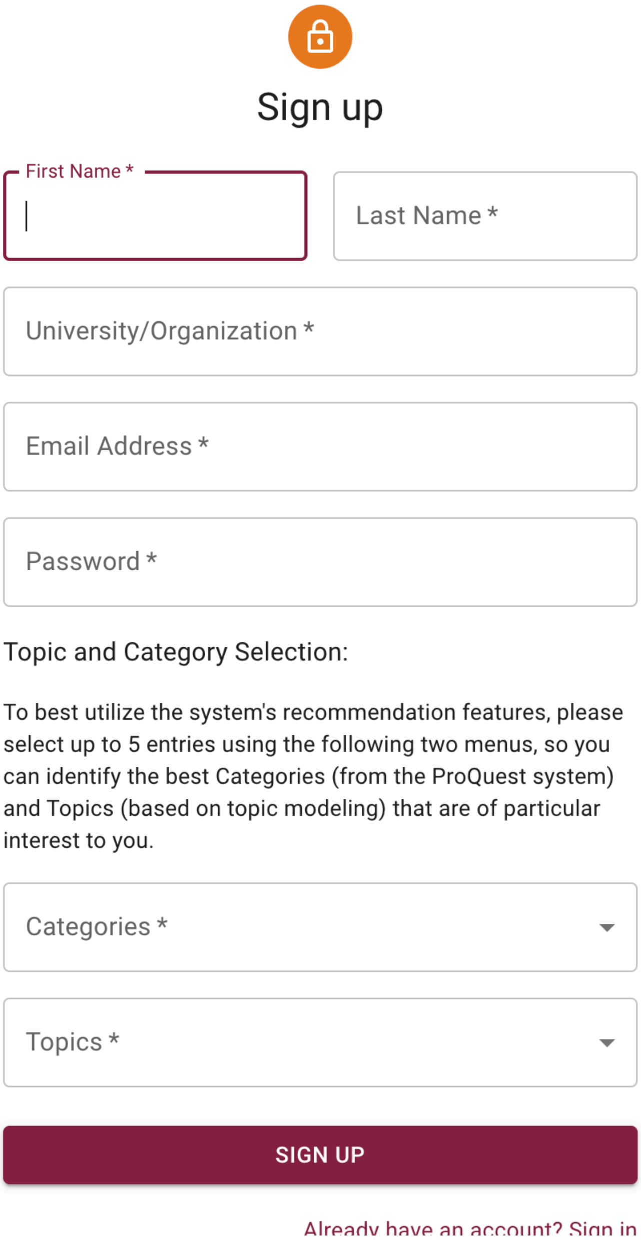


Figure 5.1: Login

Below the login box, the user can click the button to sign up if they do not have an account. This will take them to the signup page shown in Figure 5.2.



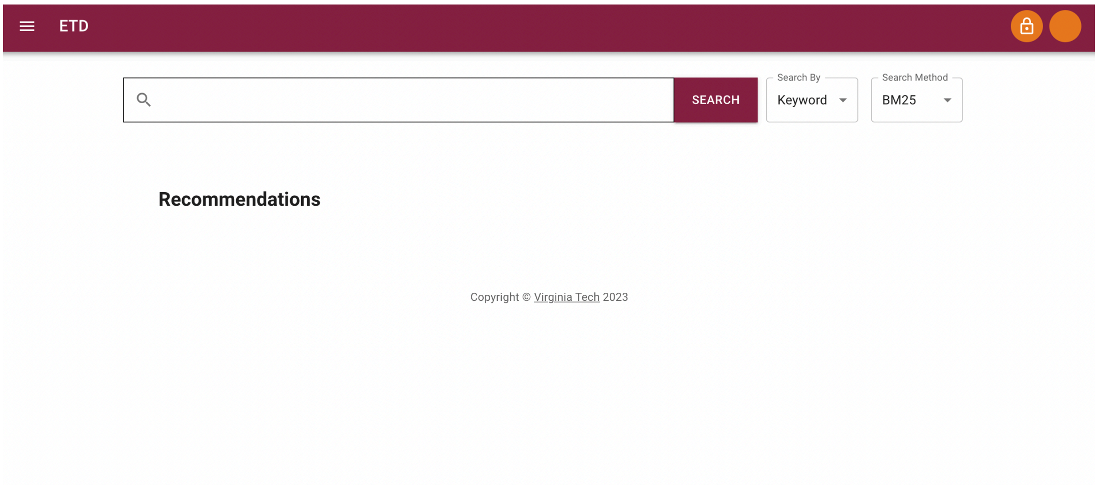
Topic and Category Selection:

To best utilize the system's recommendation features, please select up to 5 entries using the following two menus, so you can identify the best Categories (from the ProQuest system) and Topics (based on topic modeling) that are of particular interest to you.

Figure 5.2: Signup

The user will only be able to sign up if they have a vt.edu or odu.edu email address and are approved from the backend. Once signed up, the user will be directed back to the login page to login. After logging in, the user will be taken to their recommendations and search page, shown in Figure 5.3.

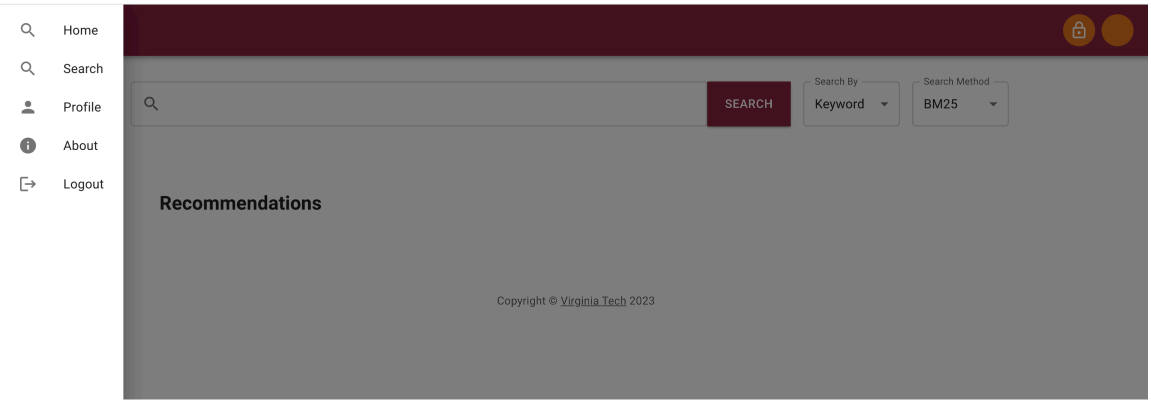
5.1.2 Recommendations



Copyright © Virginia Tech 2023

Figure 5.3: Recommendations

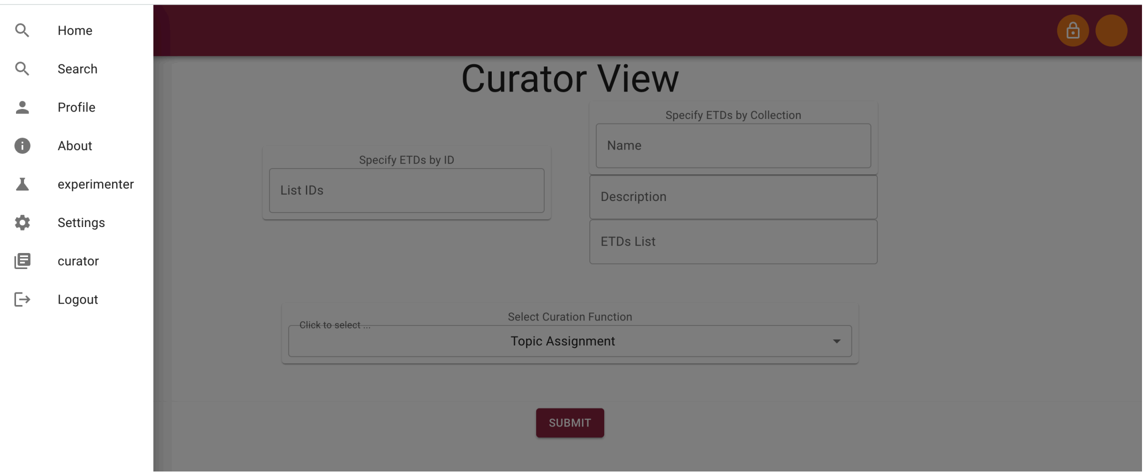
5.2 Navigation



Copyright © Virginia Tech 2023

Figure 5.4: User Navigation

The main method of navigating the site is the sidebar. Shown in Figure 5.4 is the navigation bar for a regular user.



Select Curation Function: Topic Assignment

Figure 5.5: Curator Navigation

The sidebar will only show curator and experimenter options if the user has those privileges. The curator navigation is depicted in Figure 5.5 and the experimenter navigation is depicted in Figure 5.6.

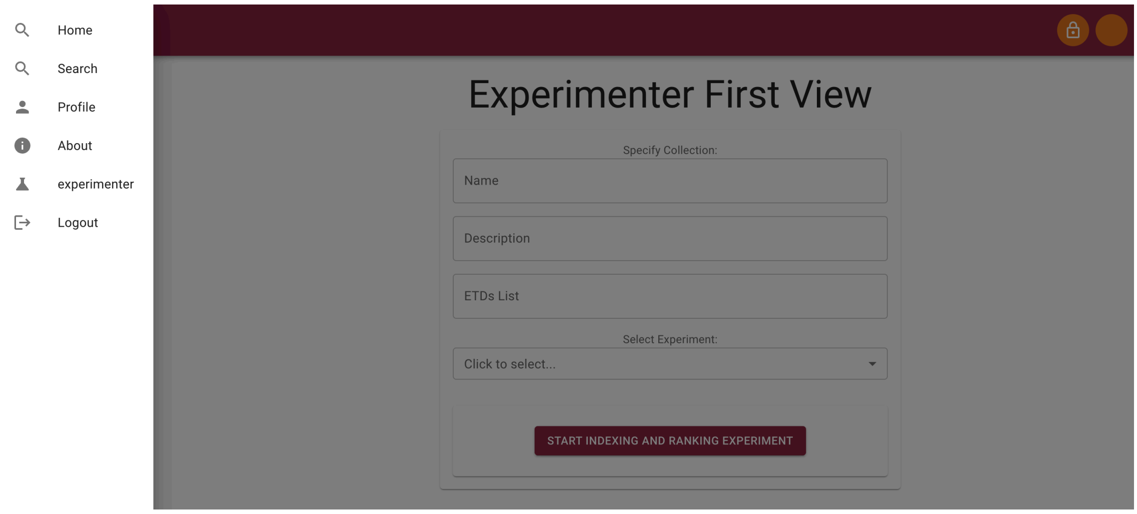
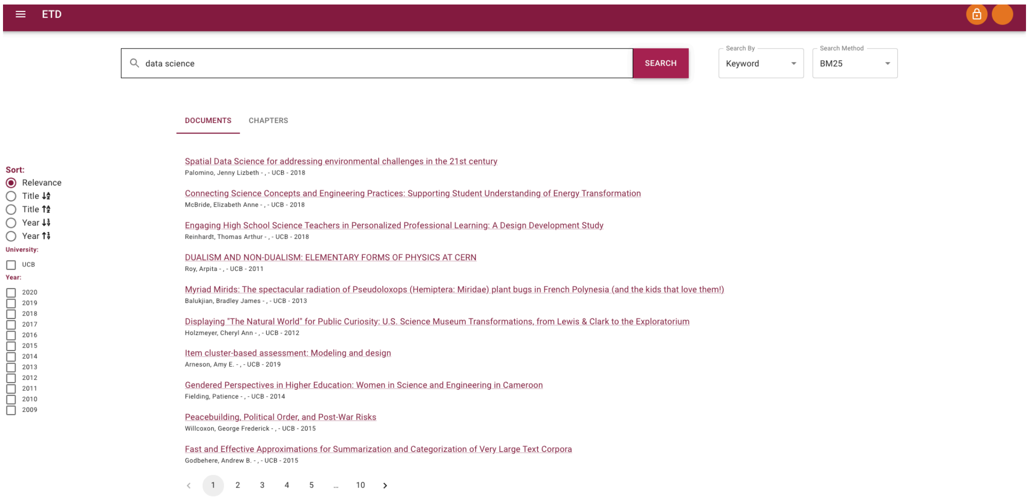


Figure 5.6: Experimenter Navigation

Once logged in, users can access the search page, where they currently have the ability to search by keyword and view documents or chapters. The user in Figure 5.7 has searched for “data science”

5.2.1 Search



data science

Search By: Keyword

Search Method: BM25

Sort:
● Relevance
○ Title ↓
○ Title ↑
○ Year ↓
○ Year ↑

University:
□ UCB

Year:
□ 2020
□ 2019
□ 2018
□ 2017
□ 2016
□ 2015
□ 2014
□ 2013
□ 2012
□ 2011
□ 2010
□ 2009

Spatial Data Science for addressing environmental challenges in the 21st century
Palomino, Jenny Lizbeth - , - UCB - 2018

Connecting Science Concepts and Engineering Practices: Supporting Student Understanding of Energy Transformation
McBride, Elizabeth Anne - , - UCB - 2018

Engaging High School Science Teachers in Personalized Professional Learning: A Design Development Study
Reinhardt, Thomas Arthur - , - UCB - 2018

DUALISM AND NON-DUALISM: ELEMENTARY FORMS OF PHYSICS AT CERN
Roy, Arpita - , - UCB - 2011

Myriad Mirids: The spectacular radiation of Pseudoloxops (Hemiptera: Miridae) plant bugs in French Polynesia (and the kids that love them!)
Balukjian, Bradley James - , - UCB - 2013

Displaying "The Natural World" for Public Curiosity: U.S. Science Museum Transformations, from Lewis & Clark to the Exploratorium
Holzmeyer, Cheryl Ann - , - UCB - 2012

Item cluster-based assessment: Modeling and design
Arneson, Amy E. - , - UCB - 2019

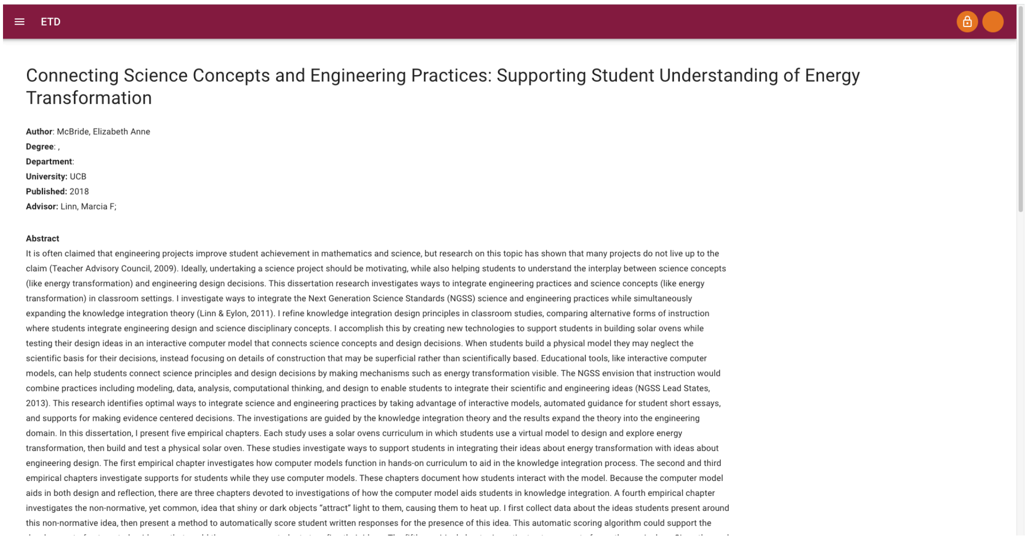
Gendered Perspectives in Higher Education: Women in Science and Engineering in Cameroon
Fielding, Patience - , - UCB - 2014

Peacebuilding, Political Order, and Post-War Risks
Willcoxon, George Frederick - , - UCB - 2015

Fast and Effective Approximations for Summarization and Categorization of Very Large Text Corpora
Godbehere, Andrew B. - , - UCB - 2015

Figure 5.7: Search

A user can click on one of the recommendation search results to view the document or chapter as shown in Figure 5.8.



Connecting Science Concepts and Engineering Practices: Supporting Student Understanding of Energy Transformation

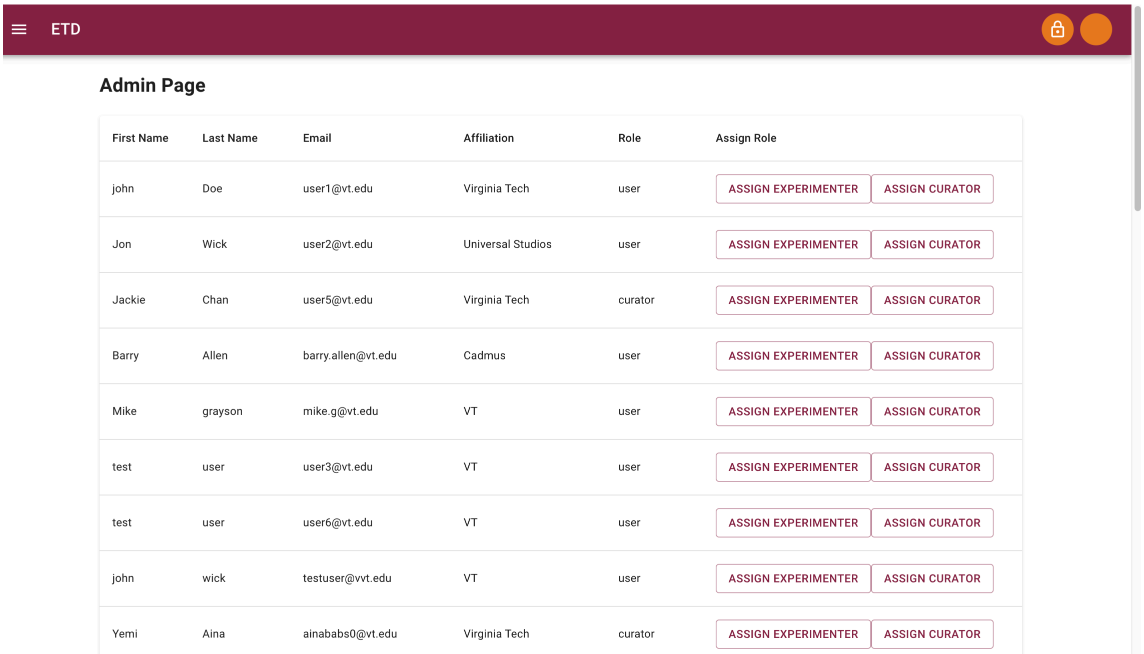
Author: McBride, Elizabeth Anne
Degree: ,
Department:
University: UCB
Published: 2018
Advisor: Linn, Marcia F;

Abstract
It is often claimed that engineering projects improve student achievement in mathematics and science, but research on this topic has shown that many projects do not live up to the claim (Teacher Advisory Council, 2009). Ideally, undertaking a science project should be motivating, while also helping students to understand the interplay between science concepts (like energy transformation) and engineering design decisions. This dissertation research investigates ways to integrate engineering practices and science concepts (like energy transformation) in classroom settings. I investigate ways to integrate the Next Generation Science Standards (NGSS) science and engineering practices while simultaneously expanding the knowledge integration theory (Linn & Eylon, 2011). I refine knowledge integration design principles in classroom studies, comparing alternative forms of instruction where students integrate engineering design and science disciplinary concepts. I accomplish this by creating new technologies to support students in building solar ovens while testing their design ideas in an interactive computer model that connects science concepts and design decisions. When students build a physical model they may neglect the scientific basis for their decisions, instead focusing on details of construction that may be superficial rather than scientifically based. Educational tools, like interactive computer models, can help students connect science principles and design decisions by making mechanisms such as energy transformation visible. The NGSS envision that instruction would combine practices including modeling, data, analysis, computational thinking, and design to enable students to integrate their scientific and engineering ideas (NGSS Lead States, 2013). This research identifies optimal ways to integrate science and engineering practices by taking advantage of interactive models, automated guidance for student short essays, and supports for making evidence centered decisions. The investigations are guided by the knowledge integration theory and the results expand the theory into the engineering domain. In this dissertation, I present five empirical chapters. Each study uses a solar ovens curriculum in which students use a virtual model to design and explore energy transformation, then build and test a physical solar oven. These studies investigate ways to support students in integrating their ideas about energy transformation with ideas about engineering design. The first empirical chapter investigates how computer models function in hands-on curriculum to aid in the knowledge integration process. The second and third empirical chapters investigate supports for students while they use computer models. These chapters document how students interact with the model. Because the computer model aids in both design and reflection, there are three chapters devoted to investigations of how the computer model aids students in knowledge integration. A fourth empirical chapter investigates the non-normative, yet common, idea that shiny or dark objects "attract" light to them, causing them to heat up. I first collect data about the ideas students present around this non-normative idea, then present a method to automatically score student written responses for the presence of this idea. This automatic scoring algorithm could support the

Figure 5.8: Document View

A user who is designated as an administrator from the backend will be able to access and change the privileges of other users from the administrator interface shown in Figure 5.9.

5.3 Admin and Roles



Admin Page

First Name	Last Name	Email	Affiliation	Role	Assign Role
john	Doe	user1@vt.edu	Virginia Tech	user	ASSIGN EXPERIMENTER ASSIGN CURATOR
Jon	Wick	user2@vt.edu	Universal Studios	user	ASSIGN EXPERIMENTER ASSIGN CURATOR
Jackie	Chan	user5@vt.edu	Virginia Tech	curator	ASSIGN EXPERIMENTER ASSIGN CURATOR
Barry	Allen	barry.allen@vt.edu	Cadmus	user	ASSIGN EXPERIMENTER ASSIGN CURATOR
Mike	grayson	mike.g@vt.edu	VT	user	ASSIGN EXPERIMENTER ASSIGN CURATOR
test	user	user3@vt.edu	VT	user	ASSIGN EXPERIMENTER ASSIGN CURATOR
test	user	user6@vt.edu	VT	user	ASSIGN EXPERIMENTER ASSIGN CURATOR
john	wick	testuser@vvt.edu	VT	user	ASSIGN EXPERIMENTER ASSIGN CURATOR
Yemi	Aina	ainababs0@vt.edu	Virginia Tech	curator	ASSIGN EXPERIMENTER ASSIGN CURATOR

Figure 5.9: Admin Roles

If a user holds curator powers, something determined by the system administrator, they will also be able to access the curator option via a sidebar menu. This will take them to the curator page shown in Figure 5.10.

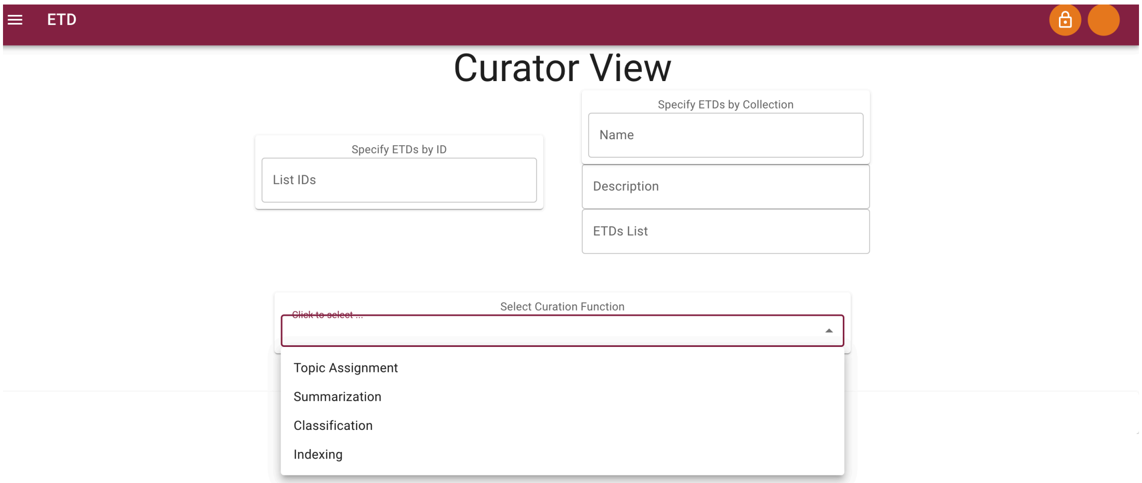


Figure 5.10: Curating page

If a user holds experimenter powers, something determined by the system administrator, they will also be able to access the experimenter option via a sidebar menu. This will take them to the experimenter page shown in Fig. 5.11. The experimenter process involved two screens, one to select the type of experiment you are trying to run and one to take you to a more specific detail page for your choices. These choices are classification and summarization, or indexing and ranking, shown in Figures 5.12 and 5.13.

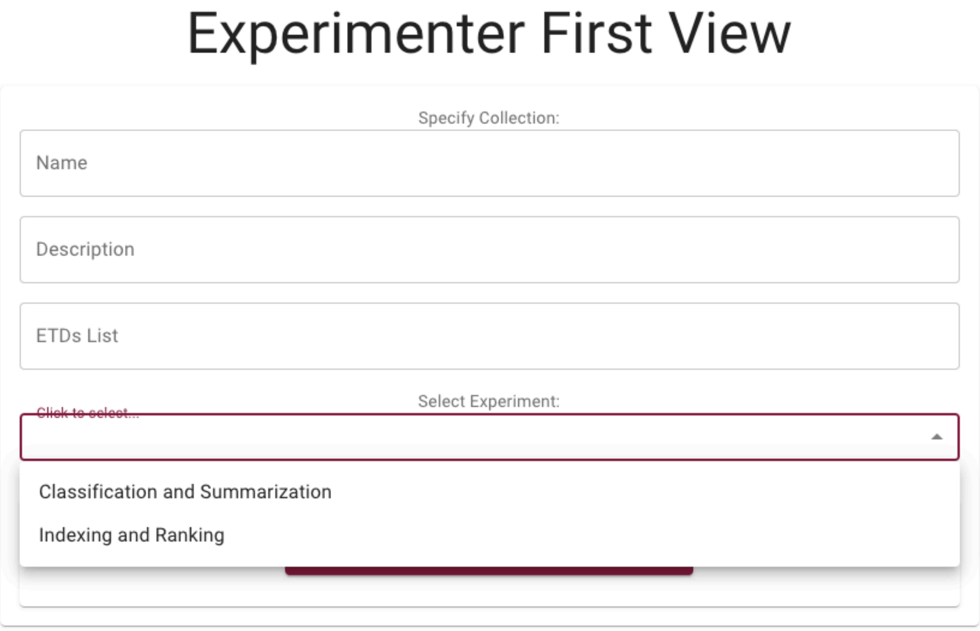


Figure 5.11: Categories of experiments in dropdown

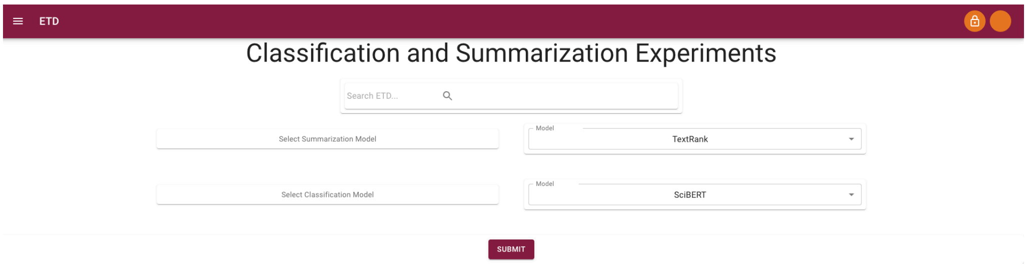

ETD
Classification and Summarization Experiments
Search ETD...
Select Summarization Model
Model
TextRank
Select Classification Model
Model
SciBERT
SUBMIT

Figure 5.12: Classification and Summarization Experiment

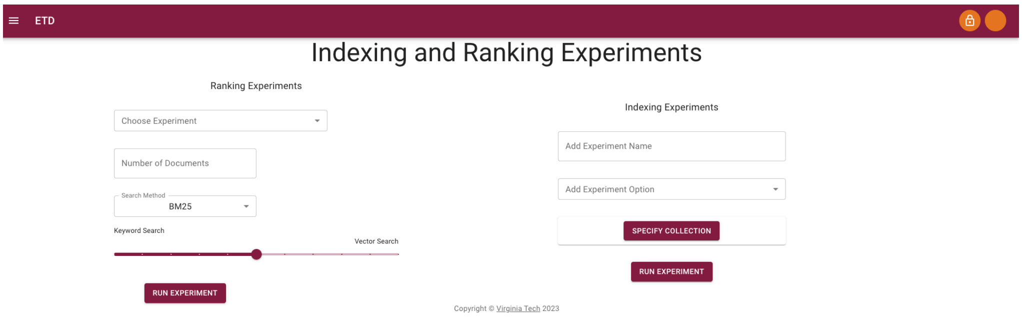


Figure 5.13: Indexing and Ranking Experiments

Chapter 6

Developer Manual

6.1 Locally Accessing and Run the Interface

In order to access and run the code base locally:

- Download the code from its Gitlab repository which can be found at https://code.vt.edu/cs5604-f2023/team6-ui/cs5604-ui-2023
- Open the project directory
- Run 'npm start'
  - This will require you to have Node.js, npm, and all the project dependencies downloaded.
  - Open the project in your preferred code editor (e.g., Visual Studio Code, Atom, etc.).
  - To install the project dependencies, run 'npm install' inside the project directory.
  - Make any necessary changes in real-time and save.
  - You can also commit and pull changes from Git with your collaborators. We learned that in order to avoid Git conflicts, always perform Fetch/Pull before committing/pushing your changes to the repository.
- Open http://localhost:3000 to view it in your browser.

6.2 Some Dependencies for the ETD Project

After forking the ETD project you might run into dependency errors. To fix these, there are some dependencies (Table 6.1) you must install locally before running the project.

Table 6.1: Library Dependencies

Library	Installation Command
React	`npm install react`
Material-UI Core	`npm install @mui/material`
Material-UI Icons	`npm install @mui/icons-material`
React Router DOM	`npm install react-router-dom`

6.3 Building and Deploying to NGINX Locally

Beyond testing your work on localhost: 3000, you can test it on the NGINX server locally to see how it would work in a live server and communicate with the backend.

Steps:

Install NGINX: You can install it globally via the terminal with this command *sudo apt-get update sudo apt-get install nginx*

Then in the NGINX folder, look for the nginx.conf file and configure as shown in Listing 6.1 below.

Listing 6.1: NGINX Configuration

```
    server {
        listen 80;
        server_name localhost;
        root /path/to/my-react-app/build;
        index index.html;
        location / {
            try_files $uri /index.html;
        }
    }
```

6.4 ETD Files and Purpose

The files that can be found in the Gitlab repository are listed in Table 6.2 along with their function in the codebase:

Table 6.2: Files and Descriptions

No.	File	Purpose
1.	ChapterResults.js	When a user searches for ETDs, clicking Chapters on the Home page or Recommendations page will display the filtered chapters.
2.	Footer.js	Displays footer content on all other pages.

Continued on the next page

Table 6.2 – continued from previous page

No.	File	Purpose
3.	Header.js	The Header contains Menu links and Login Icon.
4.	PieChart.js	This file is part of the StatisticReport, and displays report visually.
5.	Recommendations.js	Works with the user-selected topics and categories during signup to automatically generate matching search results for the user.
6.	SearchBar.js	Creates the search bar component to be reused on other pages
7.	SearchResults.js	Creates the search result component to be reused on other pages
8.	TabularRow.js	Helps format items into tables across the interface
9.	TabularView.js	Helps format items into tables across the interface
10.	TopicModelingTopics.js	Creates the element that allows a user to select preferred topics for topic modeling, to be reused on other pages such as signup.
11.	TopicsOfInterestSelect.js	Creates the element that allows a user to select preferred categories for recommendations, to be reused on other pages such as signup.
12.	About.js	Provides an introduction about the ETD system to a new user.
13.	admin.js	This page displays a list of all ETD system registered users. The curator can also assign curating and experimenting roles here. The admin-new.js is a future work to include revoke roles, including possible removal of registered users.
14.	ChapterView.js	This page displays the details of a Chapter once a user has selected it from the search results.
15.	CuratorView.js	The page for curation, where users can upload ETDs and select which curation functions to run on them.
16.	DocumentView.js	This page allows the user to view the details of a document once they have selected it from the search results.
17.	Experimenter.js	The page for running summarization and classification experiments.
18.	Home.js	The home page presented to the user after successful sign-in. It has a search for ETDs.
19.	IndexRanking.js	The page for running indexing and ranking experiments.

Continued on the next page

Table 6.2 – continued from previous page

No.	File	Purpose
20.	Login.js	This is the entry gateway to the ETD system where users are authenticated, provided they are signed up into the ETD system.
21.	Profile.js	Registered users can access their profile here and can edit as needed.
22.	Results.js	The file processes user search and provides the results.
23.	Search.js	This page renders the search results.
24.	SignUp.js	This page enables prospective users of the ETD system to register and use the system.
25.	StatisticReport.js	This page was intended to display statistics of interest for analysis and observation of trends, but it requires further implementation.
26.	Constants.js	The constants file contains API links.
27.	App.js	This file renders all pages to display on the browser.

Bibliography

[1] A. Teegalapally, M. Muralikrishnan, M. Paidiparthy, R. Ramanujan, R. Balar, S. Juvekar, and V. Murali, “CS5604 (Information Retrieval) F2022 End User (EU) Team Project.” http://hdl.handle.net/10919/114080, 2022. VTechWorks, Virginia Tech term project submission, Blacksburg, VA 24061.

[2] J. Lin, M. Pang, and K. Liu, “CS4624: Multimedia, Hypertext, and Information Access, Topic Modeling Toolkit.” http://hdl.handle.net/10919/115648, 2023. VTechWorks, Virginia Tech term project submission, Blacksburg, VA 24061.

[3] S. Chen, U. R. Thaduri, and V. K. R. Ballamudi, “Front-End Development in React: An Overview,” *Engineering International*, vol. 7, no. 2, pp. 117–126, 2019.

[4] A. Pavlenko, N. Askarbekuly, S. Megha, and M. Mazzara, “Micro-frontends: application of microservices to web front-ends,” *J. Internet Serv. Inf. Secur.*, vol. 10, no. 2, pp. 49–66, 2020.

[5] Y. Xing, J. Huang, and Y. Lai, “Research and analysis of the front-end frameworks and libraries in e-business development,” in *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering*, pp. 68–72, 2019.

[6] P. Singh, M. Srivastava, M. Kansal, A. P. Singh, A. Chauhan, and A. Gaur, “A comparative analysis of modern frontend frameworks for building large-scale web applications,” in *2023 International Conference on Disruptive Technologies (ICDT)*, pp. 531–535, IEEE, 2023.

[7] P. Dwivedi, A. Joshi, *et al.*, “ReactJS For Trading Applications,” in *2022 International Conference on Cyber Resilience (ICCR)*, pp. 01–07, IEEE, 2022.

[8] K. A. A. Bakar and G. R. Haron, “Adaptive Authentication: Issues and Challenges,” in *2013 World Congress on Computer and Information Technology (WCCIT)*, pp. 1–6, IEEE, 2013.

[9] A. Paul and M. Haldar, “UI Component and Authentication,” in *Serverless Web Applications with AWS Amplify: Build Full-Stack Serverless Applications Using Amazon Web Services*, pp. 23–73, Springer, 2023.

[10] A. De Angeli, M. Coutts, L. Coventry, G. I. Johnson, D. Cameron, and M. H. Fischer, “VIP: a visual approach to user authentication,” in *Proceedings of the working conference on advanced visual interfaces*, pp. 316–323, 2002.

[11] Q. Qian, C. Li, and X. Zhang, “On Authentication System Based on 802.1 X Protocol in LAN,” in *2010 International Conference on Internet Technology and Applications*, pp. 1–4, IEEE, 2010.

[12] M. Bartłomiejczyk, I. El Fray, M. Kurkowski, S. Szymoniak, and O. Siedlecka-Lamch, “User authentication protocol based on the location factor for a mobile environment,” *IEEE Access*, vol. 10, pp. 16439–16455, 2022.

[13] L. Lamport, “Password authentication with insecure communication,” *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[14] E. Jung and K. Hong, “Biometric verification based on facial profile images for mobile security,” *Journal of Systems and Information Technology*, vol. 17, no. 1, pp. 91–100, 2015.

[15] F. Bergadano, D. Gunetti, and C. Picardi, “User authentication through keystroke dynamics,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 4, pp. 367–397, 2002.

[16] F. Tari, A. A. Ozok, and S. H. Holden, “A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords,” in *Proceedings of the second symposium on Usable privacy and security*, pp. 56–66, 2006.

[17] W. Meng, D. S. Wong, and L.-F. Kwok, “The effect of adaptive mechanism on behavioural biometric based mobile phone authentication,” *Information Management & Computer Security*, vol. 22, no. 2, pp. 155–166, 2014.

[18] C. Herley and P. Van Oorschot, “A research agenda acknowledging the persistence of passwords,” *IEEE Security & privacy*, vol. 10, no. 1, pp. 28–36, 2011.

[19] T. Petsas, G. Tsirantonakis, E. Athanasopoulos, and S. Ioannidis, “Two-factor authentication: Is the world ready? Quantifying 2FA adoption,” in *Proceedings of the eighth European workshop on system security*, pp. 1–7, 2015.

[20] F. Yang and S. Manoharan, “A security analysis of the OAuth protocol,” in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 271–276, IEEE, 2013.

[21] M. Argyriou, N. Dragoni, and A. Spognardi, “Security flows in OAuth 2.0 framework: a case study,” in *Computer Safety, Reliability, and Security: SAFECOMP 2017 Workshops, ASSURE, DECSoS, SASSUR, TELERISE, and TIPS, Trento, Italy, September 12, 2017, Proceedings 36*, pp. 396–406, Springer, 2017.

[22] D. Hardt, “The Oauth 2.0 Authorization Framework,” *Internet Engineering Task Force (IETF) Request for Comments (RFC)*, vol. 6749, no. 20, pp. 1–39, 2012.

[23] M. Haekal *et al.*, “Token-based authentication using JSON web token on SIKASIR RESTful Web Service,” in *2016 International Conference on Informatics and Computing (ICIC)*, pp. 175–179, IEEE, 2016.

[24] S. Ahmed and Q. Mahmood, “An authentication based scheme for applications using JSON web token,” in *2019 22nd international multitopic conference (INMIC)*, pp. 1–6, IEEE, 2019.

[25] P. Mahindraka, “Insights of JSON Web Token,” *International International Journal of Recent Technology and Engineering (IJRTE) ISSN*, pp. 2277–3878, 2020.

[26] F. Staiano, *Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop.* Packt Publishing Ltd, 2022.

[27] A. Faulkner and Contributors, “Lucidchart for easy workflow mapping,” *Serials Review*, vol. 44, no. 2, pp. 157–162, 2018.