

1 Introduction

Searching in the internet can be a frustrating business. It is more frustrating when you are looking for a research-based articles. ETD Search is a web application designed for searching and querying of research thesis and dissertations. This application is going to provide advanced search options for searching through research database and make the search process quick and easy.

The remaining sections will provide a detailed description of features of ETD Search web application.

2 Architecture

The application is built in PHP with Laravel framework. Html, CSS, and JavaScript is used for the User Interface. It uses MySQL database for storing the user information. Search functionality is implemented with Elasticsearch query. The structure of the website explained in Major Functional Component Diagram (Figure 1) and Site Map (Figure 2).

Figure 1

Major Functional Component Diagram

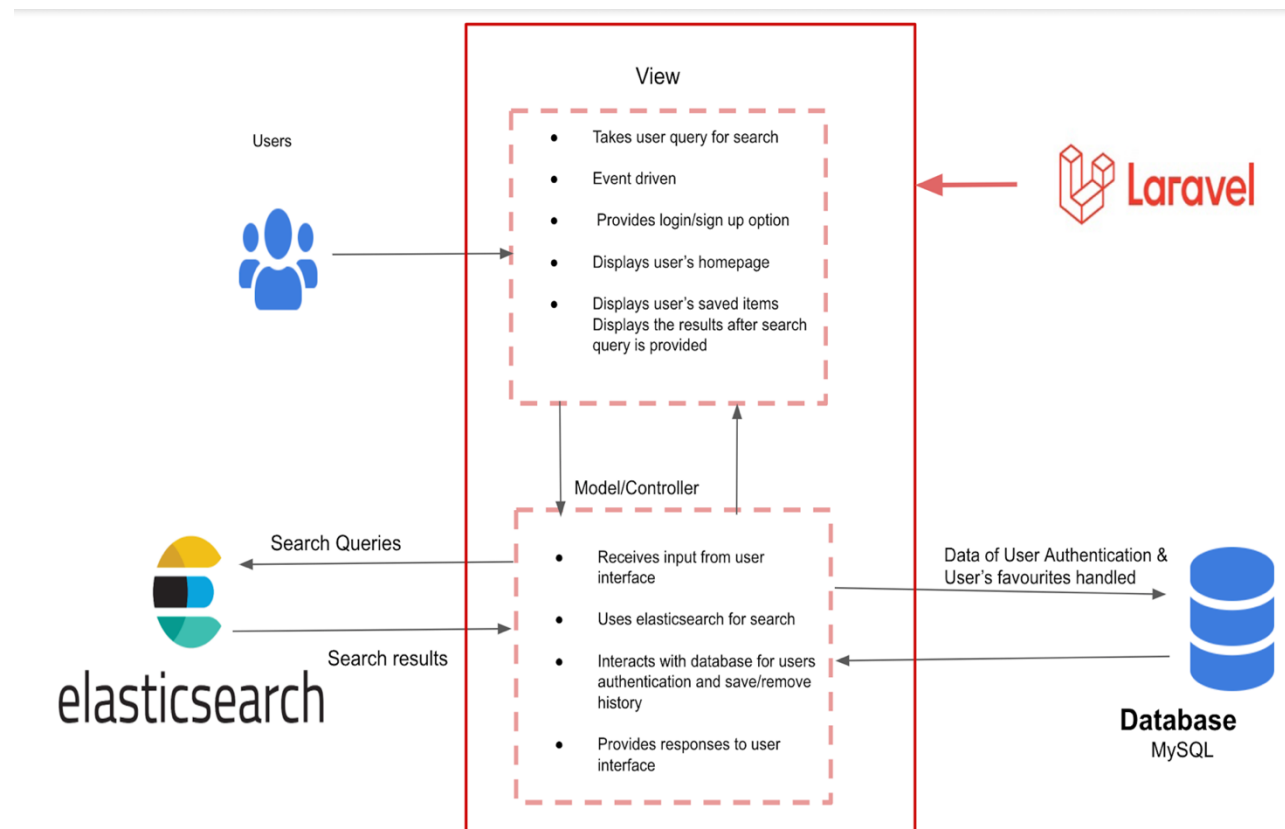
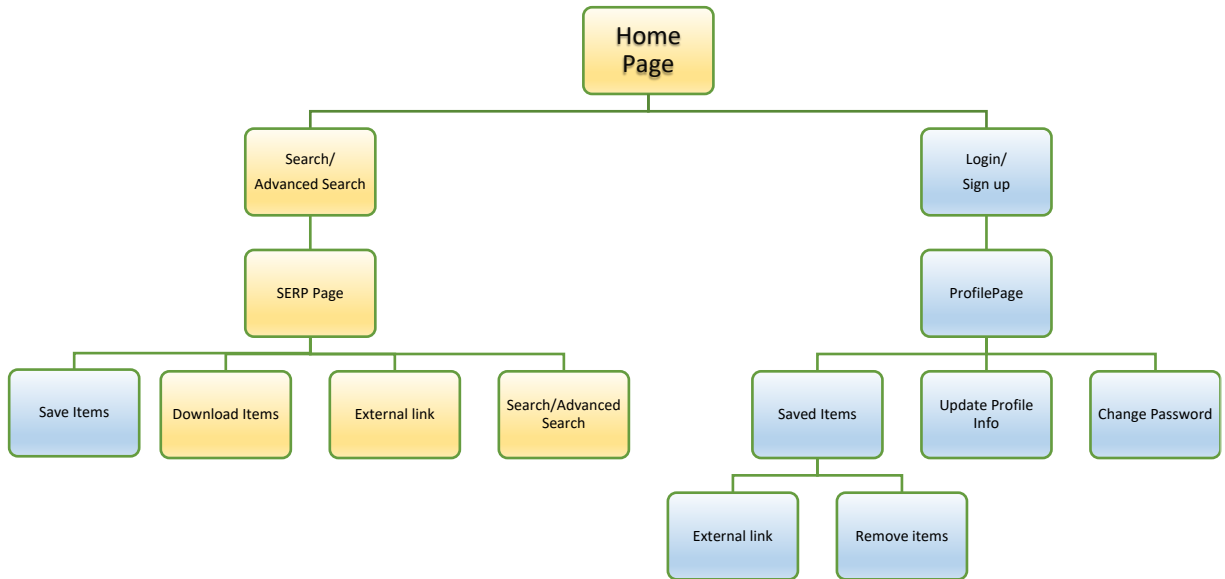


Figure 2

Site-Map



3 Milestone Accomplishments

Table with project specifications: Milestone 1, Milestone 2, Milestone 3 and Milestone 4

Fulfilled	#	Specification
Yes	1	The website should provide a search box at the landing page. The searching function may not be working.
Yes	2	There should be a search button next to the search box.
Yes	3	Users must be able to register new accounts using email addresses.

POOJA POUDEL-ETD SEARCH ENGINE REPORT

Yes	4	Password must be encrypted before storing in the database.
Yes	5	Users cannot register duplicate accounts using the same email address, or phone number.
Yes	6	Users should be able to log into your website using the accounts they registered.
Yes	7	Users should be able to reset their passwords if they forget it;
Yes	8	The user login process must use the HTTP POST method.
Yes	9	User information shall be stored in a MySQL database.
Yes	10	The website should have a homepage for each user, where they can view their profiles, change passwords, and update information.
Yes	11	Users should be able to get a confirmation email to verify their email addresses for registration or password reset.
Yes	12	The website provides an “Advanced Search” button in which users can specify more information.
Yes	13	The advanced search should return results satisfying multiple specifications.
Yes	14	The website should index at least 5000 “documents” (a document can be metadata of an image or metadata of an ETD).
Yes	15	The search engine should return search results on the search engine result page (SERP), which can be links to documents or images.
Yes	16	The search engine should display the number of returned items on SERP.
Yes	17	The SERP should contain a search box.
Yes	18	The search engine can prevent XSS vulnerability by removing tags existing in the query.
Yes	19	Users should be able to insert a new entry (document) and search engine will index it.
Yes	20	The search engine accepts a text query in the search box.
Yes	21	The search engine can highlight results that contain search terms.
Yes	22	The SERP should display the actual term (after sanitization) shown on top.
Yes	23	The search engine can return paginated results.

POOJA POUDEL-ETD SEARCH ENGINE REPORT

Yes	24	Users can click each item on SERP and go to either an external link or a page containing more information of the item.
Yes	25	Users can save items in search result to their profiles.
Yes	26	Users have to login first to save search history to their profiles.
Yes	27	reCAPTCHA should be used for both the logging in and the signing up page.
Yes	28	Users can delete items from their favorite list.
Yes	29	Items in the favorite lists should be descriptive (can't be just a link) and are linked to an external page or a summary page of the item.
Yes	30	The search engine implements at least one of the features spell check, autocomplete, Speech-to-text API, or other APIs permitted by the instructor.
Yes	31	There is a button from which users can download documents (or images) from the summary page or from the SERP (or both).

4 Data

Data of this website comes from ETD data provided by professor. It has 3966 electronic thesis and dissertations. The fields for indexing are:

- contributor_author
- contributor_committeechair
- contributor_committeemembers
- contributor_department
- date_accessioned
- date_available
- date_issued
- degree_discipline
- degree_grantor
- degree_level
- degree_name
- description_abstract
- description_degree
- description_provenance
- format_medium
- handle
- identifier_other
- identifier_uri
- publisher

- rights
- subject
- title
- type

For the **regular search**, the user's query is matched with contributor author, contributor committee chair, contributor committee members, contributor department, date issued, degree discipline, description abstract, description degree, description provenance, publisher, subject and the title.

And for the **advanced search**, user's query is matched with title, contributor author, publisher and date issued fields. For advanced search, user's input in the search box must match the specific field. For eg. if user enters "Programming" on the title search box, "Programming" must be a word in the title field of indexed document.

5 Implementation

Following features are implemented:

1. Account registration

For account registrations user's first name, last name, email and 8 characters long passwords are required. Users need to verify they aren't robot through reCAPTCHA and also need to verify their email for successful account registration.

2. Account login

For login, users need email and password. The password is matched with the encrypted stored password in "Users" table. reCAPTCHA need to be checked for successful login. The authentication process is done by Jet

3. Password reset

Password can be reset using the email. It can be reset from profile page after login in and providing correct current password. Another way to password reset is following forgot password link.

4. Users' homepage

User's homepage (aka Profile) provides current user information and option to change password. It has link to search page and Saved Items page.

5. Main search function

Main search function is implemented using Elasticsearch. It is implemented in SERP.blade.php. For the main search, the user's query is matched with contributor author, contributor committee chair, contributor committee members, contributor department, date issued, degree discipline, description abstract, description degree, description

provenance, publisher, subject and the title. Printing the search results is also implemented in SERP.blade.php.

6. Advanced search function

When the user clicks advanced search, a form in advSearch.blade.php handles it. It takes user inputs and routes to advSERP.blade.php which implements the elasticsearch query for advanced search. User's query is matched with title, contributor author, publisher and date issued fields. For advanced search, user's input in the search box must match the specific field. For eg. if user enters "Programming" on the title search box, "Programming" must be a word in the title field of indexed document.

7. SERP

SERP page provides title, description and publisher of the search results. It also provides external link to the ETD. It is implemented in SERP.blade.php

8. XSS vulnerability filtering

It is implemented using existing csrf function provided by Laravel. It is implemented in each form written in HTML throughout the project.

9. Highlighting search terms

Highlighting search terms is implemented using JavaScript in SERP.blade.php.

10. Pagination

Pagination is implemented using table format to print the data. Javascript is used for the table. Pagination is implemented in SERP.blade.php.

11. Save items to user's profiles

Users can save items if they are logged in from SERP page. The saved items can be accessed through clicking save items link from profile page. It is implemented by passing user id, title, url, description and publisher variables from SERP to the save route, which routes to Search/save.php. Save.php has SQL insertation commands to insert the data into the "histories" table of "SearchEngine" database.

It is displayed through Search/mySearch.php, in a table format. The data is accessed from the histories table. The title is clickable which links to the external page which provide more information about the ETD.

12. Remove saved items

Saved items can be removed from the Saved items page. It is done by matching item id with the database, if it matches, it is deleted from the database. This functionality is implemented in Search/remove.php.

13. Download

Download functionality is implemented in SERP.blade.php. The folder handle is obtained from “handle” field of json of the search item. When the download button is clicked all the files inside the folder are checked to see if they are pdf, if there is more than one pdf they are zipped together. So, the button downloads either one or zipped pdfs if there are more than one.

14. Google Speech-to-text API:

A microphone icon is provided next to the search box which can be clicked for giving voice queries for search. It is implemented in Landing (Search page) and in Advanced search page. It is also used in SERP page if users want to search again.

6 Challenges and Lessons

I faced two major challenges:

1. Understanding Laravel framework and MVC concept
Even though I was not experienced Laravel Framework before, I took this class as an opportunity to learn something new. It was quite challenging to understand MVC concept at first. I overcame it by taking more time to understand the basis.
2. Elasticsearch:
Because it was completely new to me, I have had hard time implementing it into the search Engine. I overcame it by focusing on understanding Elasticsearch search mechanism. My lesson from this was, I shouldn't try to integrate something in my application without understanding the basics of how it works.

If I had to do it over again, I would take more time to understand MVC concept in milestone one.

App Screenshots

Image 1

ETD Search Landing Page

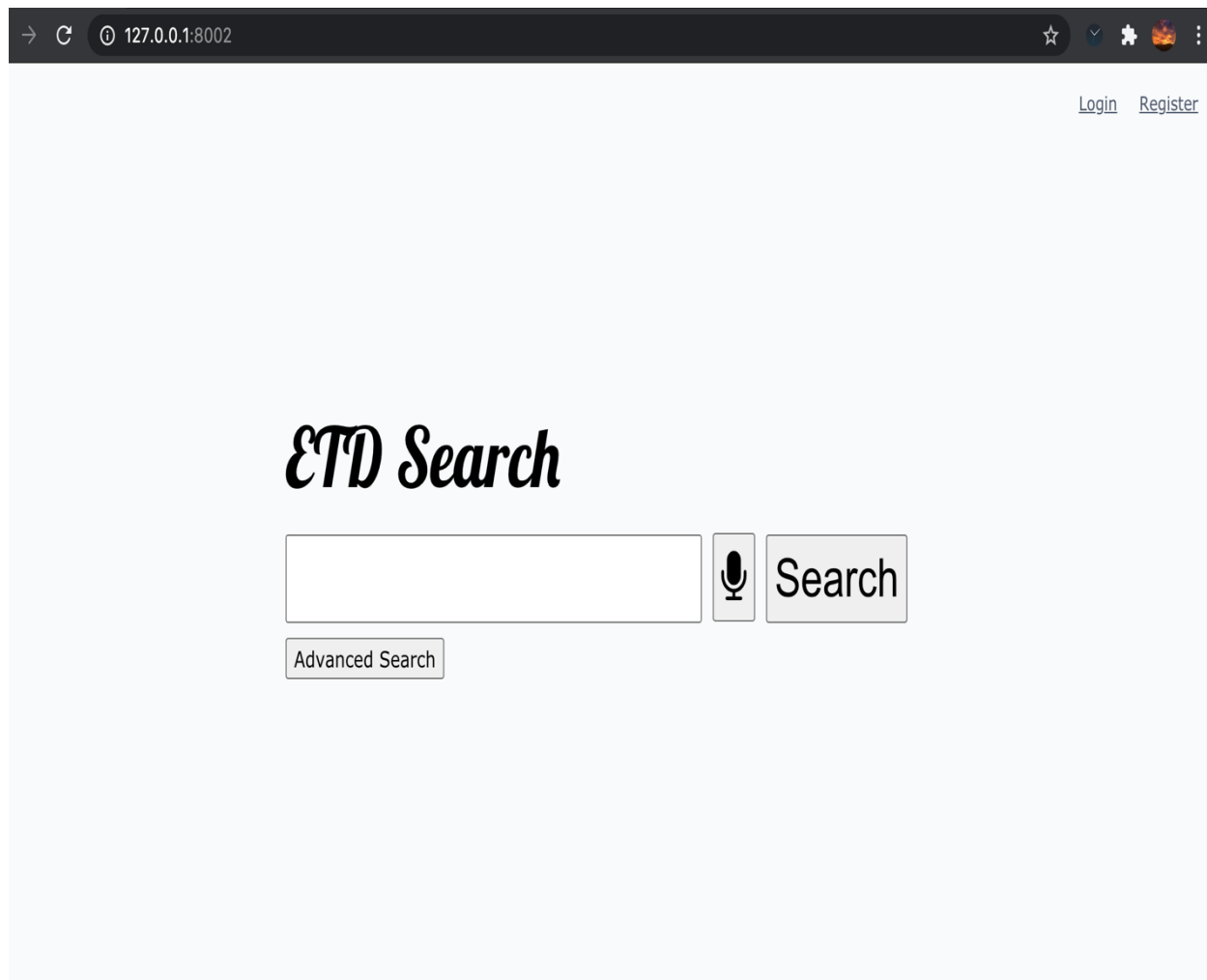
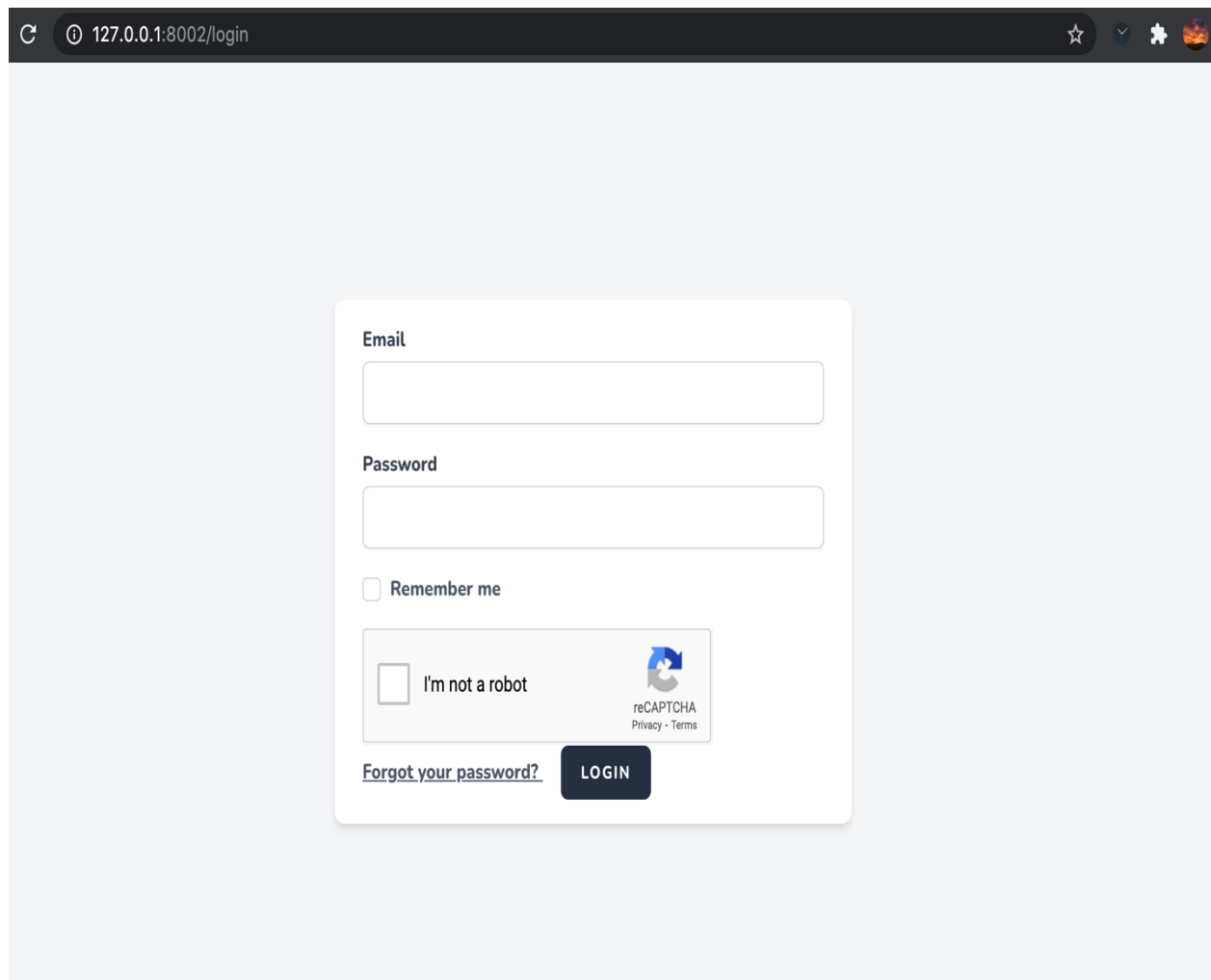


Image 2

ETD Search Login Page



The image shows a web browser window with the address bar displaying "127.0.0.1:8002/login". The page features a login form with the following elements:

- Email:** A text input field.
- Password:** A text input field.
- ☐ **Remember me**
- ☐ **I'm not a robot** (reCAPTCHA widget)
- [Forgot your password?](#)
- LOGIN** button

POOJA POUDEL-ETD SEARCH ENGINE REPORT

Image 3

User Profile Page

The screenshot displays a web browser window with the URL `127.0.0.1:8002/user/profile`. The page title is "Pooja Poudel". The main content area is titled "Profile" and features a "SAVED ITEMS" button. Below this, there are two sections: "Profile Information" and "Update Password".

Profile Information
Update your account's profile information and email address.

Name

Email

Update Password
Ensure your account is using a long, random password to stay secure.

Current Password

New Password

Confirm Password

Image 4

Authenticated- Search Engine Result Page-Top

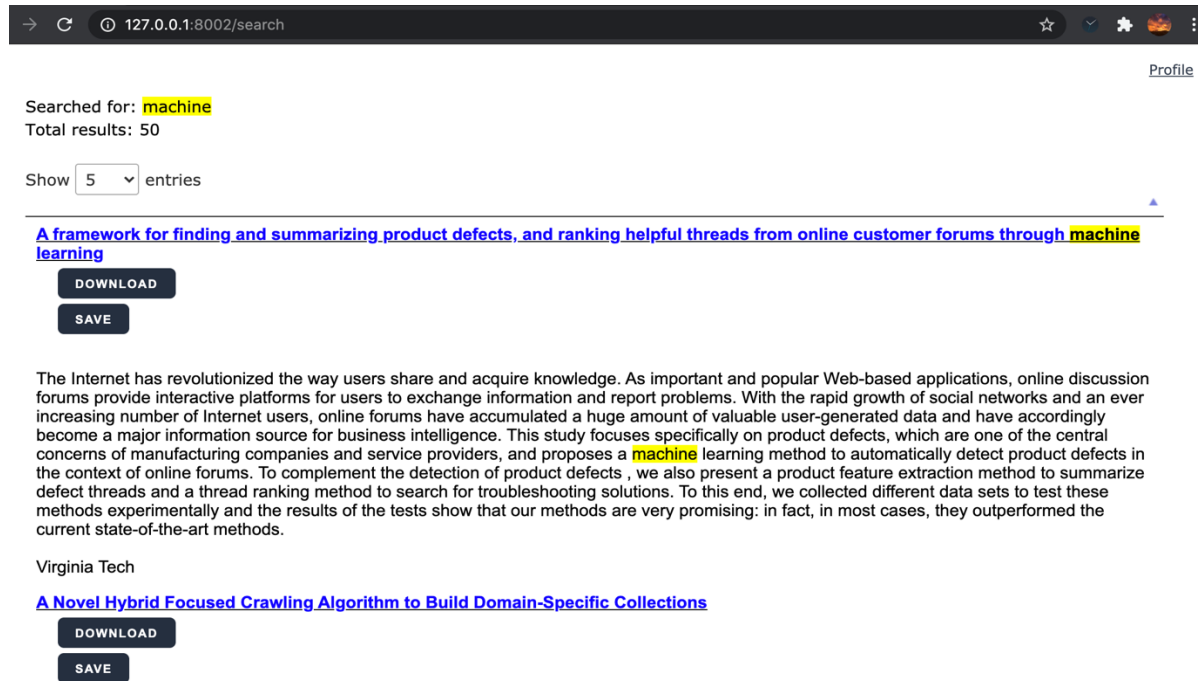


Image 5

Authenticated- Search Engine Result Page- Bottom

less attention to the loading of **machines** and initial tool allocation at the planning stage. In this research, a **machine** loading model with shared tools is proposed to maximize routing flexibility while maintaining minimum resident tools. The performance of the proposed loading heuristic is compared to that of a random loading method using hypothetically generated single stage system models. The study result indicates that better system performances can be obtained by taking into account the resident tooling ratio in assigning part types and allocating tools to **machines** at the initial planning stage.

Virginia Tech

[Algorithms for Reconstructing and Reasoning about Chemical Reaction Networks](#)

DOWNLOAD

SAVE

Recent advances in systems biology have uncovered detailed mechanisms of biological processes such as the cell cycle, circadian rhythms, and signaling pathways. These mechanisms are modeled by chemical reaction networks (CRNs) which are typically simulated by converting to ordinary differential equations (ODEs), so that the goal is to closely reproduce the observed quantitative and qualitative behaviors of the modeled process.

This thesis proposes two algorithmic problems related to the construction and comprehension of CRN models. The first problem focuses on reconstructing CRNs from given time series. Given multivariate time course data obtained by perturbing a given CRN, how can we systematically deduce the interconnections between the species of the network? We demonstrate how this problem can be modeled as, first, one of uncovering conditional independence relationships using buffering experiments and, second, of determining the properties of the individual chemical reactions. Experimental results demonstrate the effectiveness of our approach on both synthetic and real CRNs.

The second problem this work focuses on is to aid in network comprehension, i.e., to understand the motifs underlying complex dynamical behaviors of CRNs. Specifically, we focus on bistability---an important dynamical property of a CRN---and propose algorithms to identify the core structures responsible for conferring bistability. The approach we take is to systematically infer the instability causing structures (ICs) of a CRN and use **machine** learning techniques to relate properties of the CRN to the presence of such ICs. This work has the potential to aid in not just network comprehension but also model simplification, by helping reduce the complexity of known bistable systems.

Virginia Tech

Showing 1 to 5 of 50 entries

Previous 1 2 3 4 5 ... 10 Next

Image 6

Search Engine Result Page- No Authentication

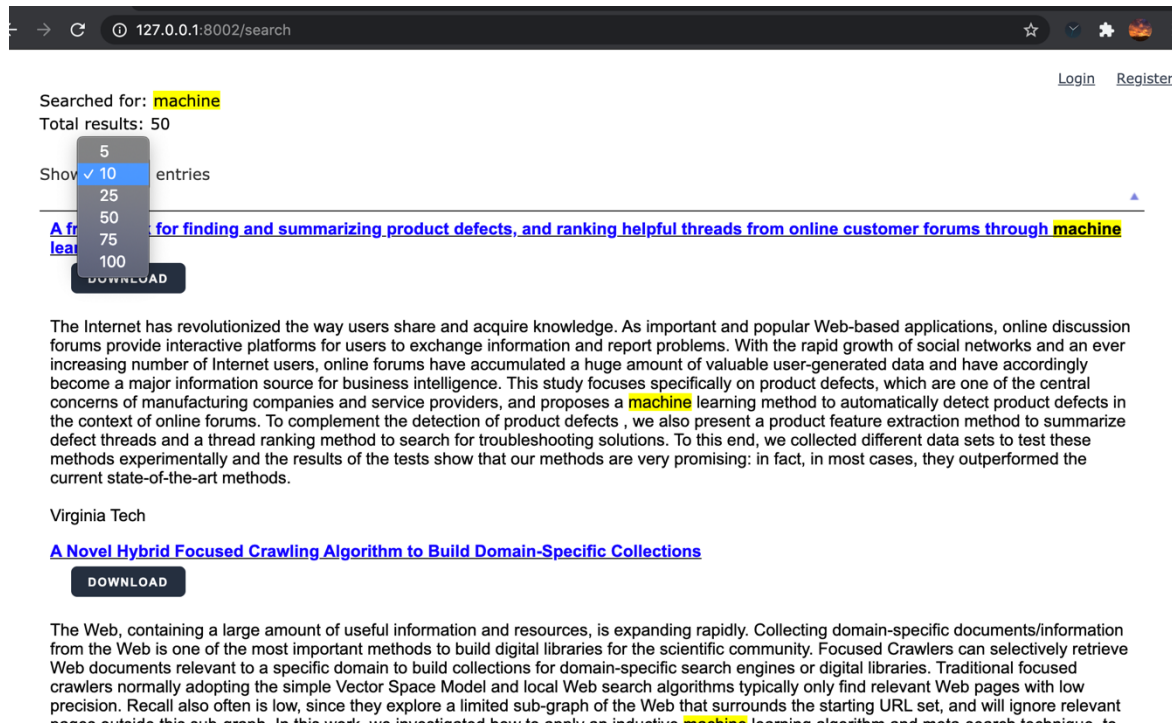
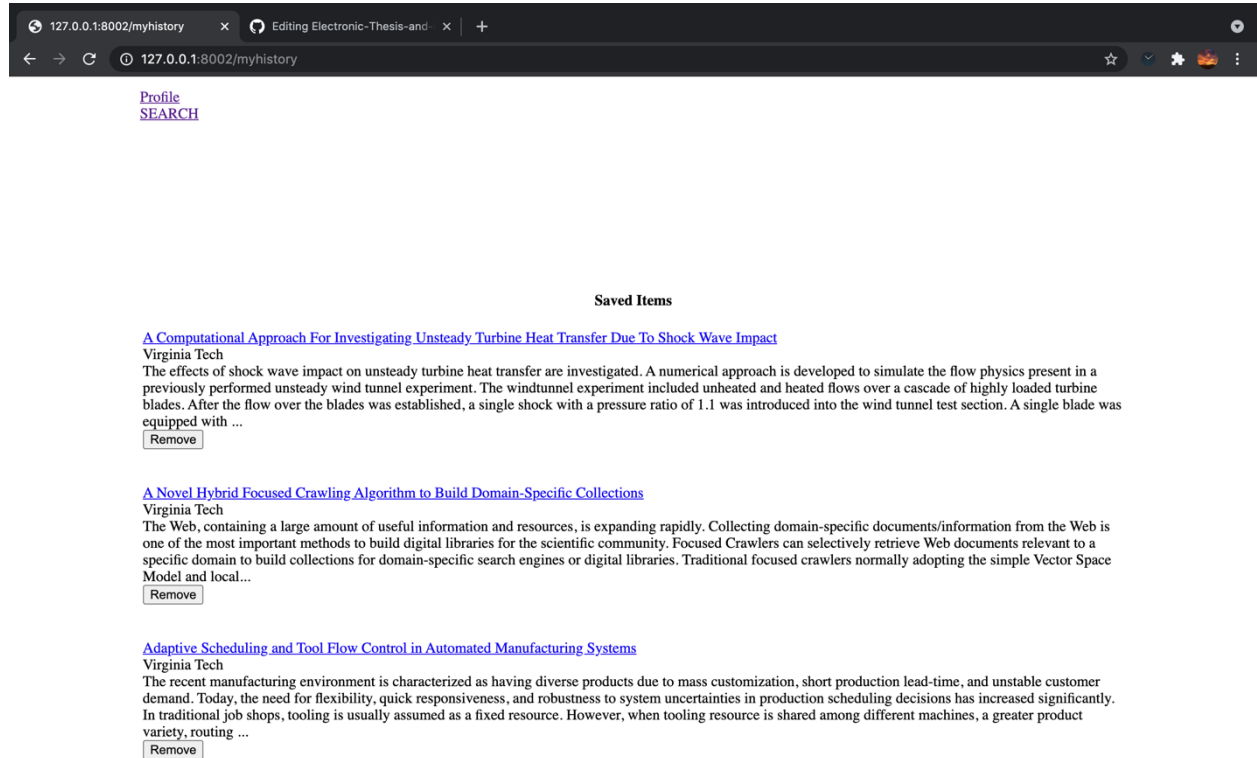


Image 7

User's Saved Items Page



POOJA POUDEL-ETD SEARCH ENGINE REPORT

Image 8

MySQL – Users Table

The screenshot displays the phpMyAdmin interface for the 'SearchEngine' database, specifically the 'users' table structure. The table is defined with 12 columns, each with specific attributes like type, collation, and nullability. Below the column list, the indexes are shown, including a primary key on 'id' and a unique index on 'email'.

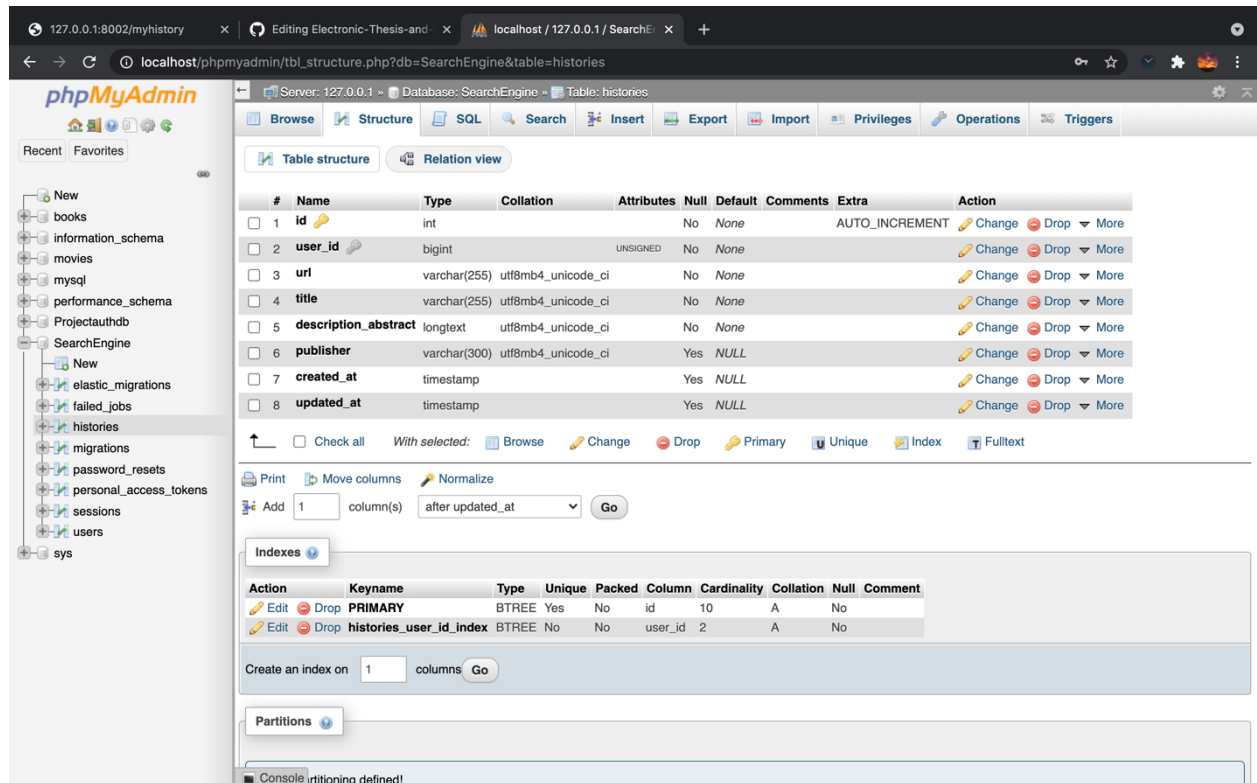
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	email_verified_at	timestamp			Yes	NULL			Change Drop More
5	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	two_factor_secret	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
7	two_factor_recovery_codes	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
8	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
9	current_team_id	bigint		UNSIGNED	Yes	NULL			Change Drop More
10	profile_photo_path	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
11	created_at	timestamp			Yes	NULL			Change Drop More
12	updated_at	timestamp			Yes	NULL			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	65	A	No	
Edit Drop	users_email_unique	BTREE	Yes	No	email	65	A	No	

POOJA POUDEL-ETD SEARCH ENGINE REPORT

Image 8

MySQL – Users Saved Items Table



Server: 127.0.0.1 » Database: SearchEngine » Table: histories

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	bigint		UNSIGNED	No	None			Change Drop More
3	url	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	description_abstract	longtext	utf8mb4_unicode_ci		No	None			Change Drop More
6	publisher	varchar(300)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
7	created_at	timestamp			Yes	NULL			Change Drop More
8	updated_at	timestamp			Yes	NULL			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	10	A	No	
Edit Drop	histories_user_id_index	BTREE	No	No	user_id	2	A	No	

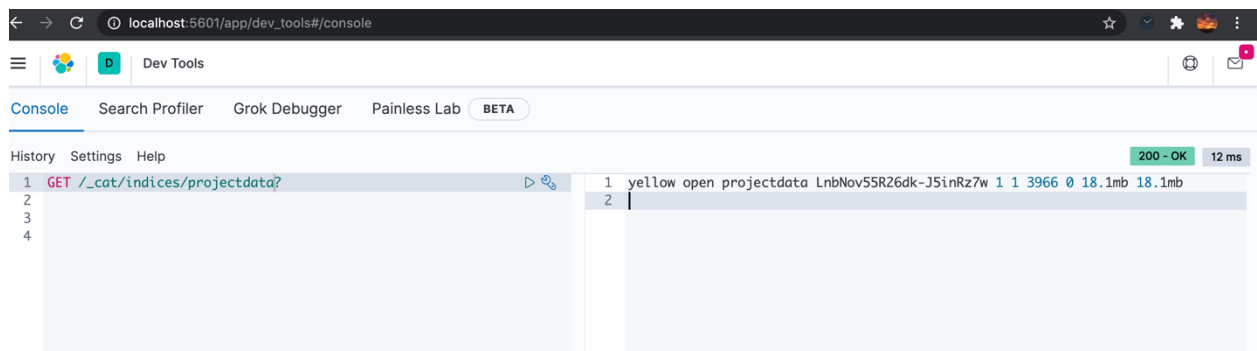
Create an index on 1 column(s) Go

Partitions

Console |rtitioning defined!

Image 9

Kibana – 3966 documents indexed



localhost:5601/app/dev_tools#/console

Dev Tools

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Help

```
1 GET /_cat/indices/projectdata?
2
3
4
```

```
1 yellow open projectdata LnbNov55R26dk-J5inRz7w 1 1 3966 0 18.1mb 18.1mb
2
```

200 - OK 12 ms